School of Computer Science

B. Tech (Computer Science & Engineering) with Specialization

Programme Handbook

2023-2027

UPES Campus, "Energy Acres" & "Knowledge Acres"
Bidholi & Kandoli
Dehradun, Uttarakhand-248001

Tel : + 91-135-2776053/54
Fax: + 91-135-2776090
URL: www.upes.ac.in

Contents

Intellectual Property Rights

1.0 Abbreviations

Cat	-	Category
Cr	-	Credits (*A credit is equivalent to one lecture hour/ one hour of tutorial/ two hours of Laboratory)*
L	-	Lecture
T	-	Tutorial
P	-	Practical
ENGG	-	Engineering Sciences (including General, Core)
HUM	-	Humanities (including Languages, Social Sciences, and others)
SCI	-	Basic Sciences (including Mathematics)
PRJ	-	Project Work (including Seminars, Dissertation, and Internships)
PE	-	Program Elective (includes Specialization courses)
TC	-	Total Credits
AIE	-	Computer Science and Engineering-Artificial Intelligence
CCE	-	Computer Science and Communication Engineering
CHY	-	Chemistry
CSE	-	Computer Science and Engineering
CVL	-	Civil Engineering
CUL	-	Cultural Education
MAT	-	Mathematics
PHY	-	Physics
UE	-	University Elective (includes Signatory, Exploratory and Open Electives)

1.0 Dean’s Message

Welcome to UPES-School of Computer Science's B. Tech in Computer Science and Engineering program. Our curriculum combines theory and hands-on experience to prepare you for the fast-changing world of Computer Science. Our faculty consists of accomplished researchers and industry professionals who are committed to academic excellence. In addition to traditional classrooms, you will have opportunities for real-world applications, networking, and industry partnerships. Embrace challenges, approach your academic journey with enthusiasm, and remember that you are part of a community dedicated to knowledge and innovation. With your dedication and the resources provided by UPES-SoCS, you will become a skilled professional and visionary leader to make a transformative impact on the global stage.

2.0 Vision and Mission of the University:

Vision of UPES

To be an Institution of Global standing for developing professionally competent talent contributing to nation building.

Mission of UPES

- Develop industry-focused professionals with an international outlook.
- Foster effective outcome-based education system to continually improve teaching-learning and research.
- Inculcate integrative thought process among students to instill lifelong learning.
- Create global knowledge eco-system through training, research & development and consultancy.
- Practice and promote high standards of professional ethics and develop harmonious relationship with environment and society.

3.0 Vision and Mission of the School

Vision

To create a dynamic, collaborative, and inclusive community that advances technology, fosters creativity, and empowers individuals to make a positive global impact. The school aspires to lead and inspire in the development of new technology, and the application of technology to solve problems of society.

Mission

Our mission is to empower the next generation of professionals in Computer Science with the knowledge, skills, and innovative mindset required to shape the future. We are dedicated to fostering a vibrant learning community where students, faculty, and industry partners collaborate to advance society and the frontiers of technology.

- Excellence in Education: Delivering a rigorous and multi-disciplinary computer science curriculum that equips our students with a deep understanding of foundational principles and the ability to adapt to evolving technologies.
- Research and Innovation: Striving to be at the forefront of technology research, pushing boundaries in areas such as artificial intelligence, data science, IoT and cybersecurity.
- Industry Engagement: Actively engaging with industry partners to ensure that our programs remain relevant and provide opportunities for students to apply their knowledge in real-world settings. Encouraging our faculty members to collaborate with industry partners on real-world projects.
- Entrepreneurship: Strongly encouraging entrepreneurship both among the faculty members and the students.
- Ethical Computing: Emphasizing the importance of ethical and responsible use of technology, instilling in our faculty, students, and the society at large a sense of social responsibility and ethical decision-making.
- Lifelong Learning: Encouraging a culture of lifelong learning among our students, faculty, and alumni, ensuring that they continue to adapt and thrive in the ever-changing field of computer science.

Through these commitments, we aim to produce graduates who are not only technically proficient but also visionary leaders who will drive positive change in the world through the transformative power of computer science.

4.0 About the School

The UPES School of Computer Science is founded with futuristic approach, to provide cutting-edge computer science education, to empower future generations to harness the transformative power of technology. The school is organized into four clusters named Artificial Intelligence, Data Science, Cloud & S/W Operations and Systems. The SoCS faculty comprises accomplished researchers and industry professionals dedicated to academic excellence. The school offers various programs like B.Tech., BCA, B.Sc., M.Tech., MCA, Ph.D. A well-crafted curriculum aligned with the industry needs, well-equipped laboratories, innovative pedagogy, are the strength of the school.

Research Focus

The school research focus encompasses cutting-edge domains, including artificial intelligence, data science, IoT, and cybersecurity. We delve into innovative studies at the intersection of these fields, driving advancements in technology.

5.0 Programme Overview

The Bachelor of Technology in Computer Science and Engineering program provides a comprehensive education in the fundamental principles and advanced concepts of computer science. Students engage in a rigorous curriculum covering areas such as algorithms, data structures, software engineering, artificial intelligence, and machine learning. The program offers various specializations through it’s program elective courses. The program emphasizes hands-on experience through practical projects, internships, and industry collaborations, enabling students to apply theoretical knowledge to real-world scenarios. With a focus on innovation and problem-solving, B.Tech CSE equips graduates with the skills to excel in diverse roles within the rapidly evolving field of computer science, fostering a strong foundation for professional success and contributions to technological advancements

6.0 Programme Educational Objectives

- **Technical Excellence:**
  Graduates will demonstrate a strong foundation in mathematical, scientific, and engineering fundamentals essential for addressing Computer Science & Engineering challenges. They will excel in technical roles, including design, development, problem-solving, and production support within software industries and R&D sectors.
- **Continuous Learning and Advancement:**
  Graduates will pursue higher education in reputable institutions, enriching their knowledge and specialization in Computer Science and Engineering.
- **Innovation and Adaptability:**
  Graduates will innovate and adapt evolving technologies to creatively address contemporary issues in Computer Science and Engineering domains.
- **Ethical and Social Responsibility:**
  Graduates will exhibit ethical and social responsibility as solution providers and entrepreneurs in Computer Science and related engineering disciplines. They will uphold high ethical values and demonstrate empathy for societal needs.

7.0 Programme Outcome and Programme Specific Outcomes

Programme Outcomes

1. Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.
6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. Individual and teamwork: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
11. Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Programme Specific Outcomes

1. Perform system and application programming using computer system concepts, the principles of data structures, algorithm development, problem-solving, and optimization techniques.
2. Apply software development and project management methodologies, integrating principles from both front-end and back-end development, and effectively utilize contemporary tools and technologies.

3. Exhibit a commitment to ethical practices, societal responsibilities, and continuous learning, contributing to the advancement of technology and addressing challenges in diverse computing domains.

8.0 Academic Integrity Policy

a. University Integrity Policy

b. Course integrity policy

9.0 Overview of Credit Allocation/ Credit Break up

Category-wise Credit distribution

Category	Number of Credits
Major Core (MC)	63
Basic Sciences - Core (SCI)	26
Engineering Sciences - Core (ENGG)	08
Major Elective (ME)	24
Signature courses (SC)*	
Life Skill Courses (LSC)*	10
Exploratory Courses (EC)*	15
Humanities (HUM)	
Projects (PRJ)	16
Total	162

* Electives

- Major core subjects include those subjects that are mandatory to all similar programmes and program specific courses. To be eligible for the degree, students must successfully finish each of the courses.
- Major elective courses provide the students the opportunity to study courses that are more complex and specialized, in their field of specialization.
-

Major Core (MC)	**Total number of Credits: 63 Credits**				
Course Code	**Course Title**	**L**	**T**	**P**	**TC**
CSEG1126	Linux Lab	0	0	4	2
CSEG1025	Programming in C	0	0	3	3
CSEG1125	Programming in C Lab	0	0	4	2
CSEG1033	Data Structures and algorithms	4	0	0	4
CSEG1034	Data Structures and algorithms Lab	0	0	2	1
CSEG1035	Python programming	2	0	0	2
CSEG1135	Python programming Lab	0	0	4	2
CSEG2046	Database Management Systems	3	0	0	3
CSEG2146	Database Management Systems Lab	0	0	4	2
CSEG2006	Discrete Mathematical Structures	3	0	0	3
CSEG2020	Object Oriented Programming	3	0	0	3
CSEG2120	Object Oriented Programming Lab	0	0	2	1
CSEG2060	Operating Systems	3	0	0	3
CSEG2064	Software Engineering	3	0	0	3
CSAI2015	Elements of AIML	2	0	0	2
CSAI2115	Elements of AIML Lab	0	0	2	1
CSEG2065	Data communication and Networks	3	0	0	3
CSEG2165	Data communication and Networks Lab	0	0	2	1
CSEG2021	Design and Analysis of Algorithms	3	0	0	3
CSEG2121	Design and Analysis of Algorithms Lab	0	0	2	1
CSEG3040	Cryptography and Network Security	3	0	0	3

CSEG3055	Formal Languages and Automata Theory	3	0	0	3
CSEG3002	Object Oriented Analysis and Design	3	0	0	3
CSEG3060	Research Methodology in CS	3	0	0	3
CSEG3015	Compiler Design	3	0	0	3
CSEG4038	IT Ethical Practices	3	0	0	3
				Total Credits	63
BASIC SCIENCES-Core(SCI)	**Total Number of Credits: 26 Credits**				
Course Code	**Course Title**	**L**	**T**	**P**	**TC**
MATH1059	Advanced Engineering Mathematics – 1	3	1	0	4
SSEN0101	Environmental Sustainability and Climate Change	2	0	0	2
PHYS1032	Physics for Computer Engineers	4	0	0	4
PHYS1132	Physics Lab for Computer Engineers	0	0	2	1
MATH1065	Advanced Engineering Mathematics – 2	3	1	0	4
SSEN0102	Environmental Sustainability and Climate Change (Living Lab)	2	0	0	2
MATH2059	Linear Algebra	3	0	0	3
CSEG3056	Probability, Entropy, and MC Simulation	3	0	0	3
CSEG3057	Statistics and Data Analysis	3	0	0	3
	Total Credits				**26**
Engineering Sciences	**Total Number of Credits: 8 Credits**				
Course Code	**Course Title**	**L**	**T**	**P**	**TC**
CSEG1027	Problem Solving	2	0	0	2
CSEG1032	Computer organization and Architecture	3	0	0	3
ECEG1012	Digital Electronics	3	0	0	3
				Total Credits	**8**
Major Elective (ME)	**Total Number of Credits: 24 Credits**				
Course Code	**Course Title**	**L**	**T**	**P**	**TC**
	PE-1	4	0	0	4
	PE-1 Lab	0	0	2	1
	PE-2	4	0	0	4
	PE-2 Lab	0	0	2	1
	PE-3	4	0	0	4
	PE-3 Lab	0	0	2	1
	PE-4	4	0	0	4
	PE-4 Lab	0	0	2	1
	PE-5	3	0	0	3
	PE-5 Lab	0	0	2	1
				Total Credits	**24**
Life Skills Courses	**Total Number of Credits: 10 Credits**				
Course Code	**Course Title**	**L**	**T**	**P**	**TC**
SLLS0101	Living Conversations	2	0	0	2
SLSG0102	Critical Thinking and Writing	2	0	0	2
SLLS0201	Design Thinking	0	0	0	2
EMPL002	EDGE-SoftSkills	1	0	0	0
SLSG0205	Start your startup	2	0	0	2
EMPL003	EDGE – Advance Communication	1	0	0	0
SLLS0103	Leadership and Teamwork	2	0	0	2
				Total Credits	**10**
Exploratory Courses	**Total Number of Credits: 15 Credits**				
Course Code	**Course Title**	**L**	**T**	**P**	**TC**
	Exploratory-1	0	0	0	3

	Exploratory-2	3	0	0	3
	Exploratory-3	3	0	0	3
	Exploratory-4	3	0	0	3
	Exploratory-5	3	0	0	3
				Total Credits	**15**
Projects (PRJ)	**Total Number of Credits: 16 Credits**				
Course Code	**Course Title**	**L**	**T**	**P**	**TC**
PROJ3154	Minor Project	0	0	5	5
PROJ4145	Capstone Project - Phase-1	0	0	5	5
SIIB4102	Summer Internship	0	0	0	1
PROJ4146	Capstone Project - Phase-2	0	0	5	5
				Total Credits	**16**

10.0 Programme Structure

The term "Program Structure" refers to a list of courses (Core, Elective, and Open Elective) that make up an academic program, describing the syllabus, credits, hours of instruction, assessment and examination systems, minimum amount of credits necessary for program graduation, etc.

Sample: B. Tech (Computer Science & Engineering) with Specialization

Semester I:

Cat	Course Code	Course Title	L	T	P	TC	prerequisites
	CSEG1126	Linux Lab	0	0	4	2	Exposure to computer systems.
	CSEG1025	Programming in C	3	0	0	3	Elementary knowledge of Computer
	CSEG1125	Programming in C Lab	0	0	4	2	Elementary knowledge of Computer Systems
	CSEG1027	Problem Solving	2	0	0	2	10+2 Mathematics
	SSEN0101	Environment Sustainability & Climate Change	2	0	0	2	Fundamentals of basic ecology, chemistry and physics
	SLLS0101	Living Conversations	2	0	0	2	
	MATH1059	Advanced Engineering Mathematics – 1	3	1	0	4	Basic Mathematics (10+2 level)
	PHYS1032	Physics for Computer Engineers	4	0	0	4	12^{th} Level Physics
	PHYS1132	Physics Lab for Computer Engineers	0	0	2	1	Basic knowledge on practical Physics (12th level) for

							understanding and performing experiments.
						Semester Credits	22

Semester II:

Cat	Course Code	Course Title	L	T	P	TC	Prerequisites
	CSEG1032	Computer Organization and Architecture	3	0	0	3	Basic Knowledge of computer systems.
	CSEG1033	Data Structures and Algorithms	4	0	0	4	Programming in C
	CSEG1034	Data Structures and Algorithms Lab	0	0	2	1	Programming in C Lab
	CSEG1035	Python Programming	2	0	0	2	Problem Solving.
	CSEG1135	Python Programming Lab	0	0	4	2	Python Programming
	ECEG1012	Digital Electronics	3	0	0	3	Computer System Architecture
	SLSG0102	Critical Thinking and Writing	2	0	0	2	
	SSEN0102	Environment Sustainability and Climate Change (living Lab)	2	0	0	2	Fundamentals of basic Environment Sustainability and Climate Change
	MATH1065	Advanced Engineering Mathematics – 2	3	1	0	4	Advanced Engineering Mathematics - I
						Semester Credits	23

Semester III:

Cat	Course Code	Course Title	L	T	P	TC	Prerequisites
	CSAI2015	Elements of AIML	2	0	0	2	Problem Solving Techniques, Probability and Statistical Analysis
	CSAI2115	Elements of AIML Lab	0	0	2	1	Python Programming
	CSEG2046	Database Management Systems	3	0	0	3	Exposure on Data structures.
	CSEG2146	Database Management Systems Lab	0	0	4	2	
	CSEG2021	Design and Analysis of Algorithms	3	0	0	3	Data structures and algorithms
	CSEG2121	Design and Analysis of Algorithms Lab	0	0	2	1	
	CSEG2006	Discrete Mathematical Structures	3	0	0	3	Advanced Engineering Mathematics – 2

	CSEG2060	Operating Systems	3	0	0	3	Computer Organization and Architecture.
		Exploratory-1	3	0	0	3	
	SLLS0201	Design Thinking	1	1	0	2	
						Semester Credits	23

Semester IV:

Cat	Course Code	Course Title	L	T	P	TC	Prerequisites
	CSEG2065	Data communication and Networks	3	0	0	3	Digital Electronics
	CSEG2165	Data communication and Networks Lab	0	0	2	1	
	CSEG2020	Object Oriented Programming	3	0	0	3	Programming in C
	CSEG2120	Object Oriented Programming Lab	0	0	2	1	Programming in C Lab
	CSEG2064	Software Engineering	3	0	0	3	
		Exploratory-2	3	0	0	3	
	MATH2059	Linear Algebra	3	0	0	3	Discrete Mathematics, Advanced Engineering Mathematics 1 & 2
	SLLS2004	Indian Constitution	0	0	0	0	
	EMPL002	EDGE-SoftSkills	1	0	0	0	
		PE-1	4	0	0	4	
		PE-1 Lab	0	0	2	1	
						Semester Credits	22

Semester V:

Cat	Course Code	Course Title	L	T	P	TC	Prerequisites
	CSEG3040	Cryptography and Network Security	3	0	0	3	Data Communication and Networks
	CSEG3055	Formal Languages and Automata Theory	3	0	0	3	Discrete Mathematics and Design and Analysis of Algorithms
	CSEG3002	Object Oriented Analysis and Design	3	0	0	3	Software Engineering
		Exploratory-3	3	0	0	3	
	SLSG0205	Start your Startup	2	0	0	2	
	CSEG3060	Research Methodology in CS	3	0	0	3	
	CSEG3056	Probability, Entropy, and MC Simulation	3	0	0	3	Advanced Engineering Mathematics – 1 & 2, Discrete

							Mathematics, Linear Algebra
	EMPL003	EDGE – Advance Communication	1	0	0	0	
		PE-2	4	0	0	4	
		PE-2 Lab	0	0	2	1	
						Semester Credits	25

Semester VI:

Cat	Course Code	Course Title	L	T	P	TC	Prerequisites
		Exploratory-4	3	0	0	3	
	SLLS0103	Leadership and Teamwork	2	0	0	2	
	CSEG3015	Compiler Design	3	0	0	3	Automata Theory and Formal Languages Data structures, knowledge of automata theory, basic knowledge of computer architecture
	CSEG3057	Statistics and Data Analysis	3	0	0	3	
	PROJ3154	Minor Project	0	0	0	5	
	EMPL004	EDGE – Advance Communication II	1	0	0	0	
		PE-3	4	0	0	4	
		PE-3 Lab	0	0	2	1	
						Semester Credits	21

Semester VII:

Cat	Course Code	Course Title	L	T	P	TC	Prerequisites
	PROJ4145	Capstone Project - Phase-1	0	0	0	5	Minor Project
	SIIB4102	Summer Internship	0	0	0	1	
		Exploratory-5	3	0	0	3	
		PE-4	4	0	0	4	
		PE-4 Lab	0	0	2	1	
		PE-5	3	0	0	3	
		PE-5 Lab	0	0	2	1	
						Semester Credits	18

Semester VIII:

Cat	Course Code	Course Title	L	T	P	TC	Prerequisites
	PROJ4146	Capstone Project - Phase-2	0	0	0	5	
	CSEG4038	IT Ethical Practices	3	0	0	3	
						Semester Credits	8

Specialization Tracks

The students enrolled in B.Tech Computer Science would have an option to specialize in one the following emerging areas:

1. Artificial Intelligence and Machine Learning
2. DevOps
3. Cloud Computing And Virtualization Technology
4. Full Stack Development
5. Cyber Security and Digital Forensics
6. Big Data
7. Data Science
8. Internet of Things (IoT)
9. Graphics & Gaming

The student must complete a minimum of 24 credits in the chosen area of specialization.

List of elective courses in specialization tracks

Program Elective 24 Credits						
Track 1 : Artificial Intelligence and Machine Learning						
Cat	**Course Code**	**Course Title**	**L**	**T**	**P**	**TC**
	CSAI2016P	Applied Machine Learning	4	0	0	4
	CSAI2116P	Applied Machine Learning Lab	0	0	2	1
	CSAI3025P	Deep Learning	4	0	0	4
	CSAI3125P	Deep Learning Lab	0	0	2	1
	CSAI3026P	Pattern and Visual Recognition	4	0	0	4
	CSAI3126P	Pattern and Visual Recognition Lab	0	0	2	1
	CSEG4034P	Computational Linguistics and Natural Language Processing	4	0	0	4
	CSEG4134P	Computational Linguistics and Natural Language Processing Lab	0	0	2	1
	CSAI4013P	Algorithm for Intelligent Systems and Robotics	3	0	0	3
	CSAI4113P	Algorithm for Intelligent Systems and Robotics Lab	0	0	2	1
						24
Track 2 : DevOps						
	CSDV2009P	DevOps Fundamentals and SCM	4	0	0	4
	CSDV2109P	DevOps Fundamentals and SCM LAB	0	0	2	1
	CSDV3022P	DevSecOps: Integrating security into DevOps practices	4	0	0	4
	CSDV3122P	DevSecOps: Integrating Security in DevOps Practices Lab	0	0	2	1

	CSDV3019P	Container Orchestration and Security	4	0	0	4
	CSDV3119P	Container Orchestration and Security Lab	0	0	2	1
	CSDV4009P	CICD Pipeline and Security	4	0	0	4
	CSDV4109P	CICD Pipeline and Security Lab	0	0	2	1
	CSDV4010P	System Provisioning and Monitoring	3	0	0	3
	CSDV4110P	System Provisioning and Monitoring Lab	0	0	2	1
Track 3 : Cloud Computing And Virtualization Technology						
	CSVT2010P	Cloud Computing Fundamentals	4	0	0	4
	CSVT2109P	Cloud Computing Fundamentals Lab	0	0	2	1
	CSVT3029P	Cloud Computing Architecture and Deployment Models	4	0	0	4
	CSVT3129P	Cloud Computing Architecture and Deployment Models Lab	0	0	2	1
	CSDV3018P	Containerization and DevOps	4	0	0	4
	CSDV3118P	Containerization and DevOps Lab	0	0	2	1
	CSVT4018P	Cloud Application Development	4	0	0	4
	CSVT4118P	Cloud Application Development Lab	0	0	2	1
	CSVT4019P	Cloud Computing Security and Management	3	0	0	3
	CSVT4119P	Cloud Computing Security and Management Lab	0	0	2	1
Track 4 : Full Stack Development						
	CSFS2003P	Frontend Development	4	0	0	4
	CSFS2101P	Frontend Development Lab	0	0	2	1
	CSFS3005P	Backend Development	4	0	0	4
	CSFS3101P	Backend Development Lab	0	0	2	1
	CSFS3007P	Microservices and Spring-Boot	3	0	0	3
	CSFS3107P	Microservices and Spring-Boot Lab	0	0	2	1
	CSVT4020P	Cloud Computing and Security	4	0	0	4
	CSVT4120P	Cloud Computing and Security Lab	0	0	2	1
	CSDV4012P	Container Orchestration and Security	4	0	0	4
	CSDV4112P	Container Orchestration and Security Lab	0	0	2	1
Track 5 : Cyber Security and Digital Forensics						
	CSSF2014P	Information Technology and Cyber Security	4	0	0	4
	CSSF2114P	Information Technology and Cyber Security Lab	0	0	2	1
	CSSF3026P	Ethical Hacking & Penetration Testing	4	0	0	4
	CSSF3110P	Ethical Hacking & Penetration Testing Lab	0	0	2	1
	CSSF3027P	Network Security	4	0	0	4
	CSSF3127P	Network Security Lab	0	0	2	1
	CSSF4015P	Digital Forensics	4	0	0	4
	CSSF4115P	Digital Forensics Lab	0	0	2	1
	CSSF4017P	OS, Application & Cloud Security	3	0	0	3
	CSSF4117P	OS, Application & Cloud Security Lab	0	0	2	1
Track 6 : Big Data						
	CSBD2010P	Big Data Overview and Ingestion	4	0	0	4
	CSBD2110P	Big Data Overview and Ingestion Lab	0	0	2	1
	CSBD3015P	Big Data Storage and Analysis	4	0	0	4

	CSBD3115P	Big Data Storage and Analysis Lab	0	0	2	1
	CSBD3016P	Big Data Processing – Disk based and In Memory	4	0	0	4
	CSBD3116P	Big Data Processing Lab	0	0	2	1
	CSBD4008P	Stream Processing	4	0	0	4
	CSBD4101P	Stream Processing Lab	0	0	2	1
	CSBD4009P	Big Data Search and Security	3	0	0	3
	CSBD4109P	Big Data Search and Security Lab	0	0	2	1
Track 7 : Data Science						
	CSDS2001P	Fundamentals of Data Science	4	0	0	4
	CSDS2101P	Fundamentals of Data Science Lab	0	0	2	1
	CSDS3001P	Data Visualization and Interpretation	4	0	0	4
	CSDS3101P	Data Visualization and Interpretation Lab	0	0	2	1
	CSDS3002P	Machine Learning and Deep Learning	4	0	0	4
	CSDS3102P	Machine Learning and Deep Learning Lab	0	0	2	1
	CSEG4034P	Computational Linguistic and Natural Language Processing	4	0	0	4
	CSEG4134P	Computational Linguistics and Natural Language Processing Lab	0	0	2	1
	CSDS4001P	Generative Artificial Intelligence	3	0	0	3
	CSDS4101P	Generative Artificial Intelligence Lab	0	0	2	1
Track 8: Internet of Things (IoT)						
	CSGG2110P	Introduction to IoT, Sensors and Microcontrollers	4	0	0	4
	CSIS2112P	Introduction to IoT, Sensors and Microcontrollers Lab	0	0	2	1
	CSIS3019P	IoT Network Architecture and Communication Protocols	4	0	0	4
	CSIS3119P	IoT Network Architecture and Communication Protocols Lab	0	0	2	1
	CSIS3020P	Industrial IoT and ARM based Embedded Programming	4	0	0	4
	CSIS3120P	Industrial IoT and ARM based Embedded Programming Lab	0	0	2	1
	CSIS4011P	Single Board Computers and IoT Applications Development	4	0	0	4
	CSIS4111P	Single Board Computers and IoT Applications Development Lab	0	0	2	1
	CSIS4012P	Data Analytics for IoT	3	0	0	3
	CSIS4012P	Data Analytics for IoT Lab	0	0	2	1
Track 9: Graphics & Gaming						
	CSGG2011P	Introduction to Graphics and Animation	4	0	0	4
	CSGG2110P	Introduction to Interactive Design and 3D Animation Lab	0	0	2	1
	CSGG3019P	Game Programming	4	0	0	4
	CSGG3113P	Game Programming Lab	0	0	2	1
	CSGG3020P	Computer Graphics	4	0	0	4
	CSGG3120P	Computer Graphics Lab	0	0	2	1

	CSGG4012P	Augmented and Virtual Reality Development	4	0	0	4
	CSGG4112P	Augmented and Virtual Reality Development Lab	0	0	2	1
	CSGG4013P	Web Programming for Interactive 3D Graphics	3	0	0	3
	CSGG4113P	Web Programming for Interactive 3D Graphics Lab	0	0	2	1

___________ Minor course requirement list

Students from other departments in the university have the option to take a minor degree from the __________ stream. The list of course requirements to obtain a ___________ minor degree is as follows. Total credit for minor requirement is minimum twenty-one.

Mandatory Courses- XX Credits*			
Course Code	**Course Name**	**credit**	**Prerequisite**
*If a student has completed 'n' number of equivalent credits among the above three courses, as a part of major curriculum then 'n' number of credits should be taken extra from the optional courses			
Optional Courses- XX Credits**			
**Equivalent courses in the student's major discipline will not be counted towards minor requirement. Apart from these equivalent courses, the student has to complete optional course requirement			

11.0 List of Electives

11.1 Programme Electives

11.1.1 Specialization (Name of the subjectives)

List of specialization to be included below.

11.1.2 Minor

If all courses from any basket are completed by the student, he/she receives a minor.

11.2 University Electives

11.2.1 Signature Courses- School for Life Courses/ Life Skill Courses

Below is the list of courses offered as Signatory courses.

11.2.2 Exploratory Courses

If the student takes up courses from different baskets, that is regarded as an exploratory course.

List of Exploratory courses to be included below.

11.2.3 Open Elective

List of open elective courses to be included below.

12.0 Course Syllabus/ Course Plans

Sample Course plan template-

SEMESTER I

Course Code	Course name	L	T	P	C
CSEG1126	**Linux Lab**	0	0	4	2
Total Units to be Covered: 12	**Total Contact Hours: 60**				
Prerequisite(s):	Exposure to computer systems.	**Syllabus version: 1.0**			

Course Objectives

1. Identify different Linux distributions available.
2. Explain the functionality of basic file operations and file viewing/editing commands.
3. Utilize basic commands to navigate the file system, create and manage files/directories, and view/edit file content.
4. Analyse complex file and directory operations, such as searching for files based on specific criteria or patterns.
5. Create shell scripts that automate tasks such as printing messages, performing arithmetic operations, and manipulating strings.

Course Outcomes

On completion of this course, the students will be able to:

CO1. Understand the functionality and purpose of different file operations, user management commands, and system information commands in Linux.

CO2. Apply the appropriate commands to navigate the file system, manage files/directories, view/edit files, manage users, and gather system information.

CO3. Analyse and identify potential issues or improvements in shell scripts by examining their logic, structure, and performance.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	1	-	-	-	-	-	-	-	1	2	-
CO 2	-	-	-	-	1	-	-	-	-	-	-	-	1	2	-
CO 3	-	-	-	-	1	-	-	-	-	-	-	-	1	2	-
Average	-	-	-	-	1	-	-	-	-	-	-	-	1	2	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Install virtual player and Linux.

Theory:

History of UNIX, The UNIX philosophy, GUI, Overview of the Linux Operating System, Unix commands, Introduction to VI editor

Lab:

Step 1: Install VMware Workstation 15 Player or VirtualBox by Oracle.

i. VMware Workstation 15 Player is a virtualization software that allows you to run multiple operating systems on a single physical machine. You can download it from the VMware website and follow the installation instructions.
ii. VirtualBox is another popular virtualization software that provides similar functionality. You can download it from the Oracle website and install it on your computer.

Step 2: Download an installation .iso for a Linux distribution like Ubuntu, Fedora, or any other of your choice.

i. An .iso file is a disk image of the Linux distribution you want to install. It contains all the necessary files for the operating system.
ii. You can download the .iso file from the official website of the Linux distribution you prefer. For example, if you want to use Ubuntu, you can visit the Ubuntu website and download the .iso file for the desired version.

Step 3: Install the .iso from the virtual VBox or VMware Workstation.

i. Launch VMware Workstation or VirtualBox on your computer.
ii. Create a new virtual machine by clicking on the "New" or "Create" button, depending on the software you are using.
iii. Follow the on-screen instructions to set up the virtual machine. This includes specifying the name, location, and specifications for the virtual machine, such as the amount of RAM and storage allocated.
iv. During the setup process, you will be prompted to select the installation .iso file. Browse and select the .iso file you downloaded in Step 2.
v. Complete the virtual machine setup by following the remaining on-screen instructions. This may involve selecting the operating system type, configuring network settings, and specifying storage options.
vi. Once the setup is complete, start the virtual machine. It will boot from the .iso file and begin the Linux installation process. vii. Follow the installation prompts and provide any required information, such as username, password, and partitioning options.
viii. Once the installation is complete, you will have a fully functional Linux virtual machine running within VMware Workstation or VirtualBox.

Experiment 2: Practice some basic commands on Linux

Theory: Overview of file systems in Linux, File permissions and ownership, modifying file permissions and ownership, Creating and editing files using text editors

Lab:

i. Basic Navigation Commands: Teach essential commands for navigating the file system, such as:

ls (list): Display the contents of a directory.

cd (change directory): Move between directories.

Pwd (print working directory): Show the current directory.

mkdir (make directory): Create new directories.

rmdir (remove directory): Remove empty directories.

ii. File Operations: Introduce fundamental file operations using the following commands: touch: Create new files. cp (copy): Copy files and directories. mv (move): Move or rename files and directories.

rm (remove): Delete files and directories.

iii. File Viewing and Editing: Introduce commands for viewing and editing files, such as: cat: Display file contents. less or more: View files with pagination. head and tail: Show the beginning or end of a file.

nano or vim: Basic text editors.

iv. User Management: Discuss commands for user management tasks, including: whoami: Display the current user. who: Show users currently logged in. passwd: Change the password for the current user.

sudo (superuser do): Execute commands with administrative privileges.

v. System Information: Introduce commands for gathering system information, such as: uname: Display system information. df (disk free): Show disk space usage. top or htop: Monitor system processes. history: View command history.

Experiment 3: Files and Directories commands

Theory: File Manipulation Commands, File Compression and Archiving, File Searching, File System Navigation and Management, File Transfer

Lab:

i. Working with Files: touch: Create an empty file or update the access/modify timestamps of an existing file. cp: Copy files and directories.

mv: Move or rename files and directories. rm: Remove files and directories. cat: Concatenate and display the contents of a file. less: Display the contents of a file one page at a time. head: Display the first few lines of a file. tail: Display the last few lines of a file.

ii. File Permissions and Ownership:

Explain the ls -l command to display detailed file information, including permissions and ownership.

Discuss the three sets of permissions: owner, group, and others.

Explain the chmod command to modify file permissions.

Discuss the chown and chgrp commands to change file ownership and group.

iii. Advanced File and Directory Operations: find: Search for files and directories based on various criteria. grep: Search for specific patterns within files.

tar: Archive files and directories into a single file. gzip/gunzip: Compress and decompress files.

ln: Create hard and symbolic links.

Experiment 4: Shell Programming

Theory: Introduction to BASH shell scripting, Basics of Shell Scripting, Types of Shell, Shell variable, Shell Keywords, Basic Operator, Positional Parameters

Lab:

i. Write a simple shell script that prints "Hello, World!" when executed.
ii. Create a script that prompts the user to enter their name and then displays a personalized greeting.
iii. Write a script that takes two numbers as input and performs various arithmetic operations like addition, subtraction, multiplication, and division.
iv. Create a script that asks the user to enter their age and displays a message based on whether they are eligible to vote or not.

Experiment 5: Shell Programming

Theory: Command Line Argument, Array, Conditional Statements, Decision Making,

Lab:

i. Write a script that takes a number as input and checks whether it is a prime number or not.
ii. Write a script that calculates the sum of the digits of a given number.
iii. Create a script that checks whether a given number is an Armstrong number or not.

Experiment 6: Shell Programming

Theory: Shell Loops, Loop control, IO Redirections, Shell Function, Regular Expressions, Script Debugging and Troubleshooting

Lab:

i. Write a script that checks whether a given number is a palindrome or not. A palindrome number reads the same backward as forward.
ii. Write a script that calculates the greatest common divisor (GCD) and the least common multiple (LCM) of two given numbers.
iii. Create a script that takes multiple numbers as input and sorts them in ascending or descending order.

Experiment 7: Shell Programming

Theory: Introduction to Processes, Process states, and process hierarchy, Process Management Commands: Viewing and monitoring running processes, Terminating or killing processes, Process Prioritization and Scheduling.

Lab:

i. Write a script that takes a filename as input and checks if it exists. If the file exists, display its content; otherwise, prompt the user to create the file.
ii. Create a script that prints the numbers from 1 to 10 using a loop.
iii. Write a script that takes a filename as a command line argument and counts the number of lines, words, and characters in that file.
iv. Create a script that defines a function to calculate the factorial of a given number and call that function with different inputs.

Experiment 8: Shell Programming

Theory: Process Control and Signals, Process Monitoring and Resource Usage, Process

Communication, Process Synchronization, Background Processes and Job Control, System Monitoring and Logging

Lab:

i. Write a script that checks the file permissions of a given file and displays whether it is readable, writable, or executable by the current user.
ii. Create a script that prompts the user to enter a string and then performs operations like string length, string concatenation, and string comparison.
iii. Write a script that searches for a specific pattern in a given file and displays the matching lines.
iv. Create a script that displays various system information like the current date and time, logged-in users, system uptime, etc.

Experiment 9: Shell Programming

Theory: System Performance Monitoring, System Security and User Management.

Lab:

i. Write a script that renames all files in a directory by adding a prefix or suffix to the filenames.
ii. Create a script that searches for files in a specified directory and its subdirectories, based on certain criteria like file extension or file size.
iii. Write a script that generates the Fibonacci series up to a given number, using loops or recursive functions.

Experiment 10: Shell Programming

Theory: Writing modular and reusable code, Script optimization

Lab:

i. Write a script that takes a string as input and calculates its length. ii. Create a script that takes a string as input and prints its reverse. iii. Write a script that prompts the user to enter two strings and concatenate them together.

Experiment 11: Shell Programming

Theory: Interacting with Users: Interactive shell scripts, Parsing and Processing Data Formats, Interacting with Databases

Lab:

i. Write a script that takes a sentence as input and splits it into individual words.
ii. Create a script that checks whether a given string is a palindrome or not.

Experiment 12: Building a Rule-Based Expert System using Shell Scripting

Theory:

Process Automation and Job Scheduling: Automating repetitive tasks using shell scripts, System Administration Scripts, Managing services and daemons.

Lab:

Objective: The objective of this lab exercise is to build a simple rule-based expert system using shell scripting. The expert system will provide recommendations based on a set of predefined rules.

Instructions:

i. Create a shell script named "expert_system.sh".
ii. Implement a set of rules using conditional statements (if-elif-else) within the script. Each rule should check for specific conditions and provide a corresponding recommendation.
iii. Example rules:
    a) If the user is experiencing fever, recommend taking a fever reducer medication.
    b) If the user has a sore throat, recommend gargling with warm saltwater.
    c) If the user has a cough and congestion, recommend drinking warm fluids and taking cough syrup.
    d) Feel free to add more rules based on your desired expert system topic.

iv. Prompt the user to input their symptoms.
v. Based on the user's input, evaluate the rules one by one and display the appropriate recommendation(s) for the symptoms identified.
vi. If none of the rules match the user's symptoms, provide a general recommendation or message.
vii. Test the expert system by running the script and providing different sets of symptoms to observe the recommendations.

viii. Modify the rules or add new rules as needed to refine the expert system's behavior.

ix. Document the logic and rules implemented in the script, along with any modifications or additions made.

x. Write a summary report discussing the challenges faced, observations made, and improvements that can be made to enhance the expert system's functionality.

Total Lab hours 60

Textbooks

1. Dayanand Ambawade, and Deven Shah, "Linux Labs and Open Source Technologies", Dreamtech Press, 2014.
2. Paul W Browning, "101 Labs - Linux LPIC1: Includes Linux Essentials", Reality Press Ltd, 2019.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme:

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50	50

Course Code	Course name		L	T	P	C
CSEG1025	Programming in C		3	0	0	3
Total Units to be Covered: 7		Total Contact Hours: 45				
Prerequisite(s):	Elementary knowledge of Computer		Syllabus version: 1.0			

Course Objectives

1. Introduce students to the basic principles and concepts of programming.
2. Develop students' ability to solve programming problems by applying the fundamental concepts of C programming.
3. Demonstrate students how to design and develop structured programs using modular programming techniques.
4. Enable students to apply their C programming skills to develop small-scale applications.

Course Outcomes

On completion of this course, the students will be able to

CO1. Demonstrate a high level of proficiency in writing correct and efficient C code.
CO2. Acquire the skills to debug and troubleshoot C programs efficiently.
CO3. Expertise in designing structured programs using modular programming techniques.
CO4. Understand file and memory management techniques.
CO5. Acquire the ability to apply their C programming skills to develop practical applications.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	3	-	-	-	1	-	-	-	-	-	-
CO 2	-	-	-	-	2	-	-	-		-	-	-	-	-	-
CO 3	-	-	-	-	1	-	-	-		-	-	-	-	-	-
CO4	-	-	-	-	2	-	-	-		-	-	-	-	-	-
CO5	-	-	2	-	2	-	-	-	2	-	-	2	-	-	-

Average	**-**	**-**	.4	**-**	2	**-**	**-**	**-**	.6	**-**	**-**	.4	**-**	**-**	**-**

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to Computing 6 Lecture Hours

Basic computer organization, Evolution of programming languages, Data representation and storage, Basics of programming environment: editors, debuggers, translators, basics of program design and execution, Algorithms, Pseudocode and Flowcharts.

Unit II: C Programming Fundamentals 6 Lecture Hours

Data types and type conversion, variables (declaration vs definition, local vs global), keywords, header files, structure of a C program. Operators: types of operators (arithmetic, relational, logical, bit-wise, increment/decrement, assignment, sizeof, ternary), operator precedence and associativity. Conditional statements: if, else, switch-case, break, continue, goto, label. Loops: for, while and do-while.

Unit III: Array and Function 6 Lecture Hours

Array, Multi-dimensional arrays, Strings, Function, Pass and Return by value, Pass and Return by Reference, Recursion, Scope Rules.

Unit IV: Structures and Pointers 7 Lecture Hours

Structure, typedef, Union, Enum, Bit-Fields, Pointer, Pointer to Arrays, Pointer Arrays, Pointer to Pointers, Address Arithmetic, Pointer to Structures, Pointer to functions, Bit-wise operator.

Unit V: File handling, Memory management 6 Lecture Hours

Data Organization, File Operations. Dynamic Memory Management: Malloc(), Calloc(), Realloc() and Free (), Garbage Collection.

Unit VI: Preprocessor, Macro, Static and Shared Library 8 Lecture Hours

Preprocessor & Directives, Macro, Macro vs Functions, C standard library: stdio.h, ctype.h, stdlib.h, assert.h, stdarg.h, time.h etc., Compilation of a C Program, Static Library, Shared Library.

Unit VII: Multithreading and Optimization **6 Lecture Hours**

Multithreaded programming. Sockets and Asynchronous I/O. Linux Inter Process Communication, Optimization and Debugging. Unit Testing.

Total lecture Hours 45

Textbooks

1. B. W. Kernighan, and D. M. Ritchie, "The C programming language", 2nd Edition, Prentice Hall, 1988.

2. P. J. Deitel, and H. M. Deitel, "C: How to program", 8th Edition, Pearson Education, 2015.

Reference Books

1. B. S. S. Gottfried, "Schaum's Outline of Programming with C", 2nd Edition, McGraw-Hill, 1996.

2. P. V. D. Linden, "Expert C Programming-Deep C Secrets", Pearson Education, 1994.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSEG1125	**Programming in C Lab**		0	0	4	2
Total Units to be Covered: 18		**Total Contact Hours: 60**				
Prerequisite(s):	Elementary knowledge of Computer Systems		**Syllabus version: 1.0**			

Course Objectives

1. Familiarize students with the basic principles and concepts of programming through hands-on practice.
2. Enhance students' problem-solving skills by applying fundamental concepts of C programming to solve various programming challenges.
3. Enable students to learn and apply modular programming techniques to design and develop structured programs effectively.
4. Enable students to apply their C programming skills to develop small-scale applications that demonstrate practical usage of programming concepts.

Course Outcomes

On completion of this course, the students will be able to:

CO1. Demonstrate a high level of proficiency in writing correct and efficient C code.

CO2. Analyse programming problems and devise efficient solutions using fundamental concepts of C programming.

CO3. Develop programs that demonstrate modularity, reusability, and maintainability.

CO4. Incorporate input/output operations, file handling, and error handling in application development.

CO5. Apply problem-solving techniques and critical thinking to address real-world programming scenarios.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	3	-	-	-	1	-	-	-	1	2	-
CO 2	-	-	-	-	2	-	-	-	-	-	-	-	1	2	-
CO 3	-	-	-	-	1	-	-	-	-	-	-	-	1	2	-

CO 4	**-**	**-**	**-**	**-**	2	**-**	**-**	**-**	**-**	**-**	**-**	**-**	1	2	**-**
CO 5	**-**	**-**	2	**-**	2	**-**	**-**	**-**	2	**-**	**-**	2	1	2	**-**
Average	**-**	**-**	.4	**-**	2	**-**	**-**	**-**	.6	**-**	**-**	.4	1	2	**-**

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

List of Experiments

Experiment 1: Installation, Environment Setup and starting with C language

1. Write a C program to print “ Hello World”
2. Write a C Program to print the address in multiple lines (new line).
3. Write a program that prompts the user to enter their name and age.
4. Write a C program to add two numbers, take number from user.

Experiment 2: Operators

1. WAP a C program to calculate the area and perimeter of a rectangle based on its length and width.
2. WAP a C program to Convert temperature from Celsius to Fahrenheit using the formula: F = (C * 9/5) + 32.

Experiment 3.1: Conditional Statements

1. WAP to take check if the triangle is valid or not. If the validity is established, do check if the triangle is isosceles, equilateral, right angle, or scalene. Take sides of the triangle as input from a user.
2. WAP to compute the BMI Index of the person and print the BMI values as per the following ranges. You can use the following formula to compute BMI= weight(kgs)/Height(Mts)*Height(Mts).

	BMI
Starvation	<15
Anorexic	15.1 to 17.5

Underweight	17.6 to 18.5
Ideal	18.6 to 24.9
Overweight	25 to 25.9
Obese	30 to 39.9
Morbidity Obese	40.0 above

3. WAP to check if three points (x1,y1), (x2,y2) and (x3,y3) are collinear or not.
4. According to the gregorian calendar, it was Monday on the date 01/01/01. If Any year is input through the keyboard write a program to find out what is the day on 1st January of this year.
5. WAP using ternary operator, the user should input the length and breadth of a rectangle, one has to find out which rectangle has the highest perimeter. The minimum number of rectangles should be three.

Experiment 3.2: Loops

1. WAP to enter numbers till the user wants. At the end, it should display the count of positive, negative, and Zeroes entered.
2. WAP to print the multiplication table of the number entered by the user. It should be in the correct formatting.   Num * 1 = Num
3. WAP to generate the following set of output.

a.

1

2 3

4 5 6

b.

1

1 1

1 2 1

1 3 3 1

1 4 6 4 1

4. The population of a town is 100000. The population has increased steadily at the rate of 10% per year for the last 10 years. Write a program to determine the population at the end of each year in the last decade.
5. Ramanujan Number is the smallest number that can be expressed as the sum of two cubes in two different ways. WAP to print all such numbers up to a reasonable limit.

   Example of Ramanujan number: 1729

   12^3 + 1^3 and 10^3 + 9^3. for a number L=20(that is limit)

Experiment 4: Variable and Scope of Variable

1. Declare a global variable outside all functions and use it inside various functions to understand its accessibility.
2. Declare a local variable inside a function and try to access it outside the function. Compare this with accessing the global variable from within the function.
3. Declare variables within different code blocks (enclosed by curly braces) and test their accessibility within and outside those blocks.
4. Declare a static local variable inside a function. Observe how its value persists across function calls.

Experiment 5: Array

1. WAP to read a list of integers and store it in a single dimensional array. Write a C program to print the second largest integer in a list of integers.
2. WAP to read a list of integers and store it in a single dimensional array. Write a C program to count and display positive, negative, odd, and even numbers in an array.

3. WAP to read a list of integers and store it in a single dimensional array. Write a C program to find the frequency of a particular number in a list of integers.
4. WAP that reads two matrices A (m x n) and B (p x q) and computes the product A and B. Read matrix A and matrix B in row major order respectively. Print both the input matrices and resultant matrix with suitable headings and output should be in matrix format only. Program must check the compatibility of orders of the matrices for multiplication. Report appropriate message in case of incompatibility.

Experiment 6: Functions

1. Develop a recursive and non-recursive function FACT(num) to find the factorial of a number, n!, defined by FACT(n) = 1, if n = 0. Otherwise, FACT(n) = n * FACT(n-1). Using this function, write a C program to compute the binomial coefficient. Tabulate the results for different values of n and r with suitable messages.
2. Develop a recursive function GCD (num1, num2) that accepts two integer arguments. Write a C program that invokes this function to find the greatest common divisor of two given integers.
3. Develop a recursive function FIBO (num) that accepts an integer argument. Write a C program that invokes this function to generate the Fibonacci sequence up to num.
4. Develop a C function ISPRIME (num) that accepts an integer argument and returns 1 if the argument is prime, a 0 otherwise. Write a C program that invokes this function to generate prime numbers between the given ranges.
5. Develop a function REVERSE (str) that accepts a string argument. Write a C program that invokes this function to find the reverse of a given string.

Experiment 7: Structures and Union

1. Write a C program that uses functions to perform the following operations:
    a. Reading a complex number.
    b. Writing a complex number.
    c. Addition and subtraction of two complex numbers

Note: represent complex number using a structure.

2. Write a C program to compute the monthly pay of 100 employees using each employee_s name, basic pay. The DA is computed as 52% of the basic pay. Gross-salary (basic pay + DA). Print the employees name and gross salary.
3. Create a Book structure containing book_id, title, author name and price. Write a C program to pass a structure as a function argument and print the book details.
4. Create a union containing 6 strings: name, home_address, hostel_address, city, state and zip. Write a C program to display your present address.

Experiment 8: Pointers

1. Declare different types of pointers (int, float, char) and initialize them with the addresses of variables. Print the values of both the pointers and the variables they point to.
2. Perform pointer arithmetic (increment and decrement) on pointers of different data types. Observe how the memory addresses change and the effects on data access.
3. Write a function that accepts pointers as parameters. Pass variables by reference using pointers and modify their values within the function.

Experiment 9: File Handling in C

1. Write a program to create a new file and write text into it.
2. Open an existing file and read its content character by character, and then close the file.
3. Open a file, read its content line by line, and display each line on the console.

Experiment 10: Dynamic Memory Allocation

1. Write a program to create a simple linked list in C using pointer and structure.
2. Write a program to insert item in middle of the linked list.

Experiment 11: Bitwise Operator

1. Write a program to apply bitwise OR, AND and NOT operators on bit level.
2. Write a program to apply left shift and right shift operator.

Experiment 12: Preprocessor and Directives in C

1. Write a program to define some constant variable in preprocessor.
2. Write a program to define a function in directives.

Experiment 13: Macros in C

1. Write a program to define multiple macro to perform arithmetic functions.

Experiment 14: Static Library in C

1. Write a program to create a static library for performing arithmetic functions.
2. Write a program to use static library in other program.

Experiment 15: Shared Library in C

1. Write a program to create a shared library for performing arithmetic functions.
2. Write a program to use shared library in other program.

Experiment 16: Multithreading in C

1. Write a program to print 1-10 numbers five times using multithreading in C.

Experiment 17: Socket Programming in C

1. Write a program to implement socket programming using C.

Experiment 18: Testing and Debugging

1. Write a program and perform testing and debugging on same implementation.

Total Lab hours 60

Textbooks

1. B. W. Kernighan, and D. M. Ritchie, "The C programming language", 2nd Edition, Prentice Hall, 1988.

2. P. J. Deitel, and H. M. Deitel, "C: How to program", 8th Edition, Pearson Education, 2015.

Reference Books

1. B. S. S. Gottfried, "Schaum's Outline of Programming with C", 2nd Edition, McGraw-Hill, 1996.

2. P. V. D. Linden, "Expert C Programming-Deep C Secrets", Pearson Education, 1994.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50	50

Course Code	Course name		L	T	P	C
CSEG1027	Problem Solving		2	0	0	2
Total Units to be Covered: 4		Total Contact Hours: 30				
Prerequisite(s):	10+2 Mathematics		Syllabus version: 1.0			

Course Objectives

1. Learn problem investigation strategies.
2. Understand the tactics for solving different problems.
3. Apply the learnt tactics to typical problems.

Course Outcomes

On completion of this course, the students will be able to

CO1: Identify various problem-solving strategies.

CO2: Infer the potential tactics for problem solving.

CO3: Develop appropriate tactics for solving the posed problems.

CO4: Analyse the solutions towards efficacy.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	3	-	1	-	-	-	-	-	-	-	-	-	-	-
CO 2	2	3	-	1	-	-	-	-	-	-	-	-	-	-	-
CO 3	2	3	-	1	-	-	-	-	-	-	-	-	-	-	-
CO4	2	3	-	1	-	-	-	-	-	-	-	-	-	-	-
Average	2	3	-	1	-	-	-	-	-	-	-	-	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: **7 Lecture Hours**

Thinking on a problem: Problems vs Exercises, the levels of problem solving, problem classification. Theoretical frameworks for problem solving: Polya’s theory, cognitive problem-solving, functional fixedness and mental set, means-end analysis, divergent and convergent thinking, backdrop of theoretical frameworks, and relevant examples

Unit II: **8 Lecture Hours**

Investigating problems: Psychological Strategies, strategies for Getting Started, Methods of Argument, other strategies; drawing pictures.

Heuristics for problem solving heuristics and heuristic reasoning, Symmetry – geometric and algebraic symmetry, the extreme principle, the Pigeonhole principle – basic, intermediate, and advanced, sample problems.

Further tactics for problem solving: Invariants, Coloring proofs, the Box principle, sample problems.

Unit III: **7 Lecture Hours**

Crossover tactics: Graph Theory - Connectivity and Cycles, Eulerian and Hamiltonian Paths, Complex Numbers - Basic Operations, roots of unity, Generating Functions - Recurrence Relations, partitions, problems based on crossover tactics.

Algebra: Polynomials – Polynomial operations, Inequalities – Fundamental ideas, the AM-GM inequality, Massage, Cauchy-Schwarz, and Chebyshev inequalities, problems based on algebraic tactics.

Unit IV: **8 Lecture Hours**

Combinatorics: Partitions and Bijections – counting subsets, The Principle of Inclusion-Exclusion – Count the Complement, PIE with Sets, PIE with Indicator Functions, sample problems based on combinatorics.

Number Theory: Primes and divisibility – The Fundamental Theorem of Arithmetic, GCD, LCM, and the Division Algorithm, Congruence – Fermat's Little Theorem, Number Theoretic Functions – divisor sums, Phi and Mu, Diophantine Equations, sample problems based on number theory.

Puzzles and challenging problems: Algorithmic puzzles and games – weighing problems, e.g., ranking weights, red, white, and blue weights, fair division problems, The tower of Hanoi, Crossing a river, the puzzle of twins, the puzzle of thinking about oneself, the bridge problem, fake coin detection, Handshake problem.

Total lecture Hours 30

Textbooks

1. G. Polya, "How to solve it: A new aspect of mathematical method", 2nd Edition, Princeton university press, 2004.

2. P. Zeitz, "The art and craft of problem solving", 3rd Edition, John Wiley & Sons, 2017.

3. R. Backhouse, "Algorithmic problem solving", John Wiley & Sons, 2011.

Reference Books

1. A. Engel, "Problem-solving strategies", Springer, 2008.

2. M. Gardner, and D. Richards, "The colossal book of short puzzles and problems", W. W. Norton & Company, 2005.

3. D. D. Riley, and K. A. Hunt, "Computational thinking for the modern problem solver", Chapman and Hall/CRC, 2014

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50%	20%	30%	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
SSEN0101	Environment Sustainability & Climate Change (Living Lab)	0	0	2	2
Total Units to be Covered: 9	Total Contact Hours: 30				
Prerequisite(s):	Fundamentals of basic ecology, chemistry and physics	Syllabus version: 1.0			

Course Objectives

1. To develop a critical understanding of the nature, cause and impact of human activities on the environment.
2. Critically engage with concepts of ecosystems, biodiversity and sustainability.
3. Research, analyse, identify problems, develop insights, and frame sustainable solutions to living issues faced by the global and local communities.
4. Learning by doing, engaging, exploring and experimenting.

Course Outcomes

CO1. Understand the concepts of ecology, sustainability, climate change and environment related to everyday life.

CO2. Distinguish and relate different types of biodiversity and natural resources and their impact on sustainable development.

CO3. Analyse various aspects of environment and adopt eco-friendly technologies to facilitate conservation and regeneration of natural resource.

CO4. Build environmental awareness through a wide range of curricular and co-curricular activities at the University and later in a professional/vocational practice.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	-	-	-	-	-	-	-	2	-	-	-	-	-
CO 2	1	-	-	-	-	-	-	-	-	2	-	-	-	-	-
CO 3	1	-	-	-	-	-	-	-	-	2	-	-	-	-	-
CO4	1	-	-	-	-	-	-	-	-	2	-	-	-	-	-

Average	1	-	-	-	-	-	-	-	-	2	-	-	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**
3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Humans and the Environment **2 lecture hours**

The man-environment interaction: Humans as hunter-gatherers; Mastery of fire; Origin of agriculture; Emergence of city-states; Great ancient civilizations and the environment; Middle Ages and Renaissance; Industrial revolution and its impact on the environment; Population growth and natural resource exploitation; Global environmental change.

The emergence of environmentalism: Anthropocentric and eco-centric perspectives (Major thinkers); The Club of Rome- Limits to Growth; UN Conference on Human Environment 1972; World Commission on Environment and Development and the concept of sustainable development; Rio Summit and subsequent international efforts.

Unit II: Natural Resources and Sustainable Development **3 lecture hours**

Overview of natural resources: Definition of resource; Classification of natural resources- biotic and abiotic, renewable and non-renewable.

Biotic resources: Major type of biotic resources- forests, grasslands, wetlands, wildlife and aquatic (fresh water and marine); Microbes as a resource; Status and challenges.

Water resources: Types of water resources- fresh water and marine resources; Availability and use of water resources; Environmental impact of over-exploitation, issues and challenges; Water scarcity and stress; Conflicts over water.

Soil and mineral resources: Important minerals; Mineral exploitation; Environmental problems due to extraction of minerals and use; Soil as a resource and its degradation.

Energy resources: Sources of energy and their classification, renewable and non-renewable sources of energy; Conventional energy sources- coal, oil, natural gas, nuclear energy; Non-conventional energy sources- solar, wind, tidal, hydro, wave, ocean thermal, geothermal, biomass, hydrogen and fuel cells; Implications of energy use on the environment.

Introduction to sustainable development: Sustainable Development Goals (SDGs)-targets and indicators, challenges and strategies for SDGs.

Unit III: Environmental Issues: Local, Regional and Global 3 lecture hours

Environmental issues and scales: Concepts of micro-, meso-, synoptic and planetary scales; Temporal and spatial extents of local, regional, and global phenomena.

Pollution: Impact of sectoral processes on Environment, Types of Pollution- air, noise, water, soil, municipal solid waste, hazardous waste; Transboundary air pollution; Acid rain; Smog.

Land use and Land cover change: land degradation, deforestation, desertification, urbanization.

Biodiversity loss: past and current trends, impact.

Global change: Ozone layer depletion; Climate change.

Unit IV: Conservation of Biodiversity and Ecosystem 4 lecture hours

Biodiversity and its distribution: Biodiversity as a natural resource; Levels and types of biodiversity; Biodiversity in India and the world; Biodiversity hotspots; Species and ecosystem threat categories.

Ecosystems and ecosystem services: Major ecosystem types in India and their basic characteristics- forests, wetlands, grasslands, agriculture, coastal and marine; Ecosystem services- classification and their significance.

Threats to biodiversity and ecosystems: Land use and land cover change; Commercial exploitation of species; Invasive species; Fire, disasters and climate change.

Major conservation policies: in-situ and ex-situ conservation approaches; Major protected areas; National and International Instruments for biodiversity conservation; the role of traditional knowledge, community-based conservation; Gender and conservation.

Unit V: Environment Pollution and Health 4 lecture Hours

Understanding pollution: Production processes and generation of wastes; Assimilative capacity of the environment; Definition of pollution; Point sources and non-point sources of pollution.

Air pollution: Sources of air pollution; Primary and secondary pollutants; Criteria pollutants- carbon monoxide, lead, nitrogen oxides, ground-level ozone, particulate matter and sulphur dioxide; Other important air pollutants- Volatile Organic compounds (VOCs), Peroxyacetyl Nitrate (PAN), Polycyclic aromatic hydrocarbons (PAHs) and Persistent organic pollutants (POPs); Indoor air pollution; Adverse health impacts of air pollutants; National Ambient Air Quality Standards.

Water pollution: Sources of water pollution; River, lake and marine pollution, groundwater pollution; water quality Water quality parameters and standards; adverse health impacts of water pollution on human and aquatic life.

Soil pollution and solid waste: Soil pollutants and their sources; Solid and hazardous waste; Impact on human health.

Noise pollution: Definition of noise; Unit of measurement of noise pollution; Sources of noise pollution; Noise standards; adverse impacts of noise on human health.

Thermal and Radioactive pollution: Sources and impact on human health and ecosystems.

Unit VI: Climate Change Impact Adaptation and Mitigation 4 lecture Hours

climate change from greenhouse gas emissions– past, present and future; Projections of global climate change with special reference to temperature, rainfall, climate variability and extreme events; Importance of 1.5 °C and 2.0 °C limits to global warming; Climate change projections for the Indian sub-continent.

Impacts, vulnerability and adaptation to climate change: Observed impacts of climate change on ocean and land systems; Sea level rise, changes in marine and coastal ecosystems; Impacts on forests and natural ecosystems; Impacts on animal species, agriculture, health, urban infrastructure; the concept of vulnerability and its assessment; Adaptation vs. resilience; Climate-resilient development; Indigenous knowledge for adaptation to climate change.

Mitigation of climate change: Synergies between adaptation and mitigation measures; Green House Gas (GHG) reduction vs. sink enhancement; Concept of carbon intensity, energy intensity and carbon neutrality; National and international policy instruments for mitigation, decarbonizing pathways and net zero targets for the future; Energy efficiency measures; Renewable energy sources; Carbon capture and storage, National climate action plan and *Intended Nationally Determined Contributions* (INDCs); Climate justice.

Unit VII: Environment Management 4 Lecture Hours

Introduction to environmental laws and regulation: Constitutional provisions- Article 48A, Article 51A (g) and other derived environmental rights; Introduction to environmental legislations on the forest, wildlife and pollution control.

Environmental management system: ISO 14001

Life cycle analysis; Cost-benefit analysis

Environmental audit and impact assessment; Environmental risk assessment

Pollution control and management; Waste Management- Concept of 3R (Reduce, Recycle and Reuse) and sustainability; Ecolabeling /Ecomark scheme

Unit VIII: Environment Treaties and Legislation 4 Lecture Hours

1. An overview of instruments of international cooperation; bilateral and multilateral agreements; conventions and protocols; adoption, signature, ratification and entry into force; binding and non-binding measures; Conference of the Parties (COP)

2. Major International Environmental Agreements: Convention on Biological Diversity (CBD); Cartagena Protocol on Biosafety; Nagoya Protocol on Access and Benefit-sharing; Convention on International Trade in Endangered Species of 6ild Flora and Fauna (CITES); Ramsar Convention on Wetlands of International Importance; United Nations Convention to Combat Desertification (UNCCD); Vienna Convention for the Protection of the Ozone Layer; Montreal Protocol on Substances that Deplete the Ozone Layer and the Kigali Amendment; Basel Convention on the Control of Transboundary Movements of Hazardous Wastes and their Disposal; Rotterdam Convention on the Prior Informed Consent Procedure for Certain Hazardous Chemicals and Pesticides in International Trade; Stockholm Convention on Persistent Organic Pollutants; Minamata Convention on Mercury; United Nations Framework

Convention on Climate Change (UNFCCC); Kyoto Protocol; Paris Agreement; India's status as a party to major conventions

3. Major Indian Environmental Legislations: The Wild Life (Protection) Act, 1972; The Water (Prevention and Control of Pollution) Act, 1974; The Forest (Conservation) Act, 1980; The Air (Prevention and Control of Pollution) Act, 1981; The Environment (Protection) Act, 1986; The Biological Diversity Act, 2002; The Scheduled Tribes and Other Traditional Forest Dwellers (Recognition of Forest Rights) Act, 2006; Noise Pollution (Regulation and Control) Rules, 2000; Industry-specific environmental standards; Waste management rules; Ramsar sites; Biosphere reserves; Protected Areas; Ecologically Sensitive Areas; Coastal Regulation Zone; Status phase-out of production and consumption of Ozone Depleting Substances by India; National Green Tribunal; Some landmark Supreme Court judgements

Major International organisations and initiatives: United Nations Environment Programme (UNEP), International Union for Conservation of Nature (IUCN),World Commission on Environment and Development (WCED), United Nations Educational, Scientific and Cultural Organization (UNESCO), Intergovernmental Panel on Climate Change (IPCC), and Man and the Biosphere (MAB) programme.

Unit IX: Living Lab Case Studies and Field Work 2 Lecture Hours

The students are expected to be engaged in some of the following or similar identified activities:

1. Discussion on one national and one international case study related to the environment and sustainable development.
2. Field visits to identify local/regional environmental issues, make observations including data collection and prepare a brief report.
3. Documentation of campus biodiversity.
4. Campus environmental management activities such as solid waste disposal, water management, and sewage treatment

Total lecture Hours 30

Textbooks

1. Michael Herbert Fisher, An environmental history of India: from earliest times to the twenty-first century. Cambridge, United Kingdom; New York, Ny: Cambridge University Press, 2018.
2. D. R. Headrick, Humans versus nature: a global environmental history. New York, Ny: Oxford University Press, 2020.
3. William P.Cunningham and Mary A. Cunningham Environmental Science: A Global Concern, Publisher (Mc-Graw Hill, USA), 2015
4. Gilbert M. Masters and W. P, An Introduction to Environmental Engineering and Science, Ela Publisher (Pearson), 2008
5. R. Rajagopalan, Environmental Studies: From Crisis to Cure. India: Oxford University Press.University Grants Commission 11 **(**2011).

Reference Books

1. William P. Cunningham and Mary A. Cunningham Environmental Science: A global concern, Publisher (Mc-Graw Hill, USA), 2015.
2. Shonil Bhagwat, Conservation and Development in India: Reimagining Wilderness, Earthscan Conservation and Development, Routledge , (Editor) (2018).
3. G. M. Masters, & W. P Ela, *Introduction to environmental engineering and science* (No. 60457). Englewood Cliffs, NJ: Prentice Hall (2008).
4. G. T. Miller, & S. Spoolman, Environmental Science. Cengage Learning (2015).
5. Central Pollution Control Board Web page for various pollution standards. https://cpcb.nic.in/standards/
6. V. K. Ahluwalia, *Environmental Pollution, and Health*. The Energy and Resources Institute (TERI), 2015.**University Grants Commission 13**
7. A. Denle, H. Azadi, J. Arbiol Global assessment of technological innovation for climate change adaptation and mitigation in developing world, Journal of Environmental Management, 2015, 161 (15): 261-275.
8. Richard A. Marcantonio, Marc Lame, Environmental Management: Concepts and Practical Skills. Cambridge University Press,2022, **University Grants Commission 15**
9. UNEP (2007) Multilateral Environmental Agreement Negotiator's Handbook, University of Joensuu, ISBN 978-952-458-992-5

10. Ministry of Environment, Forest and Climate Change (2019) A Handbook on International Environment Conventions & Programmes. https://moef.gov.in/wp-content/uploads/2020/02/convention-V-16-CURVE-web.pdf

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continous assessment

Components	**Continous Assesment**
Weightage (%)	100%

Course Code	Course name		L	T	P	C
SLLS0101	Living Conversations		2	0	0	2
Total Units to be Covered: 8		Total Contact Hours: 30				
Prerequisite(s):			Syllabus version: 1.0			

Course Objectives

The objectives of this course are:

1. Encourage critical self-reflection to develop empathy and clarity of expression for the exchange of individual and organizational ideas and information.
2. Enable qualities of deep listening and clear and concise communication skills.
3. Apply and practice varied platforms and tools of communication both formal and informal.
4. Appreciate and practice collaborative communication in a given environment and context.

Course Outcomes

CO1 Understand the importance of being a n empathetic communicator and the role of clarity in the expression.

CO2 Use and Analyze communication strategies and theories, as well as how they are practiced in the professional and social environment.

CO3 Demonstrate appropriate tools to improve one's ability to express, listen, and understand people in a given situation and context.

CO4 Articulate responses both verbally and non-verbally for group and individual work undertaken by self and by others, in the execution of the project/coursework.

CO5 Practice and Employ communication skills to engage ethically in independent and life-long learning in the broader context

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3

CO 1	-	-	-	-	-	1	1	-	3	3	1	1	-	-	-
CO 2	-	-	-	-	-	1	1	-	2	3	1	1	-	-	-
CO 3	-	-	-	-	-	-	1	1	1	3	1	1	-	-	-
CO4	-	-	-	-	-	1	1	1	2	3	-	1	-	-	-
CO5	-	-	-	-	-	-	-	1	3	1	-	3	-	-	-
Average	-	-	-	-	-	0.6	0.8	0.6	2.2	2.6	0.6	1.4	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Basics of Communication **4 Lecture Hours**

Introduction to the course, Importance, use and its application in life (personal as well as professional), Basics of Communication with Practical Examples (need – principles - process – model), Introducing Types of Communication (Verbal & non-verbal), Types of non-verbal communication & its importance in overall communication.

Unit II: Setting Communication Goals & Avoiding Breakdowns **6 Lecture Hours**

Communication goals, creating value in conversations, Internal & external factors impacting our conversations, Communication breakdowns, and how to address them.

Unit III: Listening for Improved Understanding **2 Lecture Hours**

Importance, Active & Passive listening, Barriers, Benefits, Features & Examples of Active Listening, Verbal and non-verbal signs of active listening skills, Tools & Tips for Practicing Active Listening

Unit IV: Non-verbal Communication **6 Lecture Hours**

Introduction to Non-Verbal Communication, Areas of nonverbal communication, Functions and influence of nonverbal communication, Basics of Body Language, Common Gestures, Body Language Mistakes, Improving Your Body Language, Voice Modulation.

Unit V: Public Speaking and Presentation Skills **4 Lecture Hours**

Public Speaking vs. Presentations, The Essentials of Effective Presentation, Content Development, Confidence Building, Best Practices, Virtual Presentation.

Unit VI: Communication Styles **2 Lecture Hours**

Recognizing your style and the styles of others, closing communication gaps, and being flexible without compromising one's identity.

Unit VII: Cross-cultural Communication: Navigating beyond boundaries

2 Lecture Hours

Developing greater sensitivity to cultural differences, Building greater accountability and trust on virtual teams, Uncovering hidden assumptions, and Recognizing filters in oneself and others.

Unit VIII: PROJECT WORK and Submission **4 Lecture Hours**

Total lecture Hours 30

Textbooks

1. O. Hargie, The Handbook of Communication Skills, 4th ed. Abingdon, Oxon; New York, Ny: Routledge, 2019.
2. P. A. Andersen and L. K. Guerrero, Handbook of communication and emotion: research, theory, applications, and contexts. San Diego: Academic Press, 1998.
3. T. Bretag, J. Crossman, and Sarbari Bordia, Communication skills. North Ryde, N.S.W.: Mcgraw-Hill Australia, North Ryde, N.S.W, 2008.
4. I. Tuhovsky, The Science of Effective Communication: Improve Your Social Skills and Small Talk, Develop Charisma and Learn How to Talk to Anyone. Positive Coaching LLC.
5. H. Murphy, Effective business communication. Mcgraw-Hill Co, 1999.

Reference Books

1. Bordia Crossman, Bretag. Communication Skills. Tata Macgraw Hill.
2. Tuhovsky, Ian. The Science of Effective Communication.
3. Murphy, Herta, Thomas, Jane P. Effective Business Communication. Tata MacGraw Hill

Modes of Evaluation: Project + Classroom Participation + Course Quiz

Examination Scheme: Continous assessment

Components	**Continous Assesment**
Weightage (%)	100%

Course Code	Course name	L	T	P	C
MATH1059	Advanced Engineering Mathematics – 1	3	1	0	4
Total Units to be Covered: 5	Total Contact Hours: 60				
Prerequisite(s):	Basic Mathematics (10+2 level)	Syllabus version: 1.0			

Course Objectives

The course aims to

1. Provide both theoretical as well as practical use of differential and integral calculus.
2. Employ vector analysis and vector calculus for modeling physical and engineering problems.
3. Develop capability and skill set to model situations governed by linear and nonlinear differential equations.

Course Outcomes

On completion of this course, the students will be able to

CO1. Understand the power of differential calculus for gaining insight into problems of interest.

CO2. Recognise the potiential of integral calculas to gain understanding of interesting real world challenges

CO3. Explore the concept of vector-valued functions as applied in engineering applications.

CO4. Formulate and analyze mathematical models of a variety of real-world problems.

CO5. Develop and visualize solutions of nonlinear mathematical models.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	3	1	-	1	-	-	-	-	-	-	-	-	-	-
CO 2	3	3	1	-	1	-	-	-	-	-	-	-	-	-	-
CO 3	3	3	2	-	1	-	-	-	-	-	-	-	-	-	-
CO 4	3	3	3	-	3	-	-	-	-	-	-	-	-	-	-

CO 5	3	3	3	-	3	-	-	-	-	-	-	-	-	-	-
Average	3	3	2	-	1.8	-	-	-	-	-	-	-	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit 0: Motivation **1 Lecture Hours**

Why study this course? Application of Calculus and Differential equations in addressing real-world problems.

Unit I: Differential Calculus **12 Lecture Hours**

Review: Functions and their graphs, polynomial, exponential, and logarithmic functions, Remainder and Factor theorems of polynomials.Limits, Continuity and Differentiability, Rolle's theorem, Lagrange's and Cauchy mean value theorems, Successive differentiation, Leibnitz's theorem, Taylor's series without proof, Lagrange's form of remainder, Functions of several variables, Partial differentiation, Euler's theorem, Jacobian, Maxima and minima. Recap of Unit-I.

Unit II: Integral Calculus **12 Lecture Hours**

Definite integrals and properties, Double integrals: Cartesian and Polar co-ordinates, Cylindrical and Spherical coordinates, Change of order of integration, Change of variables, Triple integrals, Gamma, Beta functions and their properties, Applications of Calculus: Real-world problems. Recap of Unit-II.

Unit III: Vector Algebra and Calculus **12 Lecture Hours**

Motivation, Vector algebra, Scalar and vector point functions, Vector Differentiation, Gradient, Divergence and Curl, Vector Integration: Line integral, Surface integral, Volume integral, Application of integrals: Work, Circulation and Flux, Green, Gauss, and Stokes theorems (without proof) and their applications. Recap of Unit-III.

Unit IV: Linear Differential Equations with Visualizations 13 Lecture Hours

Motivation: Linearity vs non-linearity, Linear superposition principle, First-order linear differential equations: Exact differential equations. Integrating Factors. Applications of first-order linear equations: Electric circuits, Radioactive decay, Population growth. Second and Higher order linear differential equations: Solution of homogeneous and non-homogeneous equations with constant coefficients, Wronskian, Solution of second-order differential equation by variation of parameters, Autonomous vs Non-autonomous systems, Applications: Harmonic oscillator, Electric Circuits. Visualization Tool: GeoGebra https://www.geogebra.org/t/differential-equation. Recap of Unit-IV.

Unit V: Non-Linear Differential Equations with Visualizations 10 Lecture Hours

Brief history of interdisciplinary studies of nonlinear systems, Pendulum equations, Phase portraits, Linearization around equilibrium, Nonlinear models: Logistic equation, Bass model of diffusion of innovation, Lotka-Volterra predator-prey model, Epidemic models-SIR, Application to the nerve impulse. Visualization Tool: GeoGebra https://www.geogebra.org/t/differential-equation. Recap of Unit V.

Total lecture Hours 60

Textbooks

1. G. B. Thomas, and R. L. Finney, "Calculus and Analytical Geometry", 9th Edition, Pearson Education India, 2010.

2. G. James, and P. Dyke, "Advanced Modern Engineering Mathematics", 5th Edition, Pearson Education, 2018.

3. E. Kreyszig, "Advanced Engineering Mathematics", 10th Edition, J. Wiley and Sons, 2023.

Reference Books

1. T. M. Apostol, "Mathematical Analysis", 2nd Edition, Narosa, 2002.

2. M. Braun, "Differential Equations and their Applications", 4th Edition, Springer, 1993.

3. S. H. Strogatz, "Nonlinear Dynamics and Chaos", 2nd Edition, CRC Press, 2014.

4. R. H., Enns, "It’s a Nonlinear World", Springer, 2010.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
PHYS1032	Physics for Computer Engineers	4	0	0	4
Total Units to be Covered: 4		Total Contact Hours: 60			
Prerequisite(s):	12th Level Physics	Syllabus version: 1.0			

Course Objectives

1. To demonstrate the principles of LASER and its applications in holography as well as in fiber-optic communications.
2. To determine gradient of scalar fields and divergence & curl vector fields.
3. To develop understanding of electromagnetics, which forms the basis of several contemporary communication systems such as fiber optics communication and it, is also a prerequisite for forthcoming semesters.
4. To utilize fundaments of quantum mechanics in various areas of Material Science and engineering.
5. To understand and apply semiconductor materials in various applications.

Course Outcomes

On completion of this course, the students will be able to

CO1. Understand the significance of lasers and its application in holography and optical fiber communication.

CO2. Illustrate the electric field for different charge geometries.

CO3. Outline the magnetic field due to different current geometries.

CO4. Utilize the fundamentals of Quantum Mechanics and analyse the behaviour of particle in a box.

CO5. Apply various applications of semiconductor materials in different instruments.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	1	-	-	-	-	-	-	-	-	-	1	-	-	-

CO 2	3	1	-	-	-	-	-	-	-	-	-	1	-	-	-
CO 3	2	3	-	-	-	-	-	-	-	-	-	1	-	-	-
CO 4	3	2	-	-	-	-	-	-	-	-	-	1	-	-	-
CO 5	3	2	-	-	-	-	-	-	-	-	-	1	-	-	-
Average	2.8	1.8	-	-	-	-	-	-	-	-	-	1	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Lasers & Fibre Optics **14 Lecture Hours**

Introduction, Spontaneous and Stimulated emission of radiation, Relation b/w Einstein's A and B coefficients, Population inversion & types of pumping, Main components of a Laser, Construction & working of Ruby Laser and its applications, Construction & working of Helium-Neon laser and its applications. Holography: Elementary idea of holography and constructive and reconstructive of holography.

Fundamental ideas about optical fiber, Types of fibers, Acceptance angle and cone, Numerical aperture, Propagation mechanism and communication in optical fiber, Attenuation and losses.

Unit II: Electro-Magnetics **16 Lecture Hours**

Electro-statics: Coordinate systems, Del operator, Gradient, Divergence, Divergence Theorem, Stoke's Theorem, Introduction to electrostatics, calculation of electric field, potential and energy due to charge distribution by vector approach, Gauss law electric flux density. Polarization in Dielectrics, Bound charges, Dielectric Constant and strength, Continuity equation and relaxation time Boundary Conditions.

Magneto-statics: Introduction, Biot-Savart's law, Ampere's Circuit Law; Applications, Magnetic flux density, Faraday's Law, Transformer and motional EMF. Displacement current, Maxwell's Equations in Final form.

Unit III: Quantum Mechanics **15 Lecture Hours**

Introduction to Quantum Mechanics, photoelectric effect, Compton Effect, Pair production & Annihilation, Wave particle duality, De Broglie waves, Davisson Germer experiment, phase and group velocities and their relations, Thought experiment-Heisenberg’s Gamma ray microscope, Uncertainty principle and its applications, Wave function and its interpretation, Normalization, Schrodinger time independent & dependent wave equations, Particle in a 1-D box; generalization to 3-D box.

Unit IV: Semiconductor Physics **15 Lecture Hours**

P and N type semiconductors, Energy Level Diagram, Conductivity and Mobility, Concept of Drift velocity, Hall effect, Barrier Formation in PN Junction Diode, Static and Dynamic Resistance, Current Flow Mechanism in Forward and Reverse Biased Diode, Avalanche breakdown, Zener breakdown, Two-terminal Devices and their Applications: Half-wave Rectifiers, Full-wave Rectifiers, Ripple Factor and Rectification Efficiency, Zener Diode and Voltage Regulation, Principle and structure of LED, Photodiode and Solar Cell

Total lecture Hours 60

Textbooks

1. H. K. Malik, A. K. Singh, "Engineering Physics", 2nd Edition, McGraw Hill Education, 2017.
2. A. Beiser, "Concepts of Modern Physics", 4th Edition, McGraw Hill Education, 2018.

Reference Books

1. D. J. Griffith, "Introduction to Electromagnetics", 4th Edition, Cambridge University Press, 2020.
2. A. Ghatak, "Optics", 7th Edition, McGraw Hill Education, 2020.
3. V. Sahni, and D. Goswami, "Nano Computing", McGraw Hill Education Asia Ltd., 2008.
4. M. N. O. Sadiku, "Elements of Electromagnetics", 3rd Edition, Oxford University Press, 2020.

5. C. T. Bhunia, "Introduction to Quantum Computing", New Age International Publishers, 2010.

6. S. M. Sze, "Semiconductor Devices: Physics and Technology", 2nd Edition, John Wiley & Sons, 2001.

7. S. Salivahanan, and N. S. Kumar, "Electronic Devices & circuits", 5th Edition, McGraw Hill, 2022.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
PHYS1132	Physics Lab for Computer Engineers		0	0	2	1
Total Units to be Covered: 10	Total Contact Hours: 30					
Prerequisite(s):	Basic knowledge on practical Physics (12th level) for understanding and performing experiments.	Syllabus version: 1.0				

Course Objectives

1. To impart hand-on skills in performing experiments, data acquisition and interpretation of the data.
2. To design the circuits and study about various experimental procedures involved.
3. Significance of the experimental results to understand and verify theoretical formulation and prediction.
4. To develop curiosity and creative ability through experimentation and investigation based on the virtual experiments.

Course Outcomes

On completion of this course, the students will be able to

CO1: Demonstrate the dual nature of light by verifying the various phenomena associated with it.

CO2: Apply the concepts of electromagnetics to study the various electrical and magnetic properties of Materials.

CO3: Evaluate and compare the universal constants by using the principle of modern Physics.

CO4: Design virtual Physics based experiments to illustrate the Photoelectric Effect.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 2	-	3	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 3	-	-	-	3	-	-	-	-	-	-	-	-	-	-	-
CO4	-	-	3	-	2	-	-	-	-	-	-	-	-	-	-

Average	**-**	1.5	.75	.75	.5	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**	**-**

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1:

To determine the frequency of AC mains by using a sonometer.

Experiment 2:

To study the Hall effect and hence determine the Hall coefficient (R_h) and carrier density (n) of a given semiconductor material.

Experiment 3:

(a) To study the induced emf as a function of velocity of the magnet passing through the coil (Faraday's Law).

(b) To study the charge delivered due to electromagnetic induction.

Experiment 4:

To study the variation of magnetic field with distance along the axis of a current carrying circular coil and hence estimate the radius of the coil.

Experiment 5:

To plot the characteristics of photocurrent vs voltage at different frequency.

Experiment 6:

To determine the Numerical Aperture of an optical fibre and study about the bending losses.

Experiment 7:

To study the laser beam diffraction.

Experiment 8:

Study of both the current - voltage characteristic and the power curve to find the maximum power point (MPP) and efficiency of a solar cell.

Experiment 9:

To find the Planck’s constant by using LEDs.

Experiment 10:

To determine the energy band gap of a given semiconductor by using Four-Probe Method.

Total Lab hours 30

Textbooks

1. H. Singh, and P. S. Hemne, "Practical Physics", S. Chand & Company Ltd., 2022.

2. S. L. Kakani, and S. Kakani, "Applied Physics-Theory & Practicals", Viva Books, 2014.

3. C. L. Arora, "Practical Physics", S. Chand & Company Ltd., 2010.

Reference Books

1. S. L. Gupta, and V. Kumar, "Practical Physics", 4th Edition, Vol. 1, Pragati Prakashan, 2017.

2. I. Prakash, R. Krishna, and A. K. Jha, "Practical Physics", Vol. 1, Kitab Mahal, 2011.

3. P. R. S. Kumar, "Practical Physics", Prentice Hall India Learning Pvt. Ltd., 2011.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50	50

SEMESTER II

Course Code	Course name		L	T	P	C
CSEG1032	Computer Organization and Architecture		3	0	0	3
Total Units to be Covered: 6		Total Contact Hours: 45				
Prerequisite(s):	Basic Knowledge of computer systems.		Syllabus version: 1.0			

Course Objectives

To equip students with the necessary knowledge and skills to comprehend, analyze, and design digital computer systems, ensuring they can effectively develop and optimize software applications and systems.

Course Outcomes

On completion of this course, the students will be able to

CO1: Analyze the components and organization of digital computers.

CO2: Apply knowledge of instruction codes, instruction formats, and addressing modes to analyze and design computer instructions in different CPU architectures.

CO3: Examine the design and organization of control units in digital computers to comprehend their role in executing instructions and managing system operations.

CO4: Analyze the organization and performance implications of memory units and input –output systems in digital computer systems.

CO5: Assess the benefits and challenges of pipelining in computer architecture on system performance and throughput.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3

CO 1	1		2	-	1	1	1	-	-	-	-	3	2	-	-
CO 2	1	1	2	-	1	1	1	-	-	-	-	3	2	-	-
CO 3	1	1	2	-	1	1	1	-	-	-	-	3	2	-	-
CO 4	1		2	-	1	1	1	-	-	-	-	3	2	-	-
CO 5	1	1	2	-	1	1	1	-	-	-	-	3	2	-	-
Average	1	0.6	2	-	1	1	1	-	-	-	-	3	2	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Digital Computers **8 Lecture Hours**

Introduction; Block Diagram for Digital Computers: CPU (Registers, ALU, Clock, Control unit), Memory, I/O subsystems, Common Bus System (External and Internal Bus: Address Bus, Data Bus and Control Bus); Computer Organization; Computer Architecture; Introduction to Vonn Neumann and Harvard Architecture, Data representation: Number System, r complement and r-1 complement arithmetic, Unsigned and Signed number representation, Big Endian and Little Endian, Signed Arithmetic- Addition, Subtraction, Multiplication (Booth Algorithm), Division, Fixed and Floating point representation. Register Transfer Language (RTL) and Micro operations (Arithmetic, Logical and Shift micro operations), Arithmetic Logic and Shift unit (ALU).

Unit II: Basic Computer Organization and Design **7 Lecture Hours**

Instruction Codes; Instruction Format (Three-Address Instructions, Two-Address Instructions, One-Address Instructions, Zero-Address Instruction); Computer Instructions, Registers (General Purpose and Special Purpose Registers); General Register Organization, Stack organization, Types of Instructions (Memory Reference, Register Reference and Input Output Instructions); Addressing Modes and its types; Instruction cycle, Interrupt cycle;

Case Study: Some common CPU architectures (Intel IA-32 Architecture, ARM),

Unit III: Control Unit Organization **6 Lecture Hours**

Hardwired Control Unit and Timing Signals, Microprogrammed control unit: control memory, Address sequencing, Microprogram Example, Designing of microprogrammed control unit. Comparison of hardwired and microprogrammed control units, RISC and CISC Processors;

Case Study: Designing a hypothetical processor with minimum number of instructions so that it can perform basic arithmetical and logical operations.

Unit IV: Memory Organization **8 Lecture Hours**

Memory hierarchy; Different types of memory: Primary (RAM-Static and Dynamic, ROM-EPROM,EEPROM, Cache-Level 1, Level 2 and Level 3) and Secondary/Auxiliary Memory (Magnetic Disk), Introduction to emerging in-situ memory technologies- ReRAM, PCM, STTRAM; Main Memory: RAM and ROM Chips, Memory Address Map, Memory Connection to CPU; Associative Memory. Cache Memory: Principle of Locality, Cache mapping techniques; Performance considerations: Hit Rate and Miss Penalty, cache coherence, cache read and write policy, caches on the Processor Chip.

Unit V: Input Output Organization **8 Lecture Hours**

Peripheral Devices; I/O interface; I/O bus and interface modules; I/O Bus vs Memory Bus, Interrupts and Types of Interrupts. Modes of data transfer: Programmed, Interrupt-initiated, Direct Memory Access (DMA), Priority Interrupt, Input Output processor.

Unit VI: Pipelining **8 Lecture Hours**

Multiprogramming, Multiprocessing, Single instruction single data stream (SISD); Single instruction multiple data stream (SIMD); Multiple instruction single data stream (MISD); Multiple instruction multiple data stream (MIMD), Multiprocessors: Shared memory and distributed memory, Parallel processing: Pipeline processing, Vector processing, Array processors. Pipelining: Arithmetic Pipeline, Instruction Pipeline: Example: Four-Segment Instruction Pipeline. Pipelining Conflicts: Resource conflicts, Data dependency and Branch difficulties, Pipeline Conflicts Handling techniques: Throughput and Speed; RISC pipeline;

Case study: Pipelining in CISC Processors, Pipelining in ColdFire Processors, and Pipelining in Intel Processors, Designing pipeline architecture for 2, 3, 4 stage pipeline.

Total lecture Hours 45

Textbooks

1. M. M. Mano, "Computer System Architecture", Revised 3rd Edition, Pearson Education, 2017.

2. Carl Hamacher, Zvonko Vranesic , Safwat Zaky, and Naraig Manjikian, "Computer Organization and Embedded Systems", 6th Edition, McGraw Hill, Standard Edition, 2023.

3. David A. Patterson, and John L. Hennessy, "Computer Organization and Design MIPS Edition: The Hardware/Software Interface", 5th Edition, The Morgan Kaufmann Series in Computer Architecture and Design, Morgan Kaufmann, 2020.

Reference Books

1. John P. Hayes, "Computer Architecture and Organization", 3rd Edition, McGraw-Hill Education, 2017.

2. William Stallings, "Computer Organization and Architecture: Designing for Performance", 11th Edition, Pearson, 2022.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSEG1033	Data Structures and Algorithms	4	0	0	4
Total Units to be Covered: 6	Total Contact Hours: 60				
Prerequisite(s):	Programming in C	Syllabus version: 1.0			

Course Objectives

1. Provide a clear understanding of the importance of data structures in organizing and manipulating data efficiently.
2. To introduce students to fundamental data structures and the properties, memory management, and basic operations of each data structure.
3. To offer practical experience in implementing data structures, common sorting, and searching algorithms.
4. Emphasize the use of data structures as tools for algorithmic problem-solving and apply their knowledge of data structures to solve real-world problems.

Course Outcomes

Upon completion of the course, the students will be able to

CO 1. State the significance and properties of the fundamental data structures.

CO 2. Implement common data structures while ensuring proper memory management and error handling.

CO 3. Illustrate expertise in understanding the common sorting and searching techniques with their complexities and implement them.

CO 4. Analyse real-world problems by understanding the trade-offs involved in identifying the appropriate data structure(s) based on problem requirements and using them to solve the problems efficiently.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	2	1	-	-	-	-	-	-	-	1	-	-	-

CO 2	1	1	2	1	-	-	-	-	-	-	-	-	-	-	-
CO 3	1	1	2	1	-	-	-	-	-	-	-	2	-	-	-
CO 4	1	2	2	2	-	-	-	-	-	-	-	2	-	-	-
Average	1	1	2	1.25	-	-	-	-	-	-	-	1.25	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: INTRODUCTION **12 Lecture Hours**

Overview, classification, and Importance of data structures in Programming/problem-solving, Basic terminology, and concepts: elements, operations, storage, memory allocation, garbage collection, and compaction. Iterative & Recursive approaches. Basic analysis of algorithms: Amortized analysis and Asymptotic Analysis. Array: Memory representation (1D and 2D), Array operations: insertion, deletion, searching. Applications of Array. Structure: Nested Structure, Function pointer as member of a structure, Self-referential structure. Anonymous Unions, ADTs.

Unit II: LINKED LIST **10 Lecture Hours**

Singly-Linked List, Doubly-Linked List, Circular Linked List, Header List, and its operations. Sentinel node. Generalized Linked List. Skip List. Applications of Linked Lists: polynomial manipulation, implementation of other data structures.

Unit III: STACK & QUEUE **10 Lecture Hours**

Stack data structure and operations. Queue data structure and operations. Implementation of Stack and Queue using Array and Linked List. Circular Queue. Deque and its types. Priority Queue. Applications: Stacks (Conversion of Infix to

Prefix/Postfix, Expression evaluation, a note on DFS in graph), Queues (Job scheduling, a note on BFS in graph).

Unit IV: TREE **10 Lecture Hours**

Introduction to Tree data structure and its terminologies, Binary Tree: properties, traversal algorithms (level-order, in-order, pre-order, post-order). Threaded Binary Tree. Binary Search Trees (BST): properties, operations (insertion, deletion, searching). Balanced BSTs. AVL Tree: properties, rotations, operations (insertion, deletion). Red-Black Tree. Multi-way search Tree: properties. B-Tree: properties, operations (search, insertion, and deletion). Applications of AVL Tree and B-Tree. Binary Heaps: properties, heapify operations, Heap sorting.

Unit V: HASH TABLE & GRAPH **10 Lecture Hours**

Hashing and hash functions. Hash table data structure: structure, collisions, collision resolution techniques, maintaining load factor. Applications of hash tables: dictionaries, symbol tables.

Introduction to graph data structure and its terminologies. Graph representations: adjacency matrix, adjacency list. Graph traversal algorithms: depth-first search, breadth-first search (BFS). Connected Components. Minimum spanning tree. Shortest path.

Unit VI: SORTING & SEARCHING **8 Lecture Hours**

Stability and In-place properties, Internal and external sorting. Simple comparison-based sorting algorithms: bubble sort, selection sort, insertion sort. Lower bound for comparison-based sorting algorithms. Recursive implementation of merge sort, quicksort, and binary search. Complexities of common sorting and searching algorithms.

Total lecture Hours 60

Textbooks

1. S. Lipschutz, "Data Structures with C", Schaum's Outline Series, McGraw-Hill Education (India) Pvt. Limited, 2017.

2. Y. P. Kanetkar, "Data structures through C", 4rd Edition, New Delhi: BPB, 2022.

Reference Books

1. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, "Data Structures and Algorithms", New Delhi: Pearson Education, 2003.

2. E. Horowitz, and S. Sahni, "Fundamentals of Data Structures in C", 2nd Edition, Hyderabad: University Press, 2008.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSEG1034	**Data Structures and Algorithms Lab**	0	0	2	1
Total Units to be Covered: 10	**Total Contact Hours: 30**				
Prerequisite(s):	Programming in C Lab	**Syllabus version: 1.0**			

Course Objectives

1. Students learn to analyze problems and design algorithms to solve them.
2. Develop students' ability to implement different data structures, common sorting, and searching algorithms using C programming language.
3. Provide students with practical exposure to the applications of data structures.

Course Outcomes

On completion of this course, the students will be able to

CO1. Implement linear data structures while ensuring proper memory management and error handling.

CO2. Design and develop code to demonstrate the use of non-linear data structures.

CO3. Illustrate expertise in understanding and implementing sorting and searching techniques.

CO4. Applying the appropriate data structure to solve real-world problems efficiently.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	2	2	1	-	-	-	-	-	-	-	1	2	1	-
CO 2	1	2	2	1	-	-	-	-	-	-	-	1	2	1	-
CO 3	1	2	2	1	-	-	-	-	-	-	-	1	2	1	-
CO 4	1	2	2	1	-	-	-	-	-	-	-	1	2	1	-
Average	1	2	2	1	-	-	-	-	-	-	-	1	2	1	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Basic Data Structure

To demonstrate the use of array, structure, and union along with dynamic memory allocation.

Experiment 2: Link List Data Structure and its Applications

To experiment with the concept of pointers, structure, and dynamic memory allocation to realize linked lists, their types, and application.

Experiment 3: Stack Data Structure

To use arrays and linked lists to implement Stack and its applications.

Experiment 4: Queue Data Structure

To demonstrate the use of arrays and linked lists to implement different variants of Queue and its applications.

Experiment 5: Trees

To demonstrate the creation of a binary tree and working with tree traversal.

Experiment 6: Heaps

To create a heap data structure and implement its operations, and its applications.

Experiment 7: Hash Tables

To implement a hash table using various collision resolution techniques, and its applications.

Experiment 8: Graphs

To demonstrate the creation of graphs and working with graph traversal algorithms.

Experiment 9: Sorting algorithms

To implement common sorting algorithms.

Experiment 10: Searching algorithms

To implement various search algorithms on data structures.

Total Lab hours 15

Textbooks

1. S. Lipschutz, "Data Structures with C", Schaum's Outline Series, McGraw-Hill Education (India) Pvt. Limited, 2017.

2. Y. P. Kanetkar, "Data structures through C", 4rd Edition, New Delhi: BPB, 2022.

Reference Books

1. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, "Data Structures and Algorithms", New Delhi: Pearson Education, 2003.

2. E. Horowitz, and S. Sahni, "Fundamentals of Data Structures in C", 2nd Edition, Hyderabad: University Press, 2008.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50	50

Course Code	Course name		L	T	P	C
CSEG1035	Python Programming		2	0	0	2
Total Units to be Covered: 06		Total Contact Hours: 30				
Prerequisite(s):	Problem Solving		Syllabus version: 1.0			

Course Objectives

1. Develop a strong foundation in Python programming language, including syntax, data types, control structures, and functions, enabling students to write efficient and reliable code.
2. Understand and apply object-oriented programming (OOP) principles in Python to design and build modular, reusable, and maintainable software solutions.
3. Gain proficiency in utilizing Python libraries and modules for tasks such as data manipulation, web scraping, data analysis, and visualization, empowering students to work with real-world data effectively.
4. Explore advanced topics in Python, including concurrency, file I/O, exception handling and equipping students with the skills to build robust and scalable applications.

Course Outcomes

CO1. Demonstrate proficiency in Python programming by writing code that adheres to Python syntax, utilizes appropriate data types, and implements control structures effectively.

CO2. Apply Python collections, such as lists, tuples, dictionaries, and sets, along with the design and implementation of reusable functions, to solve complex programming problems, demonstrating proficiency in data organization, manipulation, and modular code design.

CO3. Implement advanced Python features and techniques, such as modules and packages, file handling, exception handling and regular expression to create robust and reliable applications.

CO4. Apply object-oriented programming (OOP) concepts in Python to design and develop modular software solutions that promote code reusability and maintainability.

CO5. Utilize Python libraries and modules for data manipulation, analysis, and visualization, demonstrating the ability to work with real-world data sets and extract meaningful insights.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	2	-	-	-	-	-	-	-	2	1	-
CO 2	-	-	-	-	2	-	-	-	-	-	-	-	2	1	-
CO 3	-	-	-	-	2	-	-	-	-	-	-	-	2	1	-
CO4	-	-	-	-	2	-	-	-	-	-	-	-	2	1	-
CO5	-	-	-	-	2	-	-	-	-	-	-	-	2	2	-
Average	-	-	-	-	2	-	-	-	-	-	-	-	2	1.2	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to Python **6 Lecture Hours**

Introduction, Working with Python, Interactive mode and Scripting mode, Dynamic Types, Mutable and immutable data types, Basic Syntax, Comments, String Values, String Methods, The format Method, String Operators, Numeric Data Types, Input & Output functions, Escape sequence characters, Python Tokens (Keyword, identifier, special symbols, literals, constants, operators), Naming Conventions, Operators (Arithmetic, relational, logical, assignment, bitwise, membership, identity), Operators precedence and associativity, Type function and id function, Indentation, Decision Making Statements (if, if else, elif, nested if, match statement), range function, looping structures (while loop, for loop), break, continue, pass statement, else in loops, nested loops.

Unit II: Collections and Functions **5 Lecture Hours**

String initialization, String Operators, String functions, indexing, slicing, split() function, List initialization, List methods, List operations, indexing, slicing, list comprehension, Nesting in lists, tuple initialization, tuple methods, tuple operations, nesting in tuple, List vs Tuple,Set initialization, Set methods, Set operations, Dictionary initialization, Dictionary methods, nesting in Dictionary, Sorting data collections, typecasting collections, Applications of collections, Introduction, Defining user defined function, Parameters, Function Documentation, Keyword and Optional Parameters, default argument, Variable length Arguments, Scope, Passing Collections to a Function, Passing Functions to a Function, Recursion, map, filter, Lambda function, Inner Functions, Passing mutable and immutable datatypes in functions

Unit III: Module, Packages and Regular Expressions **4 Lecture Hours**

Introduction to modules and packages, creating modules and packages, Standard Modules – sys, math, time, os, Need for Regular Expressions, Regular Expression Functions (match, search, sub, findall, finditer), Meta characters, Character Class, Groups

Unit IV: File and Exception Handling **4 Lecture Hours**

File Access Modes, File handling Functions, Writing Data to a File, Reading Data from a File, Additional File Methods, With Statement, Working with Directories, Applications of File Handling, Errors vs Exceptions, The Exception Model, Exception Hierarchy, Exception Handling (try, except, else, finally), Handling Multiple Exceptions, raise, assert.

Unit V: Class and Objects in Python **5 Lecture Hours**

OOP Concepts, Classes in Python, Creating Classes and Objects, methods in classes, Constructor, Special Methods in classes, Class Variables and Object Variables, Public and Private data members, Built-in Class Attributes, Garbage Collection, Abstract class, Inheritance, types of inheritance, Polymorphism (Function overriding, operator overloading)

Unit VI: Data Analysis and Visualization **6 Lecture Hours**

Numpy – Overview, numpy Ndarray, Datatypes, Array creation, List vs Array, numpy attributes, numpy operations, Numpy Broadcasting, Numpy Functions (String, mathematical, statistical, sorting and searching), Numpy Special functions (reshape(), sum(), random(), zeros(), ones(), mean(), dot(), std(), empty(), arange(), numpy.linspace())

Pandas – Overview, Pandas Data Structures: Series and Data Frame, Operations on a Series (head, tail, vector operations), Data Frame operations(create, display, iteration, select column, add column, delete column), Binary operations in a Data Frame (add, sub, mul, div), Matching and broadcasting operations, Handling Missing data and filling values, Data Aggregation, Comparisons, Boolean reductions, comparing Series, Combining Data Frames, Importing/Exporting Data between CSV files and Data Frames.

Matplotlib- Introduction, Matplotlib Pyplot, Plotting, markers, Line, Labels, Grid, Customizing plots, Creating Different Types of Plots (Line Graph, Bar chart, Histograms, Scatter Plot, Pie Chart), Creating and working with Subplots

Total lecture Hours 30

Textbooks

1. Martin C. Brown, "Python: The Complete Reference", 4th Edition, McGraw Hill, 2018.

2. Paul Barry, "Head First Python", 2nd Edition, O′Reilly, 2016.

Reference Books

1. Luciano Ramalho, "Fluent Python", 2nd Edition, Learning Python Series, O′Reilly, 2015.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)

Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSEG1135	Python Programming Lab	0	0	4	2
Total Units to be Covered: 10	Total Contact Hours: 60				
Prerequisite(s):	Python Programming	Syllabus version: 1.0			

Course Objectives

1. Develop a strong foundation in Python programming language, including syntax, data types, control structures, and functions, enabling students to write efficient and reliable code.
2. Understand and apply object-oriented programming (OOP) principles in Python to design and build modular, reusable, and maintainable software solutions.
3. Gain proficiency in utilizing Python libraries and modules for tasks such as data manipulation, web scraping, data analysis, and visualization, empowering students to work with real-world data effectively.
4. Explore advanced topics in Python, including concurrency, file I/O, exception handling and equipping students with the skills to build robust and scalable applications.

Course Outcomes

On completion of this course, the students will be able to:

CO1. Demonstrate proficiency in Python programming by writing code that adheres to Python syntax, utilizes appropriate data types, and implements control structures effectively.

CO2. Apply Python collections, such as lists, tuples, dictionaries, and sets, along with the design and implementation of reusable functions, to solve complex programming problems, demonstrating proficiency in data organization, manipulation, and modular code design.

CO3. Implement advanced Python features and techniques, such as modules and packages, file handling, exception handling and regular expression to create robust and reliable applications.

CO4. Apply object-oriented programming (OOP) concepts in Python to design and develop modular software solutions that promote code reusability and maintainability.

CO5. Utilize Python libraries and modules for data manipulation, analysis, and visualization, demonstrating the ability to work with real-world data sets and extract meaningful insights.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	2	-	-	-	-	-	-	-	2	1	-
CO 2	-	-	-	-	2	-	-	-	-	-	-	-	2	1	-
CO 3	-	-	-	-	2	-	-	-	-	-	-	-	2	1	-
CO 4	-	-	-	-	2	-	-	-	-	-	-	-	2	1	-
CO 5	-	-	-	-	2	-	-	-	-	-	-	-	2	2	-
Average	-	-	-	-	2	-	-	-	-	-	-	-	2	1.2	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Python Installation and starting with python

1. Install Python and understand difference between scripting and interactive modes in IDLE.
2. Write Python programs to print strings in the given manner:
   a) Hello Everyone !!!
   b) Hello
      World
   c) Hello
            World
   d) ' Rohit' s date of birth is 12\05\1999'
3 Declare a string variable called x and assign it the value "Hello".

Print out the value of x

4 Take different data types and print values using print function.

5 Take two variable a and b. Assign your first name and last name. Print your Name after adding your First name and Last name together.

6 Declare three variables, consisting of your first name, your last name and Nickname. Write a program that prints out your first name, then your nickname in parenthesis and then your last name.

Example output : George (woody) Washington.

7 Declare and assign values to suitable variables and print in the following way :

NAME : NIKUNJ BANSAL

SAP ID : 500069944

DATE OF BIRTH : 13 Oct 1999

ADDRESS : UPES

Bidholi Campus

Pincode : 248007

Programme : AI & ML

Semester : 2

Experiment 2: Use of input statements, operators

1. Declare these variables (x, y and z) as integers. Assign a value of 9 to x, Assign a value of 7 to y, perform addition, multiplication, division and subtraction on these two variables and Print out the result.
2. Write a Program where the radius is taken as input to compute the area of a circle.
3. Write a Python program to solve (x+y)*(x+y)
   Test data : x = 4 , y = 3
   Expected output: 49
4. Write a program to compute the length of the hypotenuse (c) of a right triangle using Pythagoras theorem.
5. Write a program to find simple interest.
6. Write a program to find area of triangle when length of sides are given.
7. Write a program to convert given seconds into hours, minutes and remaining seconds.

8. Write a program to swap two numbers without taking additional variable.
9. Write a program to find sum of first n natural numbers.
10. Write a program to print truth table for bitwise operators(& , | and ^ operators)
11. Write a program to find left shift and right shift values of a given number.
12. Using membership operator find whether a given number is in sequence (10,20,56,78,89)
13. Using membership operator find whether a given character is in a string.

Experiment 3 : Conditional Statements

1. Check whether given number is divisible by 3 and 5 both.
2. Check whether a given number is multiple of five or not.
3. Find the greatest among two numbers. If numbers are equal than print "numbers are equal".
4. Find the greatest among three numbers assuming no two values are same.
5. Check whether the quadratic equation has real roots or imaginary roots. Display the roots.
6. Find whether a given year is a leap year or not.
7. Write a program which takes any date as input and display next date of the calendar
   e.g.
   I/P: day=20 month=9 year=2005
   O/P: day=21 month=9 year 2005
8. Print the grade sheet of a student for the given range of cgpa. Scan marks of five subjects and calculate the percentage.

CGPA=percentage/10

CGPA range:

0 to 3.4 -> F
3.5 to 5.0->C+
5.1 to 6->B
6.1 to 7-> B+

7.1 to 8-> A

8.1 to 9->A+

9.1 to 10-> O (Outstanding)

Sample Gradesheet

Name: Rohit Sharma

Roll Number: R17234512 SAPID: 50005673

Sem: 1 Course: B.Tech. CSE AI&ML

Subject name: Marks

PDS:	70
Python:	80
Chemistry:	90
English:	60
Physics:	50

Percentage: 70%

CGPA:7.0

Grade:

Experiment 4: Loops

1. Find a factorial of given number.
2. Find whether the given number is Armstrong number.
3. Print Fibonacci series up to given term.
4. Write a program to find if given number is prime number or not.
5. Check whether given number is palindrome or not.
6. Write a program to print sum of digits.
7. Count and print all numbers divisible by 5 or 7 between 1 to 100.
8. Convert all lower cases to upper case in a string.
9. Print all prime numbers between 1 and 100.
10. Print the table for a given number:

    5 * 1 = 5

    5 * 2 = 10………..

Experiment 5: String and Sets

1. Write a program to count and display the number of capital letters in a given string.
2. Count total number of vowels in a given string.
3. Input a sentence and print words in separate lines.
4. WAP to enter a string and a substring. You have to print the number of times that the substring occurs in the given string. String traversal will take place from left to right, not from right to left.
   Sample Input
   ABCDCDC
   CDC
   Sample Output
   2
5. Given a string containing both upper and lower case alphabets. Write a Python program to count the number of occurrences of each alphabet (case insensitive) and display the same.
   Sample Input
   ABaBCbGc
   Sample Output
   2A
   3B
   2C
   1G
6. Program to count number of unique words in a given sentence using sets.
7. Create 2 sets s1 and s2 of n fruits each by taking input from user and find:
   a) Fruits which are in both sets s1 and s2
   b) Fruits only in s1 but not in s2
   c) Count of all fruits from s1 and s2
8. Take two sets and apply various set operations on them :
   S1 = {Red ,yellow, orange , blue }
   S2 = {violet, blue , purple}

Experiment 6: Lists, tuples, dictionary

1. Scan n values in range 0-3 and print the number of times each value has occurred.
2. Create a tuple to store n numeric values and find average of all values.
3. WAP to input a list of scores for N students in a list data type. Find the score of the runner-up and print the output.
   Sample Input
   N = 5
   Scores= 2 3 6 6 5
   Sample output
   5
   Note: Given list is [2, 3, 6, 6, 5]. The maximum score is 6, second maximum is 5. Hence, we print 5 as the runner-up score.
4. Create a dictionary of n persons where key is name and value is city.
   a) Display all names
   b) Display all city names
   c) Display student name and city of all students.
   d) Count number of students in each city.
5. Store details of n movies in a dictionary by taking input from the user. Each movie must store details like name, year, director name, production cost, collection made (earning) & perform the following :-
   a) print all movie details
   b) display name of movies released before 2015
   c) print movies that made a profit.
   d) print movies directed by a particular director.

Experiment 7: Functions

1. Write a Python function to find the maximum and minimum numbers from a sequence of numbers. (Note: Do not use built-in functions.)
2. Write a Python function that takes a positive integer and returns the sum of the cube of all the positive integers smaller than the specified number.
3. Write a Python function to print 1 to n using recursion. (Note: Do not use loop)

4. Write a recursive function to print Fibonacci series upto n terms.
5. Write a lambda function to find volume of cone.
6. Write a lambda function which gives tuple of max and min from a list.
   Sample input: [10, 6, 8, 90, 12, 56]
   Sample output: (90,6)
7. Write functions to explain mentioned concepts:
   a. Keyword argument
   b. Default argument
   c. Variable length argument

Experiment 8: File Handling and Exception Handling

1. Add few names, one name in each row, in "name.txt file".
   a. Count no of names
   b. Count all names starting with vowel
   c. Find longest name
2. Store integers in a file.
   a. Find the max number
   b. Find average of all numbers
   c. Count number of numbers greater than 100
3. Assume a file city.txt with details of 5 cities in given format (cityname population(in lakhs) area(in sq KM)):
   Example:
   Dehradun 5.78 308.20
   Delhi 190 1484
   ……………
   Open file city.txt and read to:
   a. Display details of all cities
   b. Display city names with population more than 10Lakhs
   c. Display sum of areas of all cities
4. Input two values from user where the first line contains N, the number of test cases. The next N lines contain the space separated values of a and b. Perform integer division and print a/b. Handle exception in case of ZeroDivisionError or ValueError.
   Sample input

1 0

2 $

3 1

Sample Output :

Error Code: integer division or modulo by zero

Error Code: invalid literal for int() with base 10: '$' 3

5. Create multiple suitable exceptions for a file handling program.

Experiment 9: Classes and objects

1.Create a class of student (name, sap id, marks[phy,chem,maths]). Create 3 objects by taking inputs from the user and display details of all students.

2. Add constructor in the above class to initialize student details of n students and implement following methods:

a) Display() student details
b) Find Marks_percentage() of each student
c) Display result() [Note: if marks in each subject >40% than Pass else Fail]

Write a Function to find average of the class.

3. Create programs to implement different types of inheritances.

4. Create a class to implement method Overriding.

5. Create a class for operator overloading which adds two Point Objects where Point has x & y values

e.g. if

P1(x=10,y=20)

P2(x=12,y=15)

P3=P1+P2 => P3(x=22,y=35)

Experiment 10: Data Analysis and Visualization

1. Create numpy array to find sum of all elements in an array.
2. Create numpy array of (3,3) dimension. Now find sum of all rows & columns individually. Also find 2^{nd} maximum element in the array.
3. Perform Matrix multiplication of any 2 n*n matrices.
4. Write a Pandas program to get the powers of an array values element-wise.

Note: First array elements raised to powers from second array

Sample data: {'X':[78,85,96,80,86], 'Y':[84,94,89,83,86],'Z':[86,97,96,72,83]}

Expected Output:

X Y Z

0 78 84 86

1 85 94 97

2 96 89 96

3 80 83 72

4 86 86 83

5. Write a Pandas program to get the first 3 rows of a given DataFrame.

Sample Python dictionary data and list labels:

exam_data = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],

'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],

'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],

'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

Expected Output:

First three rows of the data frame:

attempts name qualify score

a 1 Anastasia yes 12.5

b 3 Dima no 9.0

c 2 Katherine yes 16.5

6. Write a Pandas program to find and replace the missing values in a given DataFrame which do not have any valuable information.

7. Create a program to demonstrate different visual forms using Matplotlib.

Total Lab hours 60

Textbooks

1. Reema Thareja, "Python Programming", Oxford University Press, 2017.

2. Mark Lutz, "Learning Python", 5th ed., O’reilly publication, 2013.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50	50

Course Code	Course name		L	T	P	C
ECEG1012	Digital Electronics		3	0	0	3
Total Units to be Covered: 5		Total Contact Hours: 45				
Prerequisite(s):	Computer System Architecture		Syllabus version: 1.0			

Course Objectives

1. Demonstrate a basic understanding of digital terminology, digital components, and systems.
2. To prepare students to perform the analysis and design of various digital electronic circuits.
3. To give the students a perspective to design combinational and sequential circuits.

Course Outcomes

On completion of this course, the students will be able to

CO1: Understand working of logic families and logic gates.

CO2: Design and implement Combinational and Sequential logic circuits.

CO3: Understand the process of Analog to Digital conversion and Digital to Analog conversion.

CO4: Understand the configuration and working of Microprocessor.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	2	-	1	-	-	-	-	-	-	-	-	-	-	-
CO 2	2	2	-	1	-	-	-	-	-	-	-	-	-	-	-
CO 3	2	2	-	1	-	-	-	-	-	-	-	-	-	-	-
CO 4	2	2	-	1	-	-	-	-	-	-	-	-	-	-	-
Average	2	2	-	1	-	-	-	-	-	-	-	-	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Fundamentals of Digital Systems and logic families 8 Lecture Hours

Digital signals, digital circuits, AND, OR, NOT, NAND, NOR and Exclusive-OR operations, Boolean algebra, examples of IC gates, number systems-binary, signed binary, octal hexadecimal number, binary arithmetic, one's and two's complements arithmetic, codes, error detecting and correcting codes, characteristics of digital lCs, digital logic families, TTL, Schottky TTL and CMOS logic, interfacing CMOS and TTL, Tri-state logic.

Unit II: Combinational Digital Circuits 10 Lecture Hours

Standard representation for logic functions, K-map representation, and simplification of logic functions using K-map, minimization of logical functions. Don't care conditions, Multiplexer, De-Multiplexer/Decoders, Adders, Subtractors, BCD arithmetic, carry look ahead adder, serial adder, ALU, elementary ALU design, popular MSI chips, digital comparator, parity checker/generator, code converters, priority encoders, decoders/drivers for display devices, Q-M method of function realization.

Unit III: Sequential circuits and systems 10 Lecture Hours

A 1-bit memory, the circuit properties of Bistable latch, the clocked SR flip flop, J-K-T And D-Types flip flops, applications of flip flops, shift registers, applications of shift registers, serial to parallel converter, parallel to serial converter, ring counter, sequence generator, ripple (Asynchronous) counters, synchronous counters, counters design using flip flops, special counter IC's, asynchronous sequential counters, applications of counters.

Unit IV: A/D and D/A Converters 10 Lecture Hours

Digital to analog converters: weighted resistor/converter, R-2R Ladder D/A converter, specifications for D/A converters, examples of D/A converter lCs, sample and hold circuit, analog to digital converters: quantization and encoding, parallel comparator A/D converter, successive approximation A/D converter, counting A/D converter, dual slope A/D converter, A/D converter using voltage to frequency and voltage to time conversion, specifications of A/D converters, example of A/D converter ICs.

Unit V: Introduction to Microprocessor **7 Lecture Hours**

Introduction to Microprocessors, Architecture of 8085 and 8086, Pin Configuration and Function; Internal Register & Flag Register, Generation of Control Signals Bus Timings: Demultiplexing of Address /Data Bus; Fetch Cycle, Execute Cycle, Instruction Cycle, Instruction Timings and Operation Status, Timing Diagram. Instruction for Data Transfer. Arithmetic and Logical Operations. Branching Operation: Machine Cycle Concept; Addressing Modes.

Total lecture Hours 45

Textbooks

1. R. P. Jain, "Modern Digital Electronics", 4th Edition, McGraw Hill Education, 2009.

2. M. M. Mano, "Digital logic and Computer design", Pearson Education India,

Reference Books

1. A. A. Kumar, "Fundamentals of Digital Circuits", 3rd Edition, Prentice Hall India, 2016.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
SLSG0102	Critical Thinking and Writing	2	0	0	2
Total Units to be Covered: 9	Total Contact Hours: 30				
Prerequisite(s):		Syllabus version: 1.0			

Course Objectives

The objectives of this course are:

1. To introduce the essential tools and approaches of critical thinking.
2. To realize how several factors hinders the process of critical thinking and how to overcome them.
3. To understand and the various components and conventions of critical writing and create appropriate documents.

Course Outcomes

CO 1 Understand the importance of Critical Thinking in the process of decision making.

CO 2 Differentiate amongst the various tools and approaches of critical thinking; look at the world around objectively and critically

CO 3 Critically analyze any text and communicate the inferences drawn after analysis; introspect and reflect on their thought processes; draw logical conclusions and identify the errors in reasoning

CO 4 Articulate written documents demonstrating critical approaches with clear, structured and quality writing.

CO 5 Apply critical thinking to any provided information. Identify, understand and define the various arguments in different contexts

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	1	-	1	1	1	1	-	3	3	1	3	-	-	-
CO 2	-	2	1	-	2	1	1	-	2	3	1	1	-	-	-
CO 3	-	1	1	-	2	-	1	1	1	3	1	1	-	-	-
CO 4	-	-	1	-	2	1	1	1	2	3	-	1	-	-	-
CO 5	-	2	-	2	3	-	-	1	3	1	-	3	-	-	-

Average	-	1.2	.6	.6	2	.6	.8	.6	2.2	2.6	.6	1.8	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Food for Thought: **2 Lecture Hours**

What is Critical Thinking?
Introduction to the course, its importance and its application in life. Focus is given on the Trolly problem and how it can never be solved.

Unit II: Learning how to learn: **2 Lecture Hours**

Cognition and Metacognition
The highlight of this unit would be learning strategies and the development of Bloom's Taxonomy. This leacture based class will focus on education and learning challenges faced by students across South Asia and how to over come them.

Unit III: How to not judge a book by its cover: **2 Lecture Hours**

Cognitive biases
Flagging the problems with assumptions in our everyday functioning, this class will highlight the various kinds of biases and how it effects our understanding of issues when it comes to problem solving.

Unit IV: Writing to read **8 Lecture Hours**

Introduction to various aspects of writing and highlighting how one is different from the other.
Understanding of capital letters and syntaxes will be another focus of these classes.

Unit V: "…but why?": **2 Lecture Hours**

The Social, The Historical, and the Political Aspects of Reasoning:
Introduction to inductive and deductive reasoning and its relevance when understanding how information is passed on to.

Unit VI: "Agree to disagree": **4 Lecture Hours**
Explanation, Justification, Persuasion
Explanation, Justification, Persuasion are three distinct critical thinking tool to convey information, support arguments, or influence others in various contexts.

Unit VII: Fact, Truth, and Misinformation **2 Lecture Hours**
Based in accuracy and reliability of information, this unite with focus on assertion of statement that can be objectively verified and proven to be true or false depending upon the situation. It is important to be critically evaluate information to differentiate one from the other.

Unit VIII: Critical Consumption **2 Lecture Hours**
Focusing on critical media consumption, this unit focuses on the contemporary forms of information consumption. It involves being actively aware if potential biases, misinformation, and the credibility of sources in the age of digital media.

Unit IX: Perspective Taking **4 Lecture Hours**
This unit focuses on the Cognitive and empathetic process in which an individual tries to understand and empathise with the thoughts, feelings, beliefs, and experiences of the other person or group from their point of view. Perspective taking is an important aspect of empathy and interpersonal communication aiding critical thinking

Unit IX: Ethical Dilemma **2 Lecture Hours**
The focus of this unit will be on the complex moral decision that involves conflicting values, principles, or interests. In such instances there is no clear or obvious choice to arrive at a conclusion.

Total lecture Hours 30

Textbooks

1. Lewis Vaughn, The Power of Critical Thinking: Effective Reasoning About Ordinary and Extraordinary Claims, 4th ed. Editorial: Oxford University Press, 2013.
2. W. Hughes and Jonathan Allen Lavery, Critical thinking : an introduction to the basic skills. Peterborough: Broadview Press, 2016.
3. A. West, "Ubuntu and Business Ethics: Problems, Perspectives and Prospects," Journal of Business Ethics, vol. 121, 2013.

Reference Books

1. S. Yu and F. Zenker, "Schemes, Critical Questions, and Complete Argument Evaluation," Argumentation, vol. 34, no. 4, pp. 469–498, Mar. 2020
2. Davies, Richard, "In Defence of a Fallacy", Studia Semiotyczne 34 (2):25-42, 2020.
3. L. Christoph, "Recognizing Argument Types and Adding Missing Reasons", Proceedings of the Ninth Conference of the International Society for the Study of Argumentation (ISSA), Amsterdam (Netherlands): pp. 769-777, 2019.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	Class writing assignment and participation	Quiz 1	Quiz 2	Writing assignment	Total
Weightage (%)	10	30	30	30	100

Course Code	Course name	L	T	P	C
SSEN0102	Environment Sustainability and Climate Change (Living Lab)	2	0	0	2
Total Units to be Covered: 07	Total Contact Hours: 30				
Prerequisite(s):	Fundamentals of basic Environment Sustainability and Climate Change	Syllabus version: 1.0			

Course Objectives

1. Understand the concept of Living Labs and their application in the environmental sustainability.
2. Develop a critical understanding of the nature, cause and impact of human activities on the environment.
3. Apply design thinking and innovative principles to develop sustainable solutions.
4. Evaluate and address legal, policy and ethical consideration in environmental research.

Course Outcomes

CO1. Gained practical skills in stakeholder engagement, environmental data collection and analysis.

CO2. Develop expertise in designing and managing Living Lab for environmental sustainability and climate action.

CO3. Acquired hands on experience with environmental monitoring tools and technologies.

CO4. Enhance the ability to think critically and creatively in developing sustainable solutions.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
CO 2	2	1	-	-	1	-	-	-	-	-	-	-	-	-	-
CO 3	3	1	-	1	2	-	-	1	-	-	2	-	-	-	-
CO 4	2	2	-	1	1	-		1	-	-	1	-	-	-	-

Average	2.25	1	-	.5	-	-	-	.5	-	-	.75	-	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

1. A two-credit course in practicum or lab work, community engagement and services, and field work in a semester means two-hour engagement per week. In a semester of 15 weeks duration, a one credit practicum in a course is equivalent to 30 hours of engagement.
2. **Case Studies and Field Work**

   The students are expected to be engaged in some of the following or similar identified activities:

   Discussion on one national and one international case study related to the environment and sustainable development.

   Examples: Bhopal Gas Tragedy, Chipko Movement, Narmada Valley Projects, National Park, Sanctuaries, Biosphere Reserve, London Smog 1952, Air Pollution in Delhi, Case studies on Current Environmental Issues, Oil Spills – Deep Water Horizon Oil Spill, BP Oil Spill etc.
3. **Field Visit**

   Field visits to identify local/regional environmental issues, make observations including data collection and prepare a brief report.
4. **Campus Environmental Management.**

   Campus environmental management activities such as solid waste disposal, water management, and sewage treatment.

   Group Project: Students are required to submit group projects on various topics related to environmental pollution, climate change, biodiversity, natural resource and sustainable development.

Broadly, Living Lab may falls in one of seven thrust areas:

1. **Indigenous technology and Traditional Ecological Knowledge (TEK)**

   The project aims to document, preserve and revitalize Traditional Ecological Knowledge (TEK) held by indigenous communities. TEK encompasses the deep understanding of local ecosystems, sustainable solution resource management practices and cultural connection to environment. This project emphasizes a collaborative approach involving indigenous elders, community members and researcher. The project not only respects and preserves the rich cultural heritage of indigenous communities but also harnesses their valuable ecological knowledge to address contemporary environmental challenges and promote sustainable practices. It emphasizes the importance of community-driven conservation efforts and the recognition of TEK as a valuable source of ecological wisdom.

2. **Climate change and its impact on Bird Migration**

   In recent years, climate change has been affecting the migration patterns of many bird species worldwide. This project aims to study and mitigate the impact of climate change on avian migration and contribute to conservation efforts. This project not only addresses the ecological impact of climate change but also contributes to the conservation of bird species that play vital roles in ecosystem health and biodiversity. It emphasizes the importance of interdisciplinary collaboration and community engagement in tackling climate-related ecological challenges. This project will focus on developing targeted conservation strategies to mitigate the impact of climate change on bird population and help in enhancing collaboration among scientists, conservationists, and local communities for bird conservation.

3. **Sustainable Communities**

   How can co-production and social learning with stakeholder communities help us understand how climate action can be implemented 'on the ground'?
   The 'living lab' offer for active engagement with a diverse student body and neighborhood groups. It reflects the wider academic recognition that universities are significant economic, social and environmental catalysts for cities and regions, offering the potential for change at a spatial scale that connects the local with the global. Project activities to empower the local community-based people to enhance their

lifestyle by doing activities. For example, we can do a few projects like utilization of Himalayan biomass for various uses. Our students can give this training and awareness program to localities.

We can work on the SMART village project by following the SDG goals given by United Nations.

Identification and selection of such communities who have some native or ancestral knowledge. For example, one farmer in Kerala has huge seed bank from the very old time (more than 200 Years).

Projects in this area will explore how meaningful policy change can be driven in expanding circles from the level of university communities to the cities, states and nations they are embedded in.

4. **Ecology, Conservation, and Climate Change**

Project within this domain will investigate the ecological characteristics of ecosystem undergoing degradation, examine the dynamics of shifts in parasite ecology and explore the enduring adaptation in hosts, parasites and explore the evolutionary adaptation in host, parasites and wildlife influenced by climate fluctuation and various environmental stressors.

UPES can work on the preservation of untouched Himalayan flora and fauna and propose one flora and fauna bank. This may be followed by the several awareness program for the locals by our staffs and students. The primary aim of projects in this area will be to establish and sustain long-term studies of how climate change impacts ecological aspects of our natural world.

5. **One Health**

This project is centered around the concept of "One Health," which is an approach that recognizes the interconnectedness of human health, animal health, and the health of the environment. It aims to address the broader context of ecological health, acknowledging that various factors such as climate change, habitat alterations, and biodiversity loss play a significant role in shaping the overall health of ecosystems and, consequently, human health. In essence, this project seeks to broaden our understanding of health by looking beyond the human dimension and recognizing the intricate web of relationships that connect human health to animal health and the environment.

One of the primary concerns is to examine the potential sources of new zoonotic diseases. Zoonotic diseases are those that can be transmitted from animals to humans, and understanding their origins is crucial for preventing future outbreaks.

6. **Climate and society**

Climate action will require disruptive transformations for society, but how can that 'transformational intent' be developed? The digital revolution and data science show how rapid transformation can happen, history offers perspectives on such transformations in the past while the business world illuminates how rapid change continues to occur in the commercial and financial sector.
This group of projects will study the interface between climate, sustainable living, and the forces that drive society and societal change.

7. **Communities Based Water Testing Kits and Soil Testing Kits and promotion of low cost water purification technologies**

Identify water borne disease in surrounding villages by conducting field surveys and sensitize local communities about water borne diseases and suggest low-cost water treatment methods.
Training local people for water and soil testing.
Sampling and analysis of drinking water

Total lecture Hours 30

Textbooks

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	Continuous Assessment
Weightage (%)	100%

Course Code	Course name		L	T	P	C
MATH1065	Advanced Engineering Mathematics – 2		3	1	0	4
Total Units to be Covered: 5		Total Contact Hours: 60				
Prerequisite(s):	Advanced Engineering Mathematics - I		Syllabus version: 1.0			

Course Objectives

The aim of the course is to prepare the students to understand and appreciate the power of mathematics as a unifying language transcending a variety of engineering and science disciplines. The focus for the designing of the syllabus has been to provide the students with insights into the mathematical concepts and their applications without much compromising mathematical rigor.

Course Outcomes

On completion of this course, the students will be able to

CO1. Formulate appropriate numerical and optimization schemes for the development of computational algorithms in science and engineering.

CO2. Understand the fundamental tools of complex analysis for addressing and solving a variety of problems viz. special functions, integral transforms, and PDEs.

CO3. Identify and illustrate the use of Bessel functions and Legendre polynomials in real world application.

CO4. Demonstrate computational implementation of integral transforms and their applications.

CO5. Model real-world phenomena evolving in space and time governed by linear PDEs.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	3	2	-	2	-	-	-	-	-	-	-	-	-	-
CO 2	3	3	2	-	1	-	-	-	-	-	-	-	-	-	-

CO 3	3	3	2	-	1	-	-	-	-	-	-	-	-	-	-
CO4	3	3	3	-	3	-	-	-	-	-	-	-	-	-	-
CO5	3	3	3	-	3	-	-	-	-	-	-	-	-	-	-
Average	3	3	2.4	-	2	-	-	-	-	-	-	-	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit 0: Motivation **01 Lecture Hours**

Why study this course- Relevance and Significance?

Unit I: Numerical Methods and Optimization **13 Lecture Hours**

Bisection and Newton-Raphson methods, Gauss Elimination and Gauss-Seidel methods, Finite difference operators, Interpolation with equal and unequal intervals, Numerical differentiation, and integration, Numerical solution of ODEs: Picard's method, Euler's method, Runge-Kutta fourth order method. Introduction to optimization, The Simplex method, Duality, Lagrange multipliers, Convex sets and functions, Elements of Gradient search algorithms: Steepest descent, Newton and Jacobi algorithms, Least Squares method, Application: The Markowitz Model, and Overview of constrained optimization, Hill climbing, Single variable search. Recap of Unit I.

Unit II: Infinite Series and Introduction to Complex Analysis **13 Lecture Hours**

Sequence and series, Convergence tests: p-series, Comparison, Ratio and root test, Alternating series. Complex number system, Euler's formula, Functions of a complex variable, Hyperbolic functions, Limit and Continuity, Derivative and Analytic functions, Holomorphic functions, Cauchy-Riemann equations, Harmonic functions, Line integral and independence of path, Cauchy's theorem, Cauchy's integral formula, Zeros and singularities of a function, Power series: Taylor's and Laurent's series. Some applications. Recap of Unit II.

Unit III: Introduction to Special Functions **07 Lecture Hours**

Introduction to Power series method, Legendre's equation and Legendre polynomials, Bessel's equation and Bessel functions. Application of Bessel functions: CV Raman's model of Indian drums. Recap of Unit III.

Unit IV: Integral Transforms **15 Lecture Hours**

Laplace Transform and its properties, Shifting Theorems, Laplace Transform of derivatives, integrals, and periodic functions, Heaviside and Dirac Delta Functions. Inverse Laplace transforms, Convolution, Solutions of differential equations using Laplace transforms. Fourier series and applications, Dirichlet's condition, Fourier Transforms, Fourier sine and cosine transforms, Properties of Fourier Transforms, Fast Fourier Transform, Inverse Fourier transforms. Recap of Unit IV.

Unit V: Introduction to PDEs and Applications **11 Lecture Hours**

Introduction to Partial differential equations (PDE) and real-world applications, Classification of PDEs: Elliptic, Hyperbolic, Parabolic, Solution of homogeneous and non-homogeneous linear PDEs, Method of separation of variables using Fourier series, Solution of Heat conduction or Diffusion equation, Connection between diffusion and randomness, Wave Equation, Laplace Equation, and Poisson Equation. Some applications: Air pollution, Traffic model. Recap of Unit V.

Total lecture Hours 60

Textbooks

1. E. Kreyszig, "Advanced Engineering Mathematics", 9th Edition, Wiley Publications, 2016.

2. G. James, and P. P. Dyke, "Advanced Modern Engineering Mathematics", 5th Edition, Pearson Education. 2018.

3. J. P. Corriou, "Numerical Methods and Optimization: Theory and Practice for Engineers", Springer, 2021.

Reference Books

1. D. G. Zill, and P. D. Shanahan, "Complex analysis", 3rd Edition, Jones & Bartlett Learning, 2015.

2. W. A. Strauss, "Partial Differential Equations - An Introduction", 2nd Edition, John Wiley & Sons Inc., 2008.

3. L. Burstein, "PDE Toolbox Primer for Engineering Applications with MATLAB® Basics", CRC Press, 2022.

4. I. Goodfellow,Y. Bengio, and A. Courville, "Deep learning", MIT Press., 2016. (Sections 4.3, 4.4, and 4.5)

5. S. Chandra, Jayadeva, and A. Mehra, "Numerical optimization with applications", Narosa, 2009.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

SEMESTER III

Course Code	Course name		L	T	P	C
CSAI2015	Elements of AIML		2	0	0	2
Total Units to be Covered: 5		Total Contact Hours: 30				
Prerequisite(s):	Problem Solving Techniques, Probability and Statistical Analysis	Syllabus version: 1.0				

Course Objectives

Students will learn the basic concepts and techniques of Artificial Intelligence and Machine Learning. They will also explore their applications.

Course Outcomes

On completion of this course, the students will be able to

CO1. Understand the basic concepts and techniques of Artificial Intelligence.

CO2. Understand the logic of AI algorithms for solving practical problems.

CO3. Understand the basics of Machine Learning and its types.

CO4. Assess and model real-world practical problems that can be handled by AI and ML.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	1	1	-	-	-	-	-	-	-	-	-	1	-	-
CO 2	1	1	1	-	-	-	-	-	-	-	-	-	1	-	-
CO 3	1	1	1	-	-	-	-	-	-	-	-	-	1	-	-
CO 4	1	1	1	-	-	-	-	-	-	-	-	-	2	-	-
Average	1	1	1	-	-	-	-	-	-	-	-	-	1.25	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction **6 Lecture Hours**

Artificial Intelligence and its applications, Artificial Intelligence Techniques, Level of models, criteria of success, Intelligent Agents, Nature of Agents, Learning Agents. AI Techniques, advantages, and limitations of AI, Impact and Examples of AI, Application domains of AI.

Unit II: Logic for AI **6 Lecture Hours**

Propositional logic, predicate logic, Resolution, Resolution in proportional logic and predicate logic, Clause form, unification algorithm

Unit III: Introduction to machine learning **6 Lecture Hours**

Introduction to Machine Learning, Usage of datasets and how to handle them for Machine Learning Feature sets, Dataset division: test, train and validation sets, cross validation, Dimensionality Reduction Techniques: PCA, LDA, ICA

Unit IV: Types of machine learning **6 Lecture Hours**

Introduction to Machine Learning Techniques: Supervised Learning: Regression and its types, Classification, Unsupervised Learning: Clustering, Reinforcement Learning, Semi-supervised Machine Learning

Unit V: Applications of AI and Machine learning **6 Lecture Hours**

AI for society, women and environment, Applications of Machine Learning in Banking, Security, Healthcare, Education, Insurance Industry, Retail and Supply Chain, Transportation and Logistics, Energy and Utilities

Total lecture Hours 30

Textbooks

1. Artificial Intelligence by Rich and Knight, The McGraw Hill, 2017.
2. Machine Learning for Dummies, By John Paul Mueller and Luca Massaron, For Dummies, 2016.

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSAI2115	Elements of AIML Lab		0	0	2	1
Total Units to be Covered: 10		Total Contact Hours: 30				
Prerequisite(s):	Python Programming	Syllabus version: 1.0				

Course Objectives

Students will learn the basic concepts and techniques of Artificial Intelligence and Machine Learning using tolls and python programming language. They will also explore their applications.

Course Outcomes

On completion of this course, the students will be able to

CO1 Understand the basic concepts and techniques of Artificial Intelligence and Machine Learning Algorithms on tools.
CO2 Explore the data analysis and preparation techniques, the logic of AI algorithms for solving practical problems.
CO3 Understand the basics of Machine Learning and its types.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	2	-	-	-	-	-	-	-	2	1	-
CO 2	-	-	-	-	2	-	-	-	-	-	-	-	2	1	-
CO 3	-	-	-	-	2	-	-	-	-	-	-	-	2	1	-
Average	-	-	-	-	2	-	-	-	-	-	-	-	2	1	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

List of Experiments

Experiment 1	Exploration of WEKA tool for Regression task.
Experiment 2	Exploration of WEKA tool for Classification task.
Experiment 3	Exploration of WEKA tool for Clustering task.
Experiment 4	Write a python program to import and export data using Pandas and show the details of the dataset like number of rows, columns, first five rows, size, number of missing values, sum, average, min and max values from the numerical columns.
Experiment 5	Using Python language do the exploratory data analysis of dataset imported in the lab 4.
Experiment 6	Implement the missing value, and outlier handling data preprocessing techniques on the dataset imported in lab 4 or any other dataset.
Experiment 7	Implement feature scaling and one hot encoding data preprocessing techniques on the dataset imported in lab 4 or any other dataset.
Experiment 8	Implement Dimensionality reduction using Principal Component Analysis (PCA) method.
Experiment 9	Implement different techniques of handling imbalanced data.
Experiment 10	Write a Python program to demonstrate various Data Visualization Techniques using Matplotlib and Seaborn libraries.

Total Lab hours 15

Textbooks

1. “Data Science for Business" by Foster Provost and Tom Fawcett.
2. "Python for Data Analysis" by Wes McKinney.

Reference Books

1. "Data Wrangling with Python" by Kevin Markham.
2. "Storytelling with Data" by Cole Nussbaumer Knaflic.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50	50

Course Code	Course name	L	T	P	C
CSEG2046	Database Management Systems	3	0	0	3
Total Units to be Covered: 7	Total Contact Hours: 45				
Prerequisite(s):	Exposure on Data structures.	Syllabus version: 1.0			

Course Objectives

1. To understand the concept of DBMS and ER Modelling.
2. To explain normalization, Query optimization and relational algebra.
3. To apply concurrency control, recovery, security and indexing for real time data.

Course Outcomes

CO 1 Understand the foundational concepts of data models, schema design, and relational databases to effectively manage and query data.

CO 2 Learn to design efficient and normalized databases, apply entity-relationship modeling, and optimize schema structures.

CO 3 Acquire skills in database security, user access control, backup and recovery, and performance tuning to ensure robust database management.

CO 4 Develop database application design and its implementation including integrity constraints, transaction management and concurrent control algorithms.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	3	2	2	-	-	-	-	-	-	-	-	2	3	-
CO 2	-	3	2	2	-	-	-	-	-	-	-	-	2	3	-
CO 3	-	3	2	3	-	-	-	-	-	-	-	-	2	3	-
CO 4	-	3	2	3	-	-	-	-	-	-	-	-	2	3	-
Average	-	3	2	2.5	-	-	-	-	-	-	-	-	2	3	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to Databases **6 Lecture Hours**

Introduction to Database, Database users, characteristics and advantages of the database, Database systems, Concepts and architecture-Data models, schemas & instances, Three-Schema architecture & data independence, database languages & interfaces, Centralized and Client/Server Architecture of DBMS.

Unit II: Data Modelling **7 Lecture Hours**

Data Modelling, Using the Entity Relationship (ER) Model, The Enhanced Entity-Relationship (EER) Model: - Entity Set, attributes and their types, Relationship Constraints (including Participation constraints and cardinality ratio), ER Diagrams, constraints and design issues, Reduction of ER and EER diagram to relational schemas. UML Class Diagrams.

Unit III: Relational Database Design and Normalization **7 Lecture Hours**

Relational model Concepts, Relational model constraint & relational database schemas, transactions, and dealing with constraint Violation, DBMS Keys, Relational Algebra, Unary relational operation, Binary relational operations and, relational algebra operations from set Theory, Relational Calculus; and implementation in SQL, Informal Design guideline for relational Schemas, Functional Dependencies, Normal forms based on primary keys, (1NF, 2NF, 3NF & BCNF), lossless join and dependency preserving decomposition, Multivalued dependencies (4NF, 5NF), domain key normal form.
SQL- Queries, Constraints, Form of SQL query, UNION, INTERSECT and EXCEPT, Nested queries, Aggregate Operators, Null values, Complex Integrity constraints in SQL and triggers.

Unit IV: DBMS Architecture, Query Processing and Optimization **7 Lecture Hours**

DBMS Instance, DBMS Internal Memory Structure, Background Processes, Data Types, Roles & Privileges, Introduction to Query Processing, Translating SQL Queries into Relational Algebra, Algorithms for External Sorting, Algorithms for SELECT and JOIN Operations, Algorithms for PROJECT and SET Operations, Implementing Aggregate Operations and Outer Joins.

Unit V: Disk Storage, File Structures, and Indexing 5 Lecture Hours

Introduction, Secondary Storage Devices, Buffering of Blocks and Placing File Records on Disk, Operations on Files, Heap Files, Sorted Files, Hashing Techniques, Parallelizing Disk Access using RAID Technology, Secondary Access Paths, Types of Single-Level Ordered Indexes, Multilevel Indexes, Dynamic Multilevel Indexes Using B-Trees and B+ Trees, Indexes on Multiple Keys

Unit VI: Transaction Management, 7 Lecture Hours

Concurrency Control and Recovery

Introduction to Transaction Processing, Transaction and System Concepts, Desirable Properties of Transactions, Characterizing Schedules based on Recoverability, Characterizing Schedules based on Serializability. Introduction to Concurrency Control, Two Phase Locking Techniques, Concurrency Control on Timestamp Ordering, Validation Concurrency Control Techniques, Granularity of Data items and Multiple Granularity Locking, Recovery Concepts, Recovery Techniques Based on Deferred and Immediate Update, Shadow Paging.

Unit VII: NoSQL Database Management 6 Lecture Hours

Introduction, Need of NoSQL, different NoSQL data models, Introduction to MongoDB, Datatypes, Document Data Model-Creating, Inserting, Updating and Deleting Documents, MongoDB Query Language, Indexing, Aggregation, Sharding in MongoDB, Join Operations, Pagination.

Total lecture Hours 45

Textbooks

1. Ramez Elmasri, and Shamkant B. Navathe, "Fundamentals of Database Systems", 7th Edition, Pearson India, 2017.

2. Raghu Ramakrishnan, "Database Management Systems", 4th Edition, Mcgraw-Hill, 2015.

Reference Books

1. A. Silberschatz, H. F. Korth, and S. Sudershan, "Database System Concepts", 6th ed., McGraw Hill, 2013.

2. Hector Garcia-Molina, Jeffrey D. Ullman, and Jennifer Widom, "Database Systems-The Complete Book", 2nd Edition, Pearson India Pearson Education, 2011.

3. Pramod J. Sadalage, and Marin Fowler, "NoSQL Distilled: A brief guide to merging world of Polyglot persistence", Addison Wesley, 2012.

4. Shannon Bradshaw, Eoin Brazil, and Kristina Chodorow, "MongoDB: The Definitive Guide", 3rd Edition, O'Reilly Media, 2019.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSEG2146	Database Management Systems Lab	0	0	4	2
Total Units to be Covered: 23	Total Contact Hours: 60				
Prerequisite(s):		Syllabus version: 1.0			

Course Objectives

1. To understand the concept of DBMS and ER Modelling.
2. To explain normalization, Query optimization and relational algebra.
3. To apply concurrency control, recovery, security and indexing for real time data.

Course Outcomes

CO 1 Explain the terminologies, features and models of database systems.

CO 2 Apply various disk storage, Indexing and hashing techniques for data storage.

CO 3 Formulate SQL queries using relational algebra and relational calculus.

CO 4 Apply normalization theory to database design.

CO 5 Develop database application design and its implementation including integrity, constraints, transaction management and concurrent control algorithms.

CO 6 Apply No SQL database concepts on real time data.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	3	2	2	-	-	-	-	-	-	-	-	2	3	-
CO 2	-	3	2	2	-	-	-	-	-	-	-	-	2	3	-
CO 3	-	3	2	3	-	-	-	-	-	-	-	-	2	3	-
CO 4	-	3	2	3	-	-	-	-	-	-	-	-	2	3	-
CO 5	-	3	2	2	-	-	-	-	-	-	-	-	2	2	-
CO 6	-	2	1	1	-	-	-	-	-	-	-	-	2	1	-
Average	-	2.8	1.8	2.1	-	-	-	-	-	-	-	-	2	2.5	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

List of Experiments

Experiment 1:

Title: Consider the following set of requirements for a UNIVERSITY database that is used to keep track of students’ transcripts.

1. The university keeps track of each student’s name, student number, Social Security number, current address and phone number, permanent address and phone number, birth date, sex, class (freshman, sophomore, ..., graduate), major department, minor department (if any), and degree program (B.A., B.S., ..., Ph.D.). Some user applications need to refer to the city, state, and ZIP Code of the student’s permanent address and to the student’s last name. Both Social Security number and student number have unique values for each student.

a. Each department is described by a name, department code, office number, office phone number, and college. Both name and code have unique values for each department.
b. Each course has a course name, description, course number, number of semester hours, level, and offering department. The value of the course number is unique for each course.
c. Each section has an instructor, semester, year, course, and section number. The section number distinguishes sections of the same course that are taught during the same semester/year; its values are 1, 2, 3, ..., up to the number of sections taught during each semester.
d. A grade report has a student, section, letter grade, and numeric grade (0,1, 2, 3, or 4).

Design an Entity-Relationship diagram for the mail order database and enter the design using a data-modeling tool such as ERWin/free tool. Specify key attributes of each entity type, and structural constraints on each relationship type. Note any unspecified requirements and make appropriate assumptions to make the specification complete.

Experiment 2

Title. Consider the following set of requirements for a Company database that is used to keep track of employee.

The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations.

a. A department controls a number of projects, each of which has a unique name, a unique number, and a single location.

b. We store each employee's name, Social Security number,2 address, salary, sex (gender), and birth date. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. We keep track of the current number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee (who is another employee).

c. We want to keep track of the dependents of each employee for insurance purposes. We keep each dependent's first name, sex, birth date, and relationship to the employee.

Design an Entity-Relationship diagram for the company database and enter the design using a data-modeling tool such as ERWin/free tool.

Experiment 3

Title: To understand DDL and DML Command

Objective: To understand the concept of designing issue related to the database with creating, populating the tables. To understand the concept of data constraints that is enforced on data being stored in the table. Focus on Primary Key and the Foreign Key.

a. **Create the tables for** Company database as per ER diagram of Exp 2.

TABLE 1: EMPLOYEE
[Fname VARCHAR(15) NOT NULL,
Minit CHAR,
Lname VARCHAR(15) NOT NULL,
Ssn CHAR(9) NOT NULL,
Bdate DATE,
Address VARCHAR(30),
Sex CHAR,
Salary DECIMAL(10,2),
Super_ssn CHAR(9),
Dno INT NOT NULL,
PRIMARY KEY (Ssn),
FOREIGN KEY (Super_ssn) REFERENCES EMPLOYEE(Ssn),
FOREIGN KEY (Dno) REFERENCES DEPARTMENT(Dnumber)
]

TABLE 2: DEPARTMENT
[Dname VARCHAR(15) NOT NULL,
Dnumber INT NOT NULL,
Mgr_ssn CHAR(9) NOT NULL,
Mgr_start_date DATE,
PRIMARY KEY (Dnumber),
UNIQUE (Dname),
FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn));
]

TABLE 3: DEPT_LOCATIONS
(Dnumber INT NOT NULL,
Dlocation VARCHAR(15) NOT NULL,

PRIMARY KEY (Dnumber, Dlocation),
FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber));

TABLE 4: PROJECT
(Pname VARCHAR(15) NOT NULL,
Pnumber INT NOT NULL,
Plocation VARCHAR(15),
Dnum INT NOT NULL,
PRIMARY KEY (Pnumber),
UNIQUE (Pname),
FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber));

TABLE 5: WORKS_ON
(Essn CHAR(9) NOT NULL,
Pno INT NOT NULL,
Hours DECIMAL(3,1) NOT NULL,
PRIMARY KEY (Essn, Pno),
FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),
FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber));

TABLE 6: DEPENDENT
(Essn CHAR(9) NOT NULL,
Dependent_name VARCHAR(15) NOT NULL,
Sex CHAR,
Bdate DATE,
Relationship VARCHAR(8),
PRIMARY KEY (Essn, Dependent_name),
FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn));

b. Insert the following data into their respective tables of Company database.

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

EMPLOYEE

FNAME	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	Smith	123456789	1965-01-09	731 Fondren, Houston TX	M	30000	333445555	5
Franklin	Wong	333445555	1965-12-08	638 Voss, Houston TX	M	40000	888665555	5
Alicia	Zelaya	999887777	1968-01-19	3321 Castle, Spring TX	F	25000	987654321	4
Jennifer	Wallace	987654321	1941-06-20	291 Berry, Bellaire TX	F	43000	888665555	4
Ramesh	Narayan	666884444	1962-09-15	975 Fire Oak, Humble TX	M	38000	333445555	5
Joyce	English	453453453	1972-07-31	5631 Rice, Houston TX	F	25000	333445555	5
Ahmad	Jabbar	987987987	1969-03-29	980 Dallas, Houston TX	M	25000	987654321	4
James	Borg	888665555	1937-11-10	450 Stone, Houston TX	M	55000	null	1

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

WORKS_ON

ESSN	PNO	HOURS
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	null

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
333445555	Alice	F	1986-04-04	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

DEPT_LOCATIONS

DNUMBER	DLOCATION
1	Houston
4	Stafford
5	Bellaire
5	Houston
5	Sugarland

Experiment 4:

Title: To understand and apply the concept of Constraints.

Objective: To understand the concept of data constraints that is enforced on data being stored in the table. Focus on Primary Key and the Foreign Key.

1. Create the tables described below:

Table name: CLIENT_MASTER

Description: used to store client information.

Column name	**data type**	**Size**	**Constraints**
CLIENTNO	Varchar	6	Primary key / first letter must start with 'C'
NAME	Varchar	20	Not Null
ADDRESS 1	Varchar	30	
ADDRESS 2	Varchar	30	
CITY	Varchar	15	
PINCODE	Integer	8	
STATE	Varchar	15	
BALDUE	Decimal	10,2	

Table Name: PRODUCT_MASTER

Description: used to store product information

Column name	**data type**	**Size**	**Attributes**
PRODUCTNO	Varchar	6	Primary Key/ first letter must start with 'P'
DESCRIPTION	Varchar	15	Not Null
PROFITPERCENT	Decimal	4,2	Not Null
UNIT MEASURE	Varchar	10	Not Null
QTYONHAND	Integer	8	Not Null
REORDERL VL	Integer	8	Not Null
SELLPRICE	Decimal	8,2	Not Null
COSTPRICE	Decimal	8,2	Not Null

Table Name: SALESMAN_MASTER

Description: used to store salesman information working for the company.

Column name	**data type**	**Size**	**Attributes**
SALESMANNO	Varchar	6	Primary Key/ first letter must start with 'S'
SALESMANNAME	Varchar	20	Not Null
ADDRESS 1	Varchar	30	Not Null
ADDRESS 2	Varchar	30	
CITY	Varchar	20	
PINCODE	Integer	8	
STATE	Varchar	20	
SALAMT	Real	8,2	Not Null , Cannot be 0
TGTTOGET	Decimal	6,2	Not Null , Cannot be 0
YTDSALES	Double	6,2	Not Null
REMARKS	Varchar	60	

1. **Insert the following data into their respective tables:**
   a) Data for **CLIENT_MASTER** table:

Client no	Name	City	Pincode	State	BalDue

C00001	Ivan bayross	Mumbai	400054	Maharashtra	15000
C00002	Mamta muzumdar	Madras	780001	Tamil nadu	0
C00003	Chhaya bankar	Mumbai	400057	Maharashtra	5000
C00004	Ashwini joshi	Bangalore	560001	Karnataka	0
C00005	Hansel colaco	Mumbai	400060	Maharashtra	2000
C00006	Deepak sharma	Mangalore	560050	Karnataka	0

b) Data for **PRODUCT_MASTER** table:

Product No	Description	Profit percent	Unit measure	Quantity On hand	Recorder Level	Sell Price	Cost Price
P00001	T-Shirt	5	Piece	200	50	350	250
P0345	Shirts	6	Piece	150	50	500	350
P06734	Cotton jeans	5	Piece	100	20	600	450
P07865	Jeans	5	Piece	100	20	750	500
P07868	Trousers	2	Piece	150	50	850	550
P07885	Pull Overs	2.5	Piece	80	30	700	450
P07965	Denim jeans	4	Piece	100	40	350	250
P07975	Lycra tops	5	Piece	70	30	300	175
P08865	Skirts	5	Piece	75	30	450	300

c) Data for **SALESMAN_MASTER** table:

Salesman No	**Name**	**Address 1**	**Address 2**	**City**	**Pin Code**	**State**

S00001	Aman	A/14	Worli	Mumbai	400002	Maharashtra
S00002	Omkar	65	Nariman	Mumbai	400001	Maharashtra
S00003	Raj	P-7	Bandra	Mumbai	400032	Maharashtra
S00004	Ashish	A/5	Juhu	Mumbai	400044	Maharashtr(a

2. **Exercise on retrieving records from a table.**
   a. Find out the names of all the clients.
   b. Retrieve the entire contents of the Client_Master table.
   c. Retrieve the list of names, city and the state of all the clients.
   d. List the various products available from the Product_Master table.
   e. List all the clients who are located in Mumbai.
   f. Find the names of salesman who have a salary equal to Rs.3000.
3. **Exercise on updating records in a table**
   a. Change the city of ClientNo 'C00005' to 'Bangalore'.
   b. Change the BalDue of ClientNo 'C00001' to Rs.1000.
   c. Change the cost price of 'Trousers' to rs.950.00.
   d. Change the city of the salesman to Pune.
4. **Exercise on deleting records in a table**
   a. Delete all salesman from the Salesman_Master whose salaries are equal to Rs.3500.

b. Delete all products from Product_Master where the quantity on hand is equal to 100.

c. Delete from Client_Master where the column state holds the value 'Tamil Nadu'.

5. **Exercise on altering the table structure**
   a. Add a column called 'Telephone' of data type integer to the Client_Master table.
   b. Change the size off SellPrice column in Product _Master to 10, 2.

6. **Exercise on deleting the table structure along with the data**
   a. Destroy the table Client_Master along with its data.

EXPERIMENT-5

Title: To understand and use SQL Sub-Query

Objective: To understand the use of sql subquery.

1. Create the following table.

Supplier-(scode,sname,scity,turnover)

Part-(pcode,weigh,color,cost,sellingprice)

Supplier_Part-(scode,pcode,qty)

2. Populate the table

3. Write appropriate SQL Statement for the following:

1. Get the supplier number and part number in ascending order of supplier number.
2. Get the details of supplier who operate from Bombay with turnover 50.
3. Get the total number of supplier.
4. Get the part number weighing between 25 and 35.
5. Get the supplier number whose turnover is null.
6. Get the part number that cost 20, 30 or 40 rupees.
7. Get the total quantity of part 2 that is supplied.
8. Get the name of supplier who supply part 2.
9. Get the part number whose cost is greater than the average cost.
10. Get the supplier number and turnover in descending order of turnover.

EXPERIMENT-6

Title: Use of Inbuilt functions and relational algebra operation

Objective: To understand the use of inbuilt function and relational algebra with sql query.

Write and execute the following queries using the Relational Algebra on the COMPANY database schema.

1. Retrieve the names of all employees in department 5 who work more than 10 hours
2. per week on the ‘ProductX’ project.
3. List the names of all employees who have a dependent with the same first name as
4. themselves.
5. Find the names of employees who are directly supervised by ‘Franklin Wong’.
6. Retrieve the names of employees who work on every project.
7. Retrieve the names of employees who do not work on any project.
8. Retrieve the names and addresses of all employees who work on at least one project
9. located in Houston but whose department has no location in Houston.
10. Retrieve the last names of all department managers who have no dependents.

EXPERIMENT-7

Title: Use of Inbuilt functions and relational algebra operation

Objective: To understand the use of inbuilt function and relational algebra with sql query.

1. **Create the following two tables (EMP and DEPT)**

EMP TABLE

EMPNO ENAME JOB MGR HIREDATE SAL COMM DEPTNO
---------- ---------- --------- ---------- --------- ---------- ---------- ----------
7369 SMITH CLERK 7902 17-DEC-80 500 800 20
7499 ALLEN SALESMAN 7698 20-FEB-81 1600 300 30
7521 WARD SALESMAN 7698 22-FEB-81 1250 500 30
7566 JONES MANAGER 7839 02-APR-81 2975 20

7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

DEPT TABLE

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Write the Queries for the following using In-built functions.

1. Retrieve average salary of all employees.
2. Retrieve the number of employees.
3. Retrieve distinct number of employee.

4. Retrieve total salary of employee group by job.
5. Display the employee information with maximum salary.
6. Find the highest paid employee in department 10.
7. List the emps whose sal is equal to the average of max and minimum.
8. List the emps who joined in the company on the same date.
9. Display the employee names in upper and lower case.
10. find the date of 3 days later from hiredate.

EXPERIMENT-8

Title: Use of different SQL clauses and join

Objective: To understand the use of group by and having clause and execute the SQL commands using JOIN

1. Consider the following schema:

Student (sid, sname, age)

Match (mid, mname, venue)

Play (sid, mid, day(date))

2. Populate all the tables.
3. nFind all information of students who have played match number B10.
4. Find the name of matches played by Amit.
5. Find the names of students who have played a match in Delhi.
6. Find the names of students who have played at least one match.
7. Find the ids and names of students who have played two different matches on the same day.
8. Find the ids of students who have played a match in Delhi or Mumbai.
9. Find the average age of students.

EXPERIMENT-9

Title: To understand the concepts of Views.

Objective: Students will be able to implement the concept of views.

1. Create table of table name: EMPLOYEES and add 6 rows

Column Name	Data Type	Width	Attributes
Employee_id	Character	10	PK
First_Name	Character	30	NN
Last_Name	Character	30	NN
DOB	Date		
Salary	Number	25	NN
Department_id	Character	10	

2. Execute the following view related queries:

1) Create View of name emp_view and the column would be Employee_id, Last_Name, salary and department_id only.
2) Insert values into view(remove the NOT NULL constraint and then insert values):
3) Modify, delete and drop operations are performed on view.
4) Creates a view named salary_view. The view shows the employees in department 20 and their annual salary.

EXPERIMENT-10

Title: Create the following views in SQL on the COMPANY database schema presented in Experiment 2.

1. A view that has the department name, manager name, and manager salary for every department.
2. A view that has the employee name, supervisor name, and employee salary for each employee who works in the 'Research' department.
3. A view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project.
4. A view that has the project name, controlling department name, number of employees, and total hours worked per week on the project for each project with more than one employee working on it.

EXPERIMENT-11

Title: To understand the concepts of Index.

Objective: Students will be able to implement the concept of index.

Create table of table name: EMPLOYEES and add 6 rows

Column Name	Data Type	Width	Attributes
Employee_id	Character	10	PK
First_Name	Character	30	NN
Last_Name	Character	30	NN
DOB	Date		
Salary	Number	25	NN
Department_id	Character	10	

1. Execute the following index related queries:

1) Create an index of name employee_idx on EMPLOYEES with column Last_Name, Department_id
2) Find the ROWID for the above table and create a unique index on employee_id column of the EMPLOYEES.
3) Create a reverse index on employee_id column of the EMPLOYEES.
4) Create a unique and composite index on employee_id and check whether there is duplicity of tuples or not.
5) Create Function-based indexes defined on the SQL functions UPPER(column_name) or LOWER(column_name) to facilitate case-insensitive searches(on column Last_Name).
6) Drop the function based index on column Last_Name.

EXPERIMENT-12

Title: To understand the concepts of Sequence.

Objective: Students will be able to implement the concept of sequence.

1) Create a sequence by name EMPID_SEQ starting with value 100 with an interval of 1.

2) Write a SQL command for finding the current and the next status of EMPID_SEQ.
3) Change the Cache value of the sequence EMPID_SEQ to 20 and maxvalue to 1000.
4) Insert values in employees table using sequences for employee_id column.
5) Drop sequence EMPID_SEQ.
6) Create a sequence called REVERSE to generate numbers in the descending order from 10000 to 1000 with a decrement of 5.

EXPERIMENT-13

Title: To understand the concepts of PL/SQL programming.

Objective: Students will be able to implement the basic concepts of Pl/SQL.

1) Write a PL/SQL code to accept the value of A, B & C display which is greater.
2) Using PL/SQL Statements create a simple loop that display message "Welcome to PL/SQL Programming" 20 times.
3) Write a PL/SQL code block to find the factorial of a number.
4) Write a PL/SQL program to generate Fibonacci series.
5) Write a PL/SQL code to fund the sum of first N numbers

EXPERIMENT-14

Title: To understand the concepts of function and procedure in PL/SQL.

Objective: Students will be able to implement the Pl/SQL programs using function and procedure.

1) Implement the above experiments of PL/SQL using functions and procedures.

EXPERIMENT-15

Title: To understand the concepts of implicit and explicit cursor.

Objective: Students will be able to implement the concept of implicit and explicit cursor.

1. Using implicit cursor update the salary by an increase of 10% for all the records in EMPLOYEES table, and finally display how many records have been updated. If no records exist display the message “**No Change**”.
2. Using explicit cursor fetch the employee name, employee_id and salary of all the records from EMPLOYEES table.
3. Using explicit cursor Insert the records from EMPLOYEES table for the columns employee_id, Last_Name and salary for those records whose salary exceeds 2500 into a new table TEMP_EMP

EXPERIMENT-16

Title: To understand the concepts of Trigger.

Objective: Students will be able to implement the concept of trigger.

CUSTOMER Table:

ID	NAME	AGE	ADDRESS	SALARY
1	Ramesh	32	Ahmedabad	2000.00
2	Khilan	25	Delhi	1500.00
3	Kaushik	23	Kota	2000.00
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
6	Komal	22	MP	4500.00

1) Create a row level trigger for the customers table that would fire for INSERT or UPDATE or DELETE operations performed on the CUSTOMERS table. This trigger will display the salary difference between the old values and new values.

EXPERIMENT-17

Title: To understand the concepts of Trigger.

Objective: Students will be able to implement the concept of trigger.

1. CREATE TRIGGER SALARY_VIOLATION BEFORE INSERT OR UPDATE OF SALARY, SUPERVISOR_SSN ON EMPLOYEE of experiment 3

EXPERIMENT-18

Title: To understand the concepts of NoSQL Database

Objective: Students will be able to implement the concept of NoSQL Database MongoDB.

1. Write a MongoDB query to display all the documents in the collection hotel.
2. Write a MongoDB query to display the fields hotel_id, name, Borough and cuisine for all the documents in the collection hotel.
3. Write a MongoDB query to display the fields hotel_id, name, Borough and cuisine, but exclude the field _id for all the documents in the collection hotel.
4. Write a MongoDB query to display the fields hotel_id, name, Borough and zip code, but exclude the field _id for all the documents in the collection hotel. hotel
5. Write a MongoDB query to display all the hotel which is in the Borough Bronx.

EXPERIMENT-19

Title: To understand the concepts of NoSQL Database

Objective: Students will be able to implement the concept of NoSQL Database MongoDB.

1. Write a MongoDB query to display the next 5 hotels after skipping first 5 which are in the Borough Bronx.
2. Write a MongoDB query to find the hotels that achieved a score, more than 80 but less than 100.
3. Write a MongoDB query to find the hotels which locate in latitude value less than -95.75
4. Write a MongoDB query to find the hotels that do not prepare any cuisine of 'American' and their grade score more than 70 and latitude less than -65.754168.

EXPERIMENT-20

Title: To understand the concepts of NoSQL Database

Objective: Students will be able to implement the concept of NoSQL Database MongoDB.

1. Write a MongoDB query to arrange the name of the cuisine in ascending order and for that same cuisine Borough should be in descending order.
2. Write a MongoDB query to know whether all the addresses contains the street or not.
3. Write a MongoDB query which will select all documents in the hotels collection where the coord field value is Double.
4. Write a MongoDB query which will select the hotel Id, name and grades for those hotels which returns 0 as a remainder after dividing the score by 7.
5. Write a MongoDB query to find the hotel name, Borough, longitude and attitude and cuisine for those hotels which contains 'mon' as three letters somewhere in its name.
6. Write a MongoDB query to find the hotel name, Borough, longitude and latitude and cuisine for those hotels which contain 'Mad' as first three letters of its name.

EXPERIMENT-21

Title: To understand the concepts of NoSQL Database

Objective: Students will be able to implement the concept of NoSQL Database MongoDB.

1. Write a MongoDB query to find the hotels which do not prepare any cuisine of 'American' and achieved a score more than 70 and located in the longitude less than -65.754168.
2. Write a MongoDB query to find the hotels which do not prepare any cuisine of 'American ' and achieved a grade point 'A' not belongs to the Borough Brooklyn. The document must be displayed according to the cuisine in descending order.
3. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which contain 'ces' as last three letters for its name.

4. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which contain 'Reg' as three letters somewhere in its name.
5. Write a MongoDB query to find the hotels which belong to the Borough Bronx and prepared either American or Chinese dish.
6. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which belong to the Borough Staten Island or Queens or Hyatt.
7. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which are not belonging to the Borough New Delhi or Queens or Hyatt.
8. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which achieved a score which is not more than 10.
9. Write a MongoDB query to find the hotel Id, name, Borough and cuisine for those hotels which prepared dish except 'American' and 'Chinees' or hotel's name begins with letter 'Wil'.
10. Write a MongoDB query to find the hotel Id, name, and grades for those hotels which achieved a grade of "A" and scored 11 on an ISODate "2014-08-11T00:00:00Z" among many of survey dates.

EXPERIMENT-22

Mini Project – On SQL

EXPERIMENT-23

Mini Project- On NoSQL

Total Lab hours 60

Textbooks

1. Ivan Bayross, "SQL, PL/SQL – The Programming Language of Oracle", 4th Revised Edition, 2010.

2. Kristina Chodorow, and Michael Dirolf, "MongoDB: The Definitive Guide", O'Reilly Media Inc., 2010.

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50	50

Course Code	Course name		L	T	P	C
CSEG2021	Design and Analysis of Algorithms		3	0	0	3
Total Units to be Covered: 6		Total Contact Hours: 45				
Prerequisite(s):	Data structures and algorithms		Syllabus version: 1.0			

Course Objectives:

The objectives of this course are as follows:

1. Define the fundamental concepts, definitions and terminologies related to algorithms, data structures, and algorithm analysis.
2. Understand the principles behind various algorithm design techniques.
3. Apply algorithms analysis techniques to evaluate the efficiency and asymptotic performance of algorithms in terms of time and space complexity.
4. Analyse and compare different algorithmic solutions for the same problem, considering their efficiency, correctness, and suitability for specific scenarios.

Course Outcomes

The outcomes of this course are as follows:

CO1. Demonstrate a solid understanding of fundamental concepts, terminologies, and principles related to algorithms, data structures, and algorithm analysis.

CO2. Apply algorithmic design techniques to solve real-world problems.

CO3. Compare and contrast multiple algorithmic approaches for the same problem, considering their efficiency, correctness, and practicality.

CO4. Select problem-solving strategies and algorithmic thinking to tackle new and challenging problem domains and classify the algorithms in different classes.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO 9	PO1 0	PO 11	PO12	PSO1	PSO2	PSO3
CO 1	3	3	1	2	-	-	-	-	-	-	-	-	3	1	-
CO 2	3	3	2	2	-	-	-	-	-	-	-	-	3	1	-
CO 3	2	2	2	1	-	-	-	-	-	-	-	-	3	1	-
CO 4	2	3	2	1	-	-	-	-	-	-	-	-	3	1	-

Average	2.5	2.75	1.75	1.5	-	-	-	-	-	-	-	-	3	1	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to Algorithms **9 Lecture Hours**

Algorithm, Characteristics of algorithm, Distinct area of study of algorithm, Different algorithm design techniques, Algorithm analysis, Growth of an algorithm, Asymptotic notations, Recurrence relation, Solving recurrence relation by iteration, substitution, recursion tree, master theorem method.

Unit II: Algorithm design paradigm: Divide and Conquer **7 Lecture Hours**

The divide and conquer paradigm, Analysis of Binary search, Merge sort, Quick sort, Strassen Method of Matrix Multiplication, Maximum subarray problem, Powering number, Celebrity problem.

Unit III: Algorithm design paradigm: Greedy Method **7 Lecture Hours**

Greedy approach design paradigm, Knapsack problem, Activity selection problem, Huffman encoding, Interval partitioning problem, Dijkastra algorithm for single source shortest path problem, Prim's and Kruskal algorithm for finding minimum cost spanning tree.

Unit IV: Algorithm design paradigm: Dynamic Programming **7 Lecture Hours**

Dynamic programming design paradigm, 0/1 Knapsack problem, Matrix chain multiplication problem, longest common subsequence problem, Optimal binary search problem, Bellman ford algorithm for single source shortest path problem, Travelling salesman problem, Difference between divide and conquer, greedy and dynamic programming algorithm design approach, Floyd warshall algorithm for all pair shortest path problem

Unit V: Algorithm design paradigm: Backtracking and Branch & Bound **7 Lecture Hours**

Introduction to backtracking and branch & bound approach, backtracking based problems: N Queen problem, Sum of Subset problem, 0/1 Knapsack problem, Branch & Bound based problems: FIFO, LIFO, & LC branch & bound, 0/1 Knapsack problem, Travelling salesperson problem.

Unit VI: Maximum Flow and String-Matching Problems **8 Lecture Hours**

Flow networks: Ford- Fulkerson method, Maximum bipartite matching, Modulo Representation of integers/polynomials: Chinese Remainder Theorem, String Matching: The naive string-matching algorithm, The Rabin-Karp algorithm, String matching with finite automata, The Knuth-Morris-Pratt algorithm, Different classes of problems: P, NP, NP Complete, NP Hard, reducibility property

Total lecture Hours 45

Textbooks

1. T. H. Cormen, C. F. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms", 4th Edition, MIT Press, 2022.

2. Ellis Horowitz, Sartaj Sahni, and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Galgotia Publication, 2010.

Reference Books

1. Jon Kleinberg, and Eva Tardos, "Algorithm Design", Addison Wesley, 2005.

2. A. V. Aho, J. Hopcroft, and J. D. Ullman, "The Design and Analysis of Algorithms", Addison-Wesley, 2002.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSEG2121	**Design and Analysis of Algorithms Lab**	0	0	2	1
Total Units to be Covered: 15	**Total Contact Hours: 30**				
Prerequisite(s):	Data structures and algorithms Lab	**Syllabus version: 1.0**			

Course Objectives

The objectives of this course are as follows:

1. Apply various algorithmic strategies, such as greedy algorithms, divide and conquer, and dynamic programming, to solve problems.
2. Analyse the time complexity, space complexity, and performance characteristics of algorithms.

Course Outcomes

The outcomes of this course are as follows:

CO1. Select ideal design approach based on the problem.

CO2. Critically analyse and evaluate algorithms based on their time complexity, space complexity, and performance characteristics.

CO3. Compare and contrast multiple algorithmic approaches for the same problem, considering their efficiency, correctness, and practicality.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	2	2	2	-	-	-	-	-	-	-	-	3	-	-
CO 2	2	2	2	2	-	-	-	-	-	-	-	-	3	-	-
CO 3	2	2	2	2	-	-	-	-	-	-	-	-	3	-	-
Average	2	2	2	2	-	-	-	-	-	-	-	-	3	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

List of Experiments

Experiment 1-4: Divide and Conquer Approach

- Implement the iterative and recursive Binary search tree and compare their performance.
- Implement divide and conquer based merge sort and quick sort algorithms and compare their performance for the same set of elements.
- Compare the performance of Strassen method of matrix multiplication with traditional way of matrix multiplication.

Experiment 5-9: Greedy & Dynamic Programming Approach

- Implement the activity selection problem to get a clear understanding of greedy approach.
- Get a detailed insight of dynamic programming approach by the implementation of Matrix Chain Multiplication problem and see the impact of parenthesis positioning on time requirements for matrix multiplication.
- Compare the performance of Dijkastra and Bellman ford algorithm for the single source shortest path problem.
- Through 0/1 Knapsack problem, analyze the greedy and dynamic programming approach for the same dataset.

Experiment 10-13: Backtracking and Branch & Bound Approach

- Implement the sum of subset and N Queen problem.
- Compare the Backtracking and Branch & Bound Approach by the implementation of 0/1 Knapsack problem. Also compare the performance with dynamic programming approach.

Experiment 14-15: String Matching Problems

- Compare the performance of Rabin-Karp, Knuth-Morris-Pratt and naive string-matching algorithms.

Total Lab hours 15

Textbooks

1. T. H. Cormen, C. F. Leiserson, R. L. Rivest, and C. Stein, "Introduction to Algorithms", 4th Edition, MIT Press, 2022.

2. Ellis Horowitz, Sartaj Sahni, and Sanguthevar Rajasekaran, "Fundamentals of Computer Algorithms", Galgotia Publication, 2010.

Reference Books

1. Jon Kleinberg, and Eva Tardos, "Algorithm Design", Addison Wesley, 2005.

2. A. V. Aho, J. Hopcroft, and J. D. Ullman, "The Design and Analysis of Algorithms", Addison-Wesley, 2002.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50	50

Course Code	Course name	L	T	P	C
CSEG2006	Discrete Mathematical Structures	3	0	0	3
Total Units to be Covered: 7	Total Contact Hours: 45				
Prerequisite(s):	Advanced Engineering Mathematics – 2	Syllabus version: 1.0			

Course Objectives

To develop mathematical reasoning skills that equip the students with ideas and techniques, required in Discrete Mathematics and its Applications, necessary for understanding and practicing the art and science of computing.

Course Outcomes

On completion of this course, the students will be able to:

CO1. Develop a comprehensive understanding of discrete structures, viz., mathematical logic, propositions, connectives, arguments, sets, functions, relations, etc., that build the foundation of computer science.

CO2. Recognize the applicability of the basic notions of number theory and counting principles such as modular arithmetic, congruence equations, Pigeonhole principle, principle of inclusion and exclusion, etc., in solving complex problems in diverse domains related to cryptography and internet security.

CO3. Illustrate the basic terminologies in graph theory which are used in design of algorithms for a wide range of applications related to shortest path problems, graph coloring problems, etc.

CO4. Use the framework of network science in building models for various problems arising in communication, social and biological networks.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	3	1	1	-	-	-	-	-	-	-	1	-	-	-
CO 2	3	3	1	1	-	-	-	-	-	-	-	1	-	-	-
CO 3	3	2	1	1	-	-	-	-	-	-	-	1	-	-	-

CO 4	3	2	1	1	-	-	-	-	-	-	-	1	-	-	-
Average	3	2.5	1	1	-	-	-	-	-	-	-	1	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Motivation **1 Lecture Hour**

Introduction to Discrete Mathematics and its significance in Computer Science. Why learn Discrete Mathematics?

Unit II: Mathematical Logic **6 Lecture Hours**

Mathematical Logic, Proposition, Connectives, Tautologies, and Contradictions, Logical Equivalences, Normal forms, and applications. Predicates and Quantifiers, Interpretation of an Argument, and its validity. Recapitulation of the Unit.

Unit III: Relations **6 Lecture Hours**

Review of Set Theory, Functions, one-one onto Functions, Relations and their properties, n-ary Relations, Illustrations through Applications, Representations, Closures, Partial Ordering, Decomposition Theorems for Partial Orders, Posets and Lattices. Recapitulation of the Unit.

Unit IV: Number Theory **5 Lecture Hours**

Modular Arithmetic, Primes, Fundamental Theorem of Arithmetic, GCD/LCM, Euclidean Algorithm, Bézout's Identity, Solving Congruences, Linear Diophantine Equation, Chinese Remainder Theorem, Fermat's Little Theorem, Illustration through Examples in Cryptography and Internet Security, Discrete Log, Orthogonal Latin Squares. Recapitulation of the Unit

Unit V: Induction and Basic Counting Principles **10 Lecture Hours**

Induction, Strong Induction, Well-ordering Principle, Recursive Definitions and Structural Induction, Pigeonhole Principle, Binomial and Multinomial Coefficients and

Identities, Elementary Applications to Discrete Probability, Recurrence Relations and Equations, Generating Function Techniques, Principles of Inclusion and Exclusion and their Applications. Recapitulation of the Unit.

Unit VI: Graph Theory **10 Lecture Hours**

Graphs and Graph Models, Basic Terminology and Special Types of Graphs, Representing Graphs and Graph Isomorphism, Vertex and Edge Connectivity, Adjacency and Incidence Matrices, Konigsberg Bridge Problem, Euler Graph, Travelling Salesman Problem, Hamiltonian Graphs, Shortest-Path Problems, Planar Graph, Euler's Formula, Graph Coloring, Welch Powell Algorithm, Decomposition Algorithm, Random Graphs. Recapitulation of the Unit.

Unit VII: Network Science **7 Lecture Hours**

Illustrative Examples of Real Networks, Network and Graph, Degree Distribution, Power-Law and Scale Free Property, Sparsity in Real Networks, Paths and Distances, Network Diameter, Connectedness and Clustering Coefficient, Random Network, Evolution of a Random Network, Small World Network: Watts-Strogatz Model, Emergence of Network Science. Recapitulation of the Unit.

Total lecture Hours 45

Textbooks

1. Rosen, K. H., "Discrete Mathematics and its Applications", 7th Edition, McGraw Hill, 2017. (Units - II, III, IV, VI)

2. R. C. Bose, and B. Manvel, "Introduction to Combinatorial Theory", Wiley Series in Probability and Mathematical Statistics, 1984. (Units - V)

3. Albert-László Barabási, "Network Science", 1st Edition, Cambridge University Press, 2016. (Unit-VI, VII)

Reference Books

1. W. K. Grassmann, and J-P Tremblay, "Logic and Discrete Mathematics: A Computer Science Perspective", 1st Edition, Pearson, 1995.

2. E. Lehman, F. T. Leighton, and A. R. Meyer, "Mathematics for Computer Science", Samurai Media Limited, 2017.

3. D. B. West, "Introduction to Graph Theory", 2nd Edition, Pearson Singapore, 2000.

4. J. L. Hein, "Discrete Structure, Logic, and Computability", 3rd Edition, Jones and Barlett Publishers, 2010.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSEG2060	Operating Systems		3	0	0	3
Total Units to be Covered: 6		Total Contact Hours: 45				
Prerequisite(s):	Computer Organization and Architecture.	Syllabus version: 1.0				

Course Objectives

To equip students with a deep understanding of operating system design and implementation, enabling them to analyze, evaluate, and apply concepts such as process management, concurrency, memory management, and storage in real-world scenarios.

Course Outcomes

CO1: Demonstrate a comprehensive understanding of operating systems.

CO2: Evaluate and analyze process and thread scheduling techniques, discerning their benefits and challenges.

CO3: Demonstrate an understanding of inter-process communication (IPC) mechanisms, process synchronization and deadlocks.

CO4: Evaluate and analyze memory and storage management techniques.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	1	1	1	1	-	1	-	-	-	-	2	3	-	-
CO 2	1	2	1	3	3	-	1	-	-	-	-	2	3	-	-
CO 3	2	2	1	2	3	-	1	-	-	-	-	2	3	-	-
CO 4	2	2	1	2	2	-	1	-	-	-	-	2	3	-	-
Average	1.5	1.75	1	2	2.25	-	1	-	-	-	-	2	3	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to Operating System 6 Lecture Hours

Computer Hardware Review; Computer System; Introduction to Operating System: Definition, Operating System view, History, Types of Operating Functions of Operating System, Services of Operating System, Computing Environments, Virtualization and Containerization, Operating System Structures, Operating System Operations, System boot. System Calls, Types of System Calls (Windows and Unix System Calls examples), Open Source Operating Systems

Unit II: Process and Thread Management 8 Lecture Hours

Process: Program and Process concept, Process in memory, Process Control Block, Process States, Process Context Switching, Process Scheduling Queues, Process Schedulers, Process Context Switching, Process Scheduling Criteria, Process Scheduling: Non preemptive and Preemptive Schedulers, FCFS, Shortest Job First, Shortest Remaining Time First, Non Preemptive Priority scheduling and Preemptive Scheduling, Priority Round Robin, Multilevel Queue, Multiple Feedback Queue, Real Time scheduling : Rate Monotonic, Earliest Deadline First; Operations on processes: Creation and Termination

Threads: Threads and its benefits, Multi-threading models, Kernel Level thread, user level thread and hybrid threads, Thread Scheduling: Content Scope, Pthread Scheduling, Threading Issues

Case study: Process Management in Linux

Unit III: Inter Process Communication and Synchronization 9 Lecture Hours

Inter Process Communication (IPC), IPC mechanisms: Shared Memory and Message Passing (Shared Memory, Pipes and Named pipes in Linux), Critical Section Problem, Race Condition, Producer Consumer Problem, Solution to Critical section Problem: Hardware and Software Solutions, Software Solutions: Semaphores: Counting semaphore, Binary semaphore, Monitors, Algorithm 1, Algorithm 2, Algorithm

3/Peterson Solution, Bakery Algorithm, Classic process synchronization problems (case studies).

Unit IV: Deadlock Handling **6 Lecture Hours**

Deadlock, Deadlock characterization: Necessary Conditions for Deadlock, Resource Allocation Graph; Methods for Handling Deadlocks: Deadlock Prevention, Deadlock Avoidance: Safe State, Resource Allocation Graph Algorithm, Bankers Algorithm; Deadlock Detection; Recovery from deadlock: process Termination and Resource Preemption

Unit V: Memory Management **9 Lecture Hours**

Memory protection, Address binding, Logical versus Physical Address Space, Dynamic Loading, Dynamic Linking, Swapping, Memory Management Strategies: Contiguous and Non-Contiguous; Contiguous memory management: static and dynamic: First Fit, Best Fit, Worst Fit, Buddy System, Internal Fragmentation, External Fragmentation, Compaction, Non Contiguous memory management: Paging, Paging Hardware Support, Structure of Page Table: Hierarchical paging, Hashed page tables and Inverted page tables, Virtual memory, Demand Paging, Page Fault, handling of Page Fault, Page Replacement, Page Replacement Algorithms, Belady's anamoly, Allocation of frames, Thrashing, Segmentation: concept, Segmentation hardware, Segmentation with Paging

Case Study: Memory Management in Linux

Unit VI: Storage Management **7 Lecture Hours**

File concepts, File system structure, File attributes, File operations, File types, File access method, File system mounting, Directory, Different logical structure of directories, Disk structure, Disk allocation methods: contiguous, linked and indexed, Free space management, Disk scheduling algorithms.

Case Studies: File System in Linux and Windows

Total lecture Hours 45

Textbooks

1. Abraham Silberschatz, Peter B. Galvin, and Greg Gagne, "Operating System Concepts", 10th Edition, John Wiley & Sons, 2018.

2. Andrew S. Tanenbaum and Herbert Bos, "Modern Operating Systems", 4th Edition, Pearson, 2021.

3. William Stallings, "Operating Systems: Internals and Design Principles", 9th Edition, Pearson, 2021.

Reference Books

1. Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau, "Operating Systems: Three Easy Pieces", 1st Edition, CreateSpace Independent Publishing Platform, 2018.

2. J. Archer Harris, "Schaum's Outline of Operating Systems", 1st Edition, McGraw-Hill Education, 2002.

3. Garry J Nutt, "Operating System – A modern perspective", 2nd Edition, Addison Wesley, 2002.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
	Exploratory-1		3	0	0	3
Total Units to be Covered:		Total Contact Hours:				
Prerequisite(s):			Syllabus version: 1.0			

Exploratory courses offered by different schools , student has a choice to opt desired course from the available tracks .

Course Code	Course name		L	T	P	C
SLLS0201	Design Thinking		1	1	0	2
Total Units to be Covered: 04		Total Contact Hours: 30				
Prerequisite(s):			Syllabus version: 1.0			

Course Objectives

The objectives of this course are to:

1. Understand human centered design/problem solution
2. explore and apply design thinking process by using tools that are collaborative, innovative and effective
3. develop a framework for solving complex problems
4. Learning by doing, engaging, exploring and experimenting

Course Outcomes

CO 1: Understand the concepts of design thinking as a human-centered approach for problem solving in everyday life

CO 2: Use and Analyze primary market research, product ideation, prototyping, and fidelity mapping.

CO 3: Demonstrate the ability and openness to innovate, experiment, and generate ideas for solving complex problems.

CO 4: Learn how to incorporate user feedback into iterations to refine and enhance designs and how to embrace an iterative mind-set for design improvement

CO 5: Practice and Employ design-led strategy for mitigating day-to-day problems

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	-	1	1	-	3	3	1	1	-	-	-
CO 2	-	-	-	-	-	1	1	-	2	3	1	1	-	-	-
CO 3	-	-	-	-	-	-	1	1	1	3	1	1	-	-	-
CO 4	-	-	-	-	-	1	1	1	2	3	-	1	-	-	-
CO5	-	-	-	-	-	-	-	1	3	1	-	3	-	-	-

Average	-	-	-	-	-	.6	.8	.6	2.2	2.6	.6	1.4	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Innovation through Design: Think, Make, Break, Repeat 9 Lecture Hours

Introduction to design thinking, its importance and application in life through theory and reallife examples from industry and different walks of life.

Unit II: The Design Thinking Approach 7 Lecture Hours

Exploring the principles, mind-set, and process of design thinking. Understanding the importance of empathy and placing the user at the centre in the design process and learning techniques for conducting user-centered research for creating solutions that address users' needs and pain points

Unit III: Problem Framing 7 Lecture Hours

Techniques for identifying and framing the right problem to solve through Design Thinking. Strategies for generating creative ideas and facilitating effective brainstorming sessions to explore a wide range of possibilities.

Unit IV: The Designer Mentality 7 Lecture Hours

Understanding the iterative design process and exploring prototyping methods and techniques for rapidly building and testing prototypes to gather feedback, iterate on designs, refine and improve solutions. Applying Design Thinking principles to create intuitive and delightful user experiences.

Total lecture Hours 30

Textbooks

1. Burnett, B., & Evans, D. (2023). Designing your new work life. Vintage.

2. Burnett, B., & Evans, D. (2022). Designing your life: How to build a well-lived, joyful life. Alfred A. Knopf.

Reference Books

1. Burnett, W., & Evans, D. J. (2020). Designing your work life how to thrive and change and find happiness at work. Chatto & Windus, an imprint of Vintage.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	COURSERA QUIZ	CLASSROOM PARTICIPATION	COURSE QUIZ	Total
Weightage (%)	30	20	50	100

SEMESTER IV

Course Code	Course name		L	T	P	C
CSEG2065	**Data communication and Networks**		3	0	0	3
Total Units to be Covered: 5		**Total Contact Hours: 45**				
Prerequisite(s):	Digital Electronics		**Syllabus version: 1.0**			

Course Objectives

The objectives of this course are as follows:

1) Understand the basic components and functions of computer networks, including network topologies, protocols, and networking devices.
2) Understand need of layered architecture and differentiate OSI and TCP/IP
3) Gain an understanding of error and flow control techniques on communication channels.
4) Explore routing algorithms and its application.
5) Get a brief idea about network analysis tools (Wireshark, NMAP).

Course Outcomes

The outcomes of this course are as follows:

CO1: Evaluate network devices functionality and network command significance.

CO2: Evaluate and address problems of error control, flow control, and channel access.

CO3: Analyze and adopt fundamental workings of routing algorithms.

CO4: Create solutions for recent challenges in large-scale networks.

CO5: Apply knowledge of network traffic analysis tool to investigate network activities.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	2	2	3	-	-	-	-	2	-	-	3	-	-	-
CO 2	-	2	3	2	-	-	-	-	-	-	-	3	2	-	-
CO 3	2	3	3	3	2	2	-	-	2	-	-	3	3	2	-
CO4	3	3	3	3	2	-	-	2	2	-	2	3	2	3	-

CO5	2	3	3	3	3	3	**-**	2	2	2	3	3	3	3	**-**
Average	1.8	2.6	2.8	2.8	1.4	1	**-**	0.8	1.6	0.4	1	3	2	1.6	**-**

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Basic of Computer Networking and Technologies **9 Lecture Hours**

Introduction to Computer Networking Concepts: Layered Network Protocol Architectures (OSI, TCP/IP); LAN, WAN, MAN, PAN, LAN Topologies; Connectivity Devices and Cable Types; Ethernet, Gigabit Ethernet (GbE); Circuit Switching, Message Switching, and Packet Switching; WiMAX, 5G and beyond, Cellular Technology, Communication Channels and performance metrics.

Unit II: Data Link Layer - Part 1 **9 Lecture Hours**

Logical Link Control (LLC) sub-layer: Framing, Data Communication Character Codes, Error Control: Error Detection (Redundancy Checking: VRC, Checksum, LRC, CRC); Retransmission, Error Correction: Forward Error Correction (Hamming Code), Character Synchronization, Reliable transmission and Automatic Repeat Request (ARQ) protocols including Stop-and-Wait, Go-back-N, Selective Repeat; Performance analysis of ARQ protocols; Example protocols such as HDLC and PPP.

Unit III: Data Link Layer - Part 2 **8 Lecture Hours**

Medium Access Control (MAC) sub-layer: Channel Allocation Problems, Multiple Access Protocols and Types: TDMA, FDMA, CSMA, CSMA/CD, CSMA/CA protocols; Hidden Node and Exposed Node Problems, Performance analysis; Shared and Switched Ethernet; IEEE Standards 802.3 & 802.11, 10-Gigabit Ethernet.

Unit IV: Network Layer **10 Lecture Hours**

Network Layer Design Issues, Network Address Translation, Internet Protocol (IP): IPv4 and IPv6 addressing; IP Addressing Techniques: Classful Addressing, Classless Addressing, Network and Host Identification, Loopback Address, Broadcast Address, Address Masking; Networks and Subnetworks: Subnetting, Subnet Mask, Supernetting; Network-Layer Protocols: ARP, RARP, IP datagram; Internetworking: Routing and Routing protocols (distance-vector and link-state); Interior and Exterior Gateway Protocol concepts; Routing Algorithms including Dijkstra's algorithm and distributed Bellman-Ford algorithm; Example protocols: OSPF, RIP, BGP, Encapsulation and Tunneling, Congestion Control, Quality of Service, Introduction of Wireshark Tool.

Unit V: Transport Layer **9 Lecture Hours**

Introduction and Transport-Layer Services, Port Address, Socket Address; Internet Transport Protocols: UDP, Introduction to UDP, Remote Procedure call, Real-time Transport Protocols; Internet Transport Protocols: TCP, service model, TCP protocol, TCP segment header, TCP Connection establishment, TCP Connection Release, TCP Connection management modeling, TCP sliding window, TCP Timer management, TCP Congestion control; Performance Issues: Performance problems in computer networks, Network Performance Management, Host Design for fast networks; Fast segment processing, Header compression, protocols for long Fat networks; Virtual Private Network (VPN); Introduction of Nmap Tool.

Total lecture Hours 45

Textbooks

1. James F. Kurose, and Keith W.Ross, "Computer Networking : A Top-Down Approach", 8th Edition, Pearson, 2022.

2. W. Tomasi, "Introduction to data communications and networking", 5th edition, Prentice-Hall, Inc., 2008.

Reference Books

1. Walter Goralski, "The illustrated network: how TCP/IP works in a modern network", 2nd Edition, Morgan Kaufmann, 2017.

2. Andrew S. Tanenbaum, "Computer Networks", 5th Edition, Pearson Education, 2023.

3. L. L. Peterson, and B. S. Davie, "Computer networks: a systems approach", 6th Edition, Elsevier, 2020.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSEG2165	Data communication and Networks Lab	0	0	2	1
Total Units to be Covered: 12	Total Contact Hours: 30				
Prerequisite(s):		Syllabus version: 1.0			

Course Objectives

The objectives of this course are as follows:

1. Understand the basic components and functions of computer networks, including network topologies, protocols, and networking devices.
2. Gain an understanding of error and flow control techniques on communication channels.
3. Explore IP addressing, subnetting, routing algorithms and their application.
4. Get a brief idea about network analysis tools (Wireshark, NMAP).

Course Outcomes

The outcomes of this course are as follows:

CO1: Evaluate network devices functionality and network command significance.

CO2: Implement error control algorithm.

CO3: Analyze and implement routing algorithms.

CO4: Implement and evaluate various network topologies.

CO5: Familiarize with network simulator and network traffic analysis tools.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	2	-	-	-	-	-	-	-	2	-	-
CO 2	-	-	-	-	2	-	-	-	-	-	-	-	2	-	-
CO 3	-	-	-	-	2	-	-	-	-	-	-	-	2	-	-
CO 4	-	-	-	-	2	-	-	-	-	-	-	-	2	-	-
CO 5	-	-	-	-	2	-	-	-	-	-	-	-	2	-	-
Average	-	-	-	-	2	-	-	-	-	-	-	-	2	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Familiarization with networking devices. **(CO1)**

Experiment 2: Write a program for bit stuffing and de-stuffing in a bit stream. **(CO2)**

Experiment 3: Write a program for CRC and Hamming Code. **(CO2)**

Experiment 4: Familiarization with Network IP, subnetting and supernetting. **(CO3)**

Experiment 5: Familiarization of basic network command and network configuration commands. **(CO1, CO5)**

Experiment 6: Set up a network topology in Cisco Packet Tracer (Ring, Bus, Star, Mesh etc.) **(CO4, CO5)**

Experiment 7: Set up network topology in two and more than two routers. **(CO4, CO5)**

Experiment 8: Distance vector routing protocol **(CO3)**

Experiment 9: Link-state vector routing protocol **(CO3)**

Experiment 10: Familiarization with network monitoring tools (NMAP and Wireshark) **(CO5)**

Experiment 11: Capture network traffic using Wireshark. **(CO5)**

Experiment 12: Analyzing network traffic using Wireshark. **(CO5)**

Total Lab hours 30

Textbooks

1. James F. Kurose, and Keith W.Ross, "Computer Networking : A Top-Down Approach", 8th Edition, Pearson, 2022.

2. Andrew S. Tanenbaum, "Computer Networks", 5th Edition, Pearson Education, 2023.

Reference Books

1. W. Tomasi, "Introduction to data communications and networking", 5th edition, Prentice-Hall, Inc., 2008.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50	50

Course Code	Course name	L	T	P	C
CSEG2020	**Object Oriented Programming**	3	0	0	3
Total Units to be Covered: 6		**Total Contact Hours: 45**			
Prerequisite(s):	Programming in C	**Syllabus version: 1.0**			

Course Objectives

1. Understand the need for OOPs and develop Java programs with object-oriented features.
2. Learn the concepts of JDBC and develop standalone application with GUI Panel.
3. Design & implement Java applications for real world scenarios.

Course Outcomes

CO1. Understand Object Oriented Programming concepts and architecture of Java.

CO2. Analyze and model the real-world entity using Java programming language.

CO3. Develop packages with Generics and Implement Interfaces with Exception handling.

CO4. Create Stand-alone Java applications using GUI swings and JDBC.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	-	-	-	-	-	-	-	-	-	-	2	3	-
CO 2	-	3	3	-	-	-	-	-	-	-	-	-	2	3	-
CO 3	-	3	3	-	-	-	-	-	-	-	-	-	2	3	-
CO 4	-	-	-	2	-	-	1	-	2	2	-	-	2	3	-
Average	.25	1.5	1.5	.5	-	-	.25	-	.5	1	-	-	2	3	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to OOPs **5 Lecture Hours**

Object Oriented Programming History and Evolution, Object Oriented Programming Principles, Features of Java, Input Output Statements, Comment Line Arguments, Data Types, Variables, Operators, Program Control Statements, Arrays, Type of Arrays, Strings.

Unit II: Classes, Inheritance, Packages and Interfaces **8 Lecture Hours**

Class Fundamentals, Objects, Constructors, Garbage Collection, this Keyword, Java's Access Modifiers, Method Overloading, static Keyword, Inheritance, Types of Inheritance, super to Access Superclass Members, Method Overriding, Abstract Classes, Using final, Packages and Interfaces, Build-in Interface, User defined Interfaces.

Unit III: Nested Classes, Exceptions, Multithreading & IO Streams **8 Lecture Hours**

Nested Classes, Types of Nested Classes, Exception Handling, Exception Handlers, Concurrent Programming, The Thread Class and Runnable Interface, Thread Priorities, Synchronization, Java's I/O Streams, Byte Streams and Character Streams, FileWriter, FileReader.

Unit IV: Generics, Lambdas, GUI Swing & Database Connectivity **8 Lecture Hours**

Generics Fundamentals, Generic Class, Generic Methods, Lambdas, Functional Interfaces, Swing, Components and Containers, Layout Managers, Swing Event Handling, Event Listeners, Event Classes and Listener Interfaces, Swing Controls, Database Connectivity, Statement, Prepared Statement, CallableStatement, Resultset. Persistent Data.

Unit V: Collections and Wrapper Class **6 Lecture Hours**

Collections, Iteration, Collection Interface, Set and SortedSet, List, Map and SortedMap, Wrapped Collections and Collections Class, Wrapper classes and loading classes.

Unit VI: Capstone Project **10 Lecture Hours**

Create Standalone Java Project, Designing of UML and database diagrams, GUI Panel development using swing, Establish connection with Database and Panel. Source Code Management and Collaboration using Git/GitHub. Unit Testing using JUnit, Integration Testing, Build and Artifactory Management.

Total lecture Hours 45

Textbooks

1. Herbert Schildt, "Java: A Beginner's Guide", 9th Edition, McGraw-Hill Education, 2022.

2. Allen B. Downey and Chris Mayeld, "Think Java: How to Think Like a Computer Scientist", 2nd Edition, O'Reilly Media Publishers, 2020.

Reference Books

1. Herbert Schildt, "Java: The Complete Reference", 12th Edition, McGraw Hill Publisher, 2022.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSEG2120	Object Oriented Programming Lab		0	0	2	1
Total Units to be Covered:11		Total Contact Hours: 30				
Prerequisite(s):	Programming in C Lab		Syllabus version: 1.0			

Course Objectives

1. Design and code the programs using java concepts.
2. Utilize the flexibility and modularity provided by OOPs using Java.
3. Implement Exception handling and Multithreading in Java
4. Develop server side applications using design patterns and data base connectivity

Course Outcomes

At the end of this course student should be able to

CO 1. Demonstrate object-oriented concepts using Java Language.

CO 2. Implement programs in Java using packages, interfaces and exceptions.

CO 3. Apply strings, threads and collections in Java.

CO 4. Develop server side applications using JSP, servlet and JDBC

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	2	2	-	2	-	-	-	-	-	-	-	1	3	-
CO 2	1	2	2	2	2	-	-	-	-	-	-	-	1	3	-
CO 3	1	2	2	1	2	-	-	-	-	-	-	-	1	3	-
CO 4	1	2	2	-	2	-	-	-	-	-	-	-	1	3	-
Average	1	2	2	.75	2	-	-	-	-	-	-	-	1	3	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

List of Experiments

Experiment 1	Introduction to Java Environment
Experiment 2	Basic Java Programming
Experiment 3	Basic Java Programming
Experiment 4	Inheritance
Experiment 5	Interface
Experiment 6	Package
Experiment 7	Exceptions
Experiment 8	Strings Handling and Wrapper Class
Experiment 9	Threads and Collections
Experiment 10	JDBC
Experiment 11	Servlets

Total Lab hours 30

Textbooks

1. Ken Arnold, and James Gosling, "The Java Programming Language", 3rd Edition, Pearson, 2018.

2. Khalid Mughal, "A premier guide to SCJP", 3rd Edition, Pearson.

3. Bruce Ackel, "Thinking in Java", 3rd Edition, Pearson.

4. Video resources http://www.youtube.com and blackboard.

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50	50

Course Code	Course name		L	T	P	C
CSEG2064	Software Engineering		3	0	0	3
Total Units to be Covered: 5		Total Contact Hours: 45				
Prerequisite(s):	Basic Knowledge of Programming		Syllabus version: 1.0			

Course Objectives

1. To explore software development methodologies (waterfall, agile, DevOps) and their integration of testing, quality assurance, reliability, and risk management.
2. To comprehend software requirements engineering and develop skills in creating well-structured Software Requirements Specifications (SRS).
3. To acquire understanding of planning a software project, its cost estimation models and to understand the software quality models.

Course Outcomes

CO 1. Understand the fundamental concepts and importance of Software Engineering in modern software development.

CO 2. Learn various software development methodologies, including Agile, Waterfall, and iterative approaches.

CO 3. Explore software design principles and architectural patterns for creating robust and maintainable software systems.

CO 4. Apply project management principles to effectively plan, monitor, and control software projects.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	3	2	2	1	-	-	-	2	-	3	-	1	2	-
CO 2	2	3	2	2	1	-	-	-	2	-	3	-	1	2	-
CO 3	2	3	2	3	1	-	-	-	2	-	3	-	1	2	-
CO 4	2	3	2	3	1	-	-	-	2	-	3	-	1	2	-
Average	2	3	2	2.5	1	-	-	-	2	-	3	-	1	2	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

Syllabus

Unit I: Introduction to Software Engineering **7 Lecture Hours**

Definition of Software Engineering, S/W characteristics, applications, Software development life cycle ; Life Cycle Models – Waterfall (classical and iterative), Spiral, Prototyping & RAD Models, Software processes, Process Models – overview Agile Model and Various Agile methodologies - Scrum, XP, Lean, and Kanban. Scope of each model and their comparison in real-world case studies.

Unit II: Requirements Modelling and Design **9 Lecture Hours**

System and software requirements; Requirements Engineering-Crucial steps; types of requirements, Functional and non-functional requirements; Domain requirements; User requirements; Elicitation and analysis of requirements; Requirements documentation – Nature of Software, Software requirements specification, Use case diagrams with guidelines, DFD, SRS Structure, SRS Case study, Design concepts and principles - Abstraction - Refinement - Modularity Cohesion coupling, Architectural design, Detailed Design Transaction Transformation, Refactoring of designs, Object-oriented Design User-Interface Design.

Unit III: Software Reliability **9 Lecture Hours**

Introduction to Software Reliability; Hardware reliability vs. Software reliability; Reliability metrics; Failure and Faults – Prevention, Removal, Tolerance, Forecast; Dependability Concept – Failure Behavior, Characteristics, Maintenance Policy; Reliability and Availability Modeling; Reliability Evaluation Testing methods, Limits, Starvation, Coverage, Filtering; Microscopic Model of Software Risk; Classes of software reliability Models; Statistical reliability models; Reliability growth models; Defining and interpreting reliability metrics; Fault Detection and Prevention; Techniques for detecting and mitigating software faults; Static analysis tools and techniques; Dynamic analysis methods; Software Fault Tolerance; Software Maintenance and Reliability; Reliability Assessment and Evaluation; Methods for assessing and quantifying software reliability; Case Studies and Real-world Applications.

Unit IV: Software Testing, metrics and Quality Assurance 10 Lecture Hours

Testing types and techniques such as black box, white box, and gray box testing, functional and structural testing; Test-driven development, code coverage, and quality metrics; Testing process, design of Test cases, testing techniques - boundary value analysis - equivalence class testing - decision table testing, cause-effect graphing, path testing, data flow testing, and mutation testing. Unit, integration, system, alpha, and beta testing, debugging techniques; verification and validation techniques, levels of testing, regression testing, quality management activities, product and process quality standards (ISO9000, CMM), metrics understanding (process, product, project metrics), size metrics (LOC, Function Count, Albrecht FPA), product metrics, metrics for software maintenance, cost estimation techniques (static, single variable, multivariable models), cost-benefit evaluation techniques, Testing tools and standards such as Jira and Selenium, test automation frameworks and tools (Selenium, Appium, JUnit), performance testing and load testing, and defect management and root cause analysis.

Unit V: Software Quality and Risk Management 10 Lecture Hours

McCall quality factors, ISO and CMM Model, Tools and Techniques for Quality Control, Pareto Analysis, Statistical Sampling, Quality Control Charts and the seven Run Rule. Modern Quality Management, Risk Management – importance, types, process and phases, qualitative and quantitative risk analysis, Risk Analysis and Assessment, Risk Strategies, Risk Monitoring and Control, Risk Response and Evaluation. Software Reliability: Reliability Metrics, Reliability Growth Modeling. Use Case: Defect Tracking and Management. Test Automation Tools: Jira, Selenium:, Appium; JUnit.

Total lecture Hours 45

Textbooks

1. Roger S. Pressman, "Software Engineering: A practitioner's approach", 7th Edition, McGraw Hill, 2009.

2. Pankaj Jalote, "An integrated approach to Software Engineering", 3rd Edition, Springer/Narosa, 2005.

Reference Books

1. James F. Peters, and Witold Pedrycz, "Software Engineering: an Engineering approach", John Wiley, 2007.

2. Waman S Jawadekar, "Software Engineering principles and practice", McGraw Hill, 2004.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
	Exploratory-2		3	0	0	3
Total Units to be Covered:		Total Contact Hours:				
Prerequisite(s):			Syllabus version: 1.0			

Exploratory courses offered by different schools , student has a choice to opt desired course from the available tracks .

Course Code	Course name	L	T	P	C
MATH2059	Linear Algebra	3	0	0	3
Total Units to be Covered: 6	Total Contact Hours: 45				
Prerequisite(s):	Discrete Mathematics, Advanced Engineering Mathematics 1 & 2	Syllabus version: 1.0			

Course Objectives

The course aims to

1. Provide students with understanding of fundamental concepts of linear algebra and their applications.

2. Develop mathematical models employing Linear Algebra framework for problems arising in a variety of disciplines.

3. Empower the students to learn and formulate problems using linear Algebra in science, engineering including emerging areas like data analytics and deep learning.

Course Outcomes

On completion of this course, the students will be able to

CO1. Model situations in diverse contexts involving vectors, matrices, and systems of linear equations.

CO2. Demonstrate an interplay between the core mathematical concepts and applications in computer and allied sciences.

CO3. Comprehend and visualize concepts of eigenvalues and eigenvectors in computer graphics and emerging applications.

CO4. Apply SVD, a powerful technique, which is crucial for various computational tasks, including machine learning, computer graphics, data compression, and image processing.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3

CO 1	2	2	2	3	-	-	-	-	2	-	-	3	-	-	-
CO 2	1	2	3	2	-	-	-	-		-	-	3	2	-	-
CO 3	2	3	3	3	2	2	-	-	2	-	-	3	3	2	-
CO 4	3	3	3	3	2	-	-	2	2	-	2	3	2	2	-
Average	2	2.5	2.75	2.75	1	.5	-	.5	1.5	-	.5	3	1.75	1	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

Syllabus

Unit I: Introduction to Linear Algebra **1 Lecture Hours**

Linearity vs Non-linearity (Real life examples), Panoramic view of linear algebra(Linear + algebra), Mention of emerging Applications of Computer Science: Google PageRank, Google Maps, etc.

Unit II: System of Linear Equations **8 Lecture Hours**

Vectors and linear combination, Visualization of the system of linear equations, Elimination using matrices, Rank of a matrix, Echelon forms, Normal form, Solution of a homogeneous and non-homogeneous system of equations, Applications in emerging areas: Machine learning models, Cryptography and Color models, Recap of Unit-II.

Unit III: Vector Spaces and Linear Transformation **12 Lecture Hours**

Vector spaces and subspaces, Linear span, Linear independence/dependence of vectors, Basis and dimension, Linear mapping, Matrix as a linear mapping, Kernel and image of linear mapping, Null space, Rank nullity theorem, Singular and non-singular mappings, Isomorphisms, Operations with linear mappings, Similarity of matrices, Change of basis, Inner product spaces, Vector and matrix norms, Orthogonality, Orthogonal sets and bases, Projections, Gram-Schmidt orthogonalization process, Applications of linear transformation: Data Smoothing, Image scaling, Recap of Unit III.

Unit IV: Eigenvalues and Eigenvectors **8 Lecture Hours**

Polynomial of matrices, Characteristic polynomial, Cayley-Hamilton theorem, Eigenvalues and eigenvectors, Geometric interpretation of eigenvectors, Diagonalization, Power of a matrix, Function of matrices, Diagonalization of symmetric matrices, Quadratic forms, Methods for computing Eigenvalues, Method of Least squares, Application of Eigensystems: Facial and Ear recognition, Feature extraction, Internet search engines. Recap of Unit IV.

Unit V: Singular value decomposition **8 Lecture Hours**

Spectral decomposition, Singular value decomposition (SVD), Best rank k approximations, Power method for computing the Singular value decomposition, Applications of Singular value decomposition: Principal component analysis, Singular vectors, Centering data, Ranking documents and Web pages, Clustering, Recap of Unit V.

Unit VI: Applications of Linear Algebra: **8 Lecture Hours**

Computational Approach

Applications of linear systems: Design of traffic patterns/Circuit with one closed loop/Balancing chemical equations; Data compression using SVD, Word Embeddings and Exploring Biases in Data, Markov Matrices and Applications to PageRank, Game of strategy, Vector space models for information retrievals, Vector matrix: Moves on a chessboard, Distribution of genotypes in a population.

Total lecture Hours 45

Textbooks

1. G. Strang, "Linear Algebra and its Applications", 4th Edition, Cengage Learning, 2005.

2. G. Williams, "Linear Algebra with Applications", 8th Edition, Jones and Bartlett Learnings, 2012.

3. H. Anton, and C. Rorres, "Elementary linear algebra with supplemental applications", 11th Edition, Wiley, 2016.

Reference Books

1. I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning", The MIT Press, 2016.

2. K. Singh, "Linear Algebra Step by Step", Oxford University Press, 2013.

3. D. C. Lay, Steven R. Lay, and Judi J. McDonald, "Linear Algebra and its Applications", 5th Edition, Pearson Education India, 2023.

4. H. Wendland, "Numerical Linear Algebra An Introduction", Cambridge University Press, 2018.

5. W. Ford, "Numerical linear algebra with applications using MATLAB", Academic Press, 2014. [Chapter 7, 15, 17]

6. J. MacCormick, "Nine Algorithms that Changed the Future", Princeton University Press, 2021.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
SLLS2004	Indian Constitution	0	0	0	0
Total Units to be Covered: 04	Total Contact Hours:				
Prerequisite(s):		Syllabus version: 1.0			

Course Objectives

The course aims to

1. To realise the significance of constitution of India to students from all walks of life and help them to understand the basic concepts of Indian constitution.
2. To identify the importance of fundamental rights as well as fundamental duties.
3. To understand the functioning of Union, State and Local Governments in Indian federal system.
4. To learn procedure and effects of emergency, composition and activities of election commission and amendment procedure.

Course Outcomes

CO1. Understand and explain the significance of Indian Constitution as the fundamental law of the land.

CO2. Exercise his fundamental rights in proper sense at the same time identifies his responsibilities in national building.

CO3. Analyse the Indian political system, the powers and functions of the Union, State and Local Governments in detail

CO4. Understand Electoral Process, Emergency provisions and Amendment procedure.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	-	2	-	1	-	-	-	-	-	-	-
CO 2	-	-	-	-	-	2	-	1	-	-	-	-	-	-	-
CO 3	-	-	-	-	-	2	-	1	-	-	-	-	-	-	-
CO 4	-	-	-	-	-	2	-	1	-	-	-	-	-	-	-

Average	-	-	-	-	-	2	-	1	-	-	-	-	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to Constitution:

Meaning and importance of the Constitution, salient features of Indian Constitution. Preamble of the Constitution**.** Fundamental rights**-** meaning and limitations. Directive principles of state policy and Fundamental duties -their enforcement and their relevance.

Unit II: Union Government:

Union Executive- President, Vice-president, Prime Minister, Council of Ministers. Union Legislature- Parliament and Parliamentary proceedings. Union Judiciary- Supreme Court of India – composition and powers and functions.

Unit III: State and Local Governments:

State Executive- Governor, Chief Minister, Council of Ministers. State Legislature- State Legislative Assembly and State Legislative Council. State Judiciary-High court. Local Government-Panchayat raj system with special reference to 73rd and Urban Local Self Govt. with special reference to74th Amendment.

Unit IV: Election provisions, Emergency provisions, Amendment of the constitution

Election Commission of India-composition, powers and functions and electoral process. Types of emergency-grounds, procedure, duration and effects. Amendment of the constitution- meaning, procedure and limitations.

Textbooks

1. Ethics and Politics of the Indian Constitution Rajeev Bhargava Oxford University Press, New Delhi, 2008
2. The Constitution of India B.L. Fadia Sahitya Bhawan; New edition (2017)
3. Introduction to the Constitution of India DD Basu Lexis Nexis; Twenty-Third 2018 edition

Reference Books

1. Introduction to Indian Knowledge System: Concepts and Applications, authored by Dr. B. Mahadevan, Professor of IIM Bangalore and founding Vice Chancellor of Chinmaya Vishwa Vidyapeeth; Dr. Vinayak Rajat Bhat, Associate Professor, Chanakya University, Bengaluru; and Dr. Nagendra Pavana R.N., Faculty at the School of Vedic Knowledge Systems, Chinmaya Vishwa Vidyapeeth; Forewords by Dr. Anil Sahasrabudhe, Former Chairman AICTE; Prof. Subhash Kak, Oklahoma State University, USA; and Dr. S. Sadagopan, Chairman, BoG, IIITDM – Kancheepuram & Founder Director, IIITBangalore

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme continuous assement **/**NPTEL

Course Code	Course name	L	T	P	C
EMPL002	EDGE-SoftSkills	1	0	0	0
Total Units to be Covered:		Total Contact Hours:			
Prerequisite(s):		Syllabus version: 1.0			

Employment training related course.

SEMESTER V

Course Code	Course name		L	T	P	C
CSEG3040	Cryptography and Network Security		3	0	0	3
Total Units to be Covered: 5		Total Contact Hours: 45				
Prerequisite(s):	Data Communication and Networks		Syllabus version: 1.0			

Course Objectives

1. Understand the fundamental concepts of cryptography and its applications.
2. Learn about various encryption and decryption techniques.
3. Explore network security protocols and mechanisms.
4. Develop the skills to analyze, design, and implement secure communication systems.

Course Outcomes

On completion of this course, the students will be able to

CO1. Relate the historical development of cryptography, its basic concepts, and terminology, and identify the different types of attacks and security services in a networked environment.

CO2. Illustrate the principles of symmetric-key cryptography and Public Key Cryptography, including the working of algorithms like DES, AES, and RSA and evaluate the security of block cipher modes of operation and its applications.

CO3. Use cryptographic hash functions, such as MD5, SHA-1, and SHA-256, for ensuring data integrity, and design applications that implement HMAC for message authentication.

CO4. Demonstrate an understanding of network security principles, including authentication, access control, and various network attacks, and propose countermeasures to mitigate those attacks.

CO5. Design and implement secure communication systems using cryptographic techniques and network security principles, and critically analyze the effectiveness of these systems in real-world scenarios.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	2	1	-	-	-	-	-	-	-	-	-	2	-	-
CO 2	2	2	3	-	-	-	-	-	-	-	-	-	2	-	-
CO 3	2	2	3	-	-	-	-	-	-	-	-	-	2	-	-
CO 4	2	2	1	2	-	-	-	-	-	-	-	-	2	-	-
CO 5	2	2	1	2	-	-	-	-	-	-	-	-	2	-	-
Average	2	2	1.8	0.8	-	-	-	-	-	-	-	-	2	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to Cryptography **5 Lecture Hours**

Historical development of cryptography, Basic concepts and terminology, Types of attacks and security services, Classical cryptography and its limitations, stream and block ciphers, cryptanalysis, stereography.

Unit II: Symmetric-Key Cryptography **10 Lecture Hours**

Substitution and transposition ciphers, Shannon's theory of confusion and diffusion, fiestal structure, Data Encryption Standard (DES), Triple DES, Advanced Encryption Standard (AES), Block cipher modes of operation, traffic confidentiality, key distribution, random number generation.

Unit III: Introduction to Number Theory and Public-Key Cryptography

10 Lecture Hours

Introduction to graph, ring and field, prime and relative prime numbers, modular arithmetic, Fermat's and Euler's theorem, primality testing, Euclid's Algorithm, Introduction to public-key cryptography, RSA algorithm, Diffie-Hellman key exchange, introductory idea of Elliptic curve cryptography, Digital signatures, and certificates

Unit IV: Cryptographic Hash Functions **10 Lecture Hours**

Properties and applications of hash functions, Message Digest algorithms (MD5, SHA-1, SHA-256), HMAC (Hash-based Message Authentication Code), Kerberos and X.509, directory authentication service.

Unit V: Network Security **10 Lecture Hours**

Secure communication principles, Authentication and access control, Network attacks and countermeasures, Virtual Private Networks (VPNs), Transport Layer Security (TLS) and Secure Sockets Layer (SSL), Wireless Network Security, Security in Mobile and Cloud Computing

Total lecture Hours 45

Textbooks

1. William Stallings, "Cryptography and Network Security: Principles and Practice", 7th Edition, Pearson, 2017.

2. Christof Paar, and Jan Pelzl, "Understanding Cryptography: A Textbook for Students and Practitioners", Springer, 2014.

Reference Books

1. Wade Trappe, and Lawrence C. Washington, "Introduction to Cryptography with Coding Theory", 2nd Edition, Pearson, 2005.

2. William Stallings, "Network Security Essentials: Applications and Standards", 6th Edition, Pearson, 2018.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSEG3055	**Formal Languages and Automata Theory**	3	0	0	3
Total Units to be Covered: 4	**Total Contact Hours: 45**				
Prerequisite(s):	Discrete Mathematics and Design and Analysis of Algorithms	**Syllabus version: 1.0**			

Course Objectives

1. Understand the foundations and principles of formal languages used in computing.
2. Formulate formal and rigorous mathematical arguments.
3. Prove or disprove theorems in automata theory.
4. Develop a formal computational model pertaining to real-world problems.

Course Outcomes

On completion of this course, the students will be able to,

CO1. Recognize the various language classes and their relationships.
CO2. Comprehend regular grammar and expressions using finite automata.
CO3. Develop grammar and recognizer for various formal languages.
CO4. Analyze the decidability and intractability of different computational problems.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	3	2	2	-	-	-	-	-	-	-	2	2	-	-
CO 2	3	3	2	2	-	-	-	-	-	-	-	2	3	-	-
CO 3	2	3	2	2	-	-	-	-	-	-	-	2	3	-	-
CO 4	2	3	2	2	-	-	-	-	-	-	-	2	2	-	-
Average	2.25	3	2	2	-	-	-	-	-	-	-	2	2.5	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Finite Automata & Regular Languages **15 Lecture Hours**

Computational problems vs. Formal Languages. Finite State Automata- Deterministic and Non-Deterministic, Regular Expression and Language, Closure property of Regular Language, Limitations of Regular Language, Pumping Lemma, Minimization Algorithm, Myhill-Nerode relations, Finite Automata with outputs- Mealy and Moore Machines.

Unit II: Grammar's Classification and Push Down Automata **15 Lecture Hours**

Grammars and Chomsky Classification, Regular Grammar, Context Free Grammar and Languages: Ambiguity, Simplification of CFGs, Normal Forms for CFGs, Pumping Lemma for CFLs, Applications to Parsing, Pushdown Automata (PDA), PDA vs CFLs, Deterministic CFLs, Linearly Bounded Automata (LBA).

Unit III: Turing Machine **10 Lecture Hours**

Introduction to Turing Machines, Configurations, Multi-tape Turing machines, Halting vs Looping, Recursive and Recursively Enumerable Languages, Decidable and Undecidable Languages.

Unit IV: Decidability & Intractability **5 Lecture Hours**

Undecidability of Halting Problem, Introduction to the Theory of NP-completeness, Reductions, Rice theorem, Post Correspondence Problem, Church-Turing Thesis, Cook-Levin Theorem.

Total lecture Hours 45

Textbooks

1. John E. Hopcroft, Rajiv Motwani, and Jeffrey D. Ullman, "Automata Theory, Languages, and Computation", 3rd Edition, Pearson, 2008.

2. Dexter C. Kozen, "Automata and Computability", Springer Publishers, 2007.

Reference Books

1. Peter Linz, "An Introduction to Formal Languages and Automata", 6th edition, Jones & Bartlett India, 2016.

2. Manish Kr. Jha, "Automata Theory", 2nd Edition, S.Chand Publication, 2015.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSEG3002	**Object Oriented Analysis and Design**	3	0	0	3
Total Units to be Covered: 5	**Total Contact Hours: 45**				
Prerequisite(s):	Software Engineering	**Syllabus version: 1.0**			

Course Objectives

1. Learn object-oriented analysis and design and UML diagrams.
2. Apply the UML to solve several common modelling problems.
3. Learn to apply object-oriented concepts to all stages of the software development.

Course Outcomes

On completion of this course, the students will be able to

CO1. Describe OOAD Principles including encapsulation, inheritance, and polymorphism, and comprehend their relevance in software development.

CO2. Illustrate UML Modeling and the role of actors, classes, and objects in a software design

CO3. Illustrate activity, interaction, event, signals to design state diagram.

CO4. Modeling software systems using OOAD methodologies to tackle complex real-world challenges.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	2	3	1	-	-	-	-	-	-	-	-	-	2	-
CO 2	3	2	3	1	-	-	-	-	-	-	-	-	-	2	-
CO 3	3	3	2	1	-	-	-	-	-	-	-	-	-	2	-
CO 4	3	2	2	1	-	-	-	-	-	-	-	-	-	2	-
Average	3	2.25	2.5	1	-	-	-	-	-	-	-	-	-	2	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: OOAD Basics **10 Lecture Hours**

Overview of object oriented system development, Basic notion of objects, Object-oriented concepts: objects, classes, encapsulation, inheritance, polymorphism, Benefits of object-oriented analysis and design, Object basics –The unified process, Multiple Views of Objects, Modelling concepts, Modeling as a design technique, Analysis and modelling, UML and its role in software development, UML diagrams - Class diagrams, use case diagrams, Class, State, Interaction Modelling sequence diagrams, Modeling relationships between classes and objects, UML Rational Unified Process(RUP) Contrasting with Procedural Computation, Client-Server/Message Passing.

Unit II: Basic & Advanced Structural Modeling **10 Lecture Hours**

Requirement Modeling: Requirement Engineering, Requirement Modeling: Use Cases Diagrams, Basic structural Modeling: Class diagrams, their components, Modeling Relationships, Common Mechanisms, Diagrams, abstract classes, Interfaces, Types and Roles, Packages. Class & object Diagrams: Terms, concepts, Modeling techniques for class & object diagrams Identifying operations and Specifying operations using CRC card.

Advanced Structural Modeling Concepts: design patterns, architectural modeling, and framework integration, Exploring advanced UML concepts like constraints, derived properties, and visibility control. Overview of C++: Procedural Extension of C, Objects, Classes and Encapsulation, Overloading, Inheritance & Polymorphism, Type Casting.

Unit III: Basic Behavioral Modeling **10 Lecture Hours**

Basic Behavioral Modeling-I: Use Case Modeling, Use Case Diagrams using UML notation, use case diagram extensions and variations, Use Case Descriptions, Activity Diagrams, State Diagrams, Swim lanes Design Architecture

Basic Behavioral Modeling-II: Interaction, Interaction Diagrams, Behavioral Patterns, Scenario-Based Analysis, Verification and Validation of Behavior Models

Unit IV: Advanced Behavioral Modeling **10 Lecture Hours**

Advanced Behavioral Modeling: Advanced Use Case Modeling , Activity Diagrams, State Machine Diagrams, Interaction Diagrams: Sequence and Communication Diagrams, Event-Driven Architectures, State Machines, Processes and Threads, Time and Space, State chart diagrams & Sequential Diagrams, Business Process Modeling, Modeling Reactive Systems, Advanced Behavioral Patterns, Case Studies and Practical Applications: Analyzing and modeling real-world systems using advanced behavioral modeling techniques, Case studies on event-driven systems, business processes, and reactive systems, Integration of advanced behavioral models with software development practices

Unit V: Advanced Architectural Modeling **5 Lecture Hours**

Component Diagrams, Deployment Diagrams, Architectural Styles (client-server, microservices, event-driven), Architectural patterns (e.g., MVC, MVVM, layered architecture), Architectural Decision-Making: Enterprise Architecture, Distributed Systems Architecture - Case studies of distributed system architectures (e.g., cloud computing, IoT platforms), Domain-Driven Design (DDD), Service-Oriented Architecture (SOA, Event-Driven Architecture (EDA), Cloud and Serverless Architectures, Security and Privacy in Architectural Modeling, Emerging Trends and Technologies in Architecture: containerization, microservices, and serverless computing, Case studies and discussions on the practical application of new architectural approaches

Total lecture Hours 45

Textbooks

1. Michael Blaha and James Rumbaugh, “Object-oriented modeling and design with UML”, 2nd Edition, Prentice-Hall of India, 2007.

2. Grady Booch, James Rumbaugh, and Ivar Jacobson, "The Unified Modeling Language User Guide", 2nd Edition, Pearson Education, 2012.

3. Russ Miles, and Kim Hamilton, "Learning UML 2.0: A Pragmatic Introduction to UML", O'Reilly, 2006.

Reference Books

1. Meilir Page-Jones, "Fundamentals of Object-Oriented Design in UML", Pearson Education, 2002.

2. Pascal Roques, "Modeling Software Systems Using UML2", WILEY- Dreamtech India Pvt. Ltd, 2004.

3. Atul Kahate, "Object Oriented Analysis & Design", The McGraw-Hill Companies, 2004.

4. John W. Satzinger, Robert B. Jackson, and Stephen D Burd, "Object-Oriented Analysis and Design with the Unified Process", Cengage Learning, 2007.

5. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado, "UML 2 Toolkit", WILEY-Dreamtech India Pvt. Ltd, 2003.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
	Exploratory-3		3	0	0	3
Total Units to be Covered:		**Total Contact Hours:**				
Prerequisite(s):			**Syllabus version: 1.0**			

Exploratory courses offered by different schools , student has a choice to opt desired course from the available tracks .

Course Code	Course name		L	T	P	C
SLSG0205	Start your Startup		2	0	0	2
Total Units to be Covered: 10		Total Contact Hours: 30				
Prerequisite(s):		Syllabus version: 1.0				

Course Objectives

1. Foster an entrepreneurial mindset, including creativity, adaptability, and resilience, enabling students to identify and evaluate potential business opportunities!
2. Develop skills in conducting market research and competitor analysis.
3. Equip students with the tools to create comprehensive business plans and foster an understanding of the importance of effective team building and management.
4. Provide insights into funding options and strategies for securing investment.
5. Understand the challenges of scaling operations and developing sustainable growth strategies.

Course Outcomes

On completion of this course, the students will be able to:

CO1. Foster, identify, assess, and demonstrate an entrepreneurial mindset, including creativity, adaptability, and resilience, and enable students to identify, evaluate potential business opportunities for viability and market fit.

CO2. Develop skills in conducting thorough market research and competitor analysis to gather relevant data and insights.

CO3. Use market research findings to make informed business decisions and create a well-structured and comprehensive business plan.

CO4. Equip students with the tools to create comprehensive business plans and foster an understanding of the importance of effective team building and management.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	3	2	-	-	-	-	-	-	-	3	-	-	3	3
CO 2	3	3	-	-	-	-	-	-	-	-	-	-	-	3	3
CO 3	3	3	3	3	-	-	-	-	-	-	-	-	3	3	3
CO 4	3	3	3	3	-	-	-	-	-	-	-	-	3	3	3
Average	3	3	2	1.5	-	-	-	-	-	-	.75	-	1.5	3	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: **3 Lecture Hours**

Introduction to Startup world, Understanding the basics, Successful Entrepreneur's journey (can be done with alumni /seniors from Runway cohort)

Definition and characteristics of entrepreneur, motivation for starting a new startup, Exploring entrepreneurial ecosystem, entrepreneurial mindset and skills

Unit II: **3 Lecture Hours**

SWOT Analysis for an Entrepreneur, challenges and overcoming challenges in Entrepreneurship.

Unit III: **3 Lecture Hours**

Startup Idea Scouting Techniques and Exercises, Successful startups and validation process.

Unit IV: **3 Lecture Hours**

Your target audience and validation process Part 1, Your target audience and validation process Lean Startup Methodology and MVP Development, product development and Prototyping.

Unit V: **3 Lecture Hours**

Introduction to Business Model Canvas Framework, identifying key elements of startup business model Value Proposition Canvas, How to create your VPC.

Unit VI: **3 Lecture Hours**

Legal considerations for startups, Intellectual Property Protection and Contracts
Sales and marketing strategies for startups Digital marketing and customer acquisition

Unit VII: **3 Lecture Hours**

Managing growth and avoiding common pitfalls, scalability and Growth strategies
Importance of storytelling, Storytelling.

Unit VIII: **3 Lecture Hours**

Creating a compelling pitch deck, investor's perspectives on pitching to potential investors.

Unit IX: **3 Lecture Hours**

Social Impact and Sustainability in Entrepreneurship, entrepreneurship in emerging technologies and opportunities.

Unit X: **3 Lecture Hours**

Review and recap, key concepts and lessons learnt, final leg Q/A, Evaluation, guidance and hand holding.

Total lecture Hours 30

Textbooks

1. Entrepreneurship: The Practice and Mindset Start Your Own Business.

Reference Books

1. Simon Sinek - The Infinite Game

2. Eric Ries - The Lean Startup
3. Chris Gillebeau - The $100 Startup
4. Simon Sinek - Start With Why
5. Peter Thiel - Zero to One

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Component	QUIZ	Pitch Weightage	Total
Weightage (%)	60	40	100

Course Code	Course name	L	T	P	C
CSEG3060	**Research Methodology in CS**	3	0	0	3
Total Units to be Covered: 06		**Total Contact Hours: 45**			
Prerequisite(s):		**Syllabus version: 2.0**			

Course Objectives

1. To Familiarize Students with Research Problem Identification
2. To Introduce Various Research Methodologies in Computing Science
3. To Promote Ethical Conduct in Research
4. To Enhance Technical Writing and Proposal Development Skills
5. To Provide Insights into Intellectual Property and Patents

Course Outcomes

On completion of this course, the students will be able to

CO1. Identify and formulate research problems using defined criteria and characteristics.

CO2. Utilize various investigative approaches and methodologies for problem-solving in computing science research.

CO3. Demonstrate ethical conduct and awareness of research ethics and plagiarism in their work.

CO4. Produce technically sound and well-structured research reports and proposals.

CO5. Analyze and apply knowledge of intellectual property rights, particularly patents, in the context of computing science.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	2	2	3	-	-	-	-	-	-	-	2	-	-	-
CO 2	-	2	2	3	-	-	-	-	-	-	-	2	-	-	-
CO 3	-	2	2		-	-	-	-	-	-	-	2	-	-	-
CO 4	-	-	2	-	-	-	-	3	-	2	2	2	-	-	-
CO 5	-	-	2	-	-	-	-	2	-	2	2	2	-	-	-

Average	-	1.2	2	1.2	-	-	-	1	-	.8	.8	2	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to Research Problem **9 Lecture Hours**

Understanding Research Problem: Meaning and Significance, Sources and Identification

Criteria and Characteristics of a Good Research Problem: Essential Attributes, Errors in Selection ***Scope, Objectives, and Approaches:*** Defining Scope and Objectives, Investigative Approaches for Solutions, Data Collection, Analysis, Interpretation, Necessary Instrumentation.

Unit II: Research Methods in Computing Science **13 Lecture Hours**

Dialectic of Research in Computing Science: Models of Argument, Proof Methods (Demonstration, Empirical, Mathematical)

Deduction and Induction for Computer Science: Theoretical Models and Approaches, Algorithmic and Software Engineering Approaches, Mathematical Modelling, Performance Estimation and Evaluation

Unit III: Ethical Conduct and Literature Studies **8 Lecture Hours**

Effective Literature Studies: Approaches and Analysis, Addressing Plagiarism, ***Research Ethics:*** Ethical Considerations in Research.

Unit IV: Technical Writing and Research Proposal Development **5Lecture Hours**

Effective Technical Writing, Report Writing Techniques, Developing a Research Proposal, Proposal Format and Components, Presentation and Review Committee Assessment.

Unit V: Intellectual Property and Patents **5 Lecture Hours**

Nature and Process of Intellectual Property: Overview of Patents, Designs, Copyrights, Patenting Process: Research, Innovation, Development

International Scenario and Patenting: International Cooperation on Intellectual Property

Procedures for Patent Grants and Patenting under PCT

Unit VI: Patent Rights and New Developments in IPR **5 Lecture Hours**

Scope and Transfer of Patent Rights: Licensing and Technology Transfer***,*** Patent Information and Databases

New Developments in Intellectual Property Rights: Administration of Patent System***,*** IPR in Computer Software, etc.***,*** Case Studies and IPR in Educational Institutes.

Total lecture Hours 45

Textbooks

1. "Research methodology: an introduction for science & engineering students'", Stuart Melville and Wayne Goddard
2. Research methodology for engineers by R Ganesan, MJP Publishers

Reference Books

1. Algorithm Design Manual by Steven Skiena, Springer publication.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total

Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSEG3056	Probability, Entropy, and MC Simulation		3	0	0	3
Total Units to be Covered: 6		Total Contact Hours: 45				
Prerequisite(s):	Advanced Engineering Mathematics – 1 & 2, Discrete Mathematics, Linear Algebra		Syllabus version: 1.0			

Course Objectives

This course envisions to impart into the students to:

1. Develop a solid foundation in probability, enabling them to recognize probability distributions in real-world scenarios and effectively address probabilistic problems.
2. Equipping with the necessary skills to proficiently utilize Monte Carlo algorithms in tackling intricate problems that involve uncertainty and randomness.
3. Enhances students' analytical and computational abilities to tackle practical applications within the realm of computer science.
4. Possess the requisite knowledge and tools to confidently approach real-world problems that demand expertise in probabilistic analysis, information processing, and simulation techniques.

Course Outcomes

On completion of this course, the students will be able to

CO1. Identify appropriate probability distributions, discrete / continuous, in problems with uncertainty and compute quantities of interest.

CO2. Analyze multivariate data and draw statistical information by identifying the appropriate random variate.

CO3. Gain understanding of probabilistic behavior of complex uncertain systems through Monte-Carlo simulation experiments.

CO4. Calculate uncertainty in a random experiment using entropy function and learn to compute the distance between two probability distributions using KL divergence.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	2	1	-	-	-	-	-	-	-	-	1	2	1	-
CO 2	2	2	1	-	-	-	-	-	-	-	-	1	2	1	-
CO 3	2	2	1	2	1	1	-	-	-	-	-	1	2	1	-
CO 4	2	2	1	2	1	1	-	-	-	-	-	1	2	1	-
Average	2	2	1	1	.5	.5	-	-	-	-	-	1	2	1	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

Syllabus

Unit 0: Motivation **1 Lecture Hours**

Randomness/Uncertainty all pervasive, Probability theory gives rational description while entropy provides quantitative measure of uncertainty, Generation of random numbers for conducting numerical experiments for simulations.

Unit I: Probability **8 Lecture Hours**

Motivation: Frequency based probability, Probability models, Sample space, Algebra of events, Notion of statistical regularity, Simpson’s Paradox, Axioms of probability, Some probability propositions, Conditional probability, Independence of events, Birthday problem, How to make a fair coin from a biased coin?, Reliability: Series and parallel systems, Theorem of total probability, Bayes’ theorem, Debate: Frequentist vs. Bayesian, Bernoulli trials, Geometric probability – continuous sample space. Recap of Unit I.

Unit II: Discrete Random Variables **8 Lecture Hours**

Definition of random variables, Distribution function, Discrete RV, Probability mass function, Expectation, Mean and variance, Moment generating function. Special discrete distributions: Bernoulli and Binomial, Poisson, Geometric and memoryless property of Geometric pmf, Negative binomial, Hypergeometric, Uniform. Indicator random variable. Recursive formulation for binomial probabilities. Recap of Unit II.

Unit III: Continuous Random Variables **9 Lecture Hours**

Definition, Distribution function, Probability density function, Expectation, Mean and variance, Moment generating function. Special continuous distributions: Uniform, Exponential, Gamma, Normal, Pareto. Weibull distribution, Reliability, and failure rate. Functions of random variable. Markov inequality and Chebyshev's inequality. Limiting distributions and Stirling's approximation. Recap of Unit III.

Unit IV: Jointly Distributed Random Variables **7 Lecture Hours**

Random vectors, Joint probability distribution, Independent random variables, Sum of independent random variable, Conditional distribution, Conditional expectation, Expected number of comparisons in Quick sort, Covariance, Correlation coefficient, Multivariate normal distribution. Limit theorems: Law of large numbers and Central limit theorem, Breakdown of CLT. Recap of Unit IV.

Unit V: Monte-Carlo (MC) Simulation **7 Lecture Hours**

MC methods – sampling and simulation, Analogy between probability and volume, Estimation of π, Buffon's needle problem, Estimation of integrals, Linear congruential methods, Pseudo random numbers, Random variable generation - Inverse transform method, Acceptance rejection method. Generating continuous random variables: Exponential, Normal - Box-Muller approach, Gamma distribution. Generating discrete random variables: Binomial and Poisson variables. Illustration of CLT using random numbers. MC simulation of some probability models. Recap of Unit V.

Unit VI: Information Theory and its Applications **5 Lecture Hours**

Overview of information theory, Information / Surprise and Entropy, Entropy as a measure of randomness, Properties of Entropy function, Applications of Entropy: English language, music, logical problems, lossless compression. Kullback-Leibler measure of divergence. Recap of Unit VI.

Total lecture Hours 45

Textbooks

1. S. M. Ross, "A first course in probability", Pearson, 2020.

2. K. S. Trivedi, "Probability & statistics with reliability, queuing and computer science applications", John Wiley & Sons, 2008. Chapters 1 to 5.

3. R. Y. Rubinstein, and D. P. Kroese, "Simulation and the Monte Carlo method", John Wiley & Sons, 2016.

Reference Books

1. M. Mitzenmacher, and E. Upfal, "Probability and computing: Randomization and probabilistic techniques in algorithms and data analysis", Cambridge University Press, 2017.

2. R. Nelson, "Probability, stochastic processes, and queueing theory: the mathematics of computer performance modeling", Springer Science & Business Media, 2013.

3. I. Goodfellow, Y. Bengio, and A. Courville, "Deep learning", MIT press, 2016. Chapter 3.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
EMPL003	**EDGE – Advance Communication**	1	0	0	0
Total Units to be Covered:	**Total Contact Hours:**				
Prerequisite(s):		**Syllabus version: 1.0**			

Employment training related course

SEMESTER VI

Course Code	Course name	L	T	P	C
	Exploratory-4	3	0	0	3
Total Units to be Covered:	**Total Contact Hours:**				
Prerequisite(s):		**Syllabus version: 1.0**			

Exploratory courses offered by different schools , student has a choice to opt desired course from the available tracks .

Course Code	Course name		L	T	P	C
SLLS0103	Leadership and Teamwork		2	0	0	2
Total Units to be Covered: 6		Total Contact Hours: 30				
Prerequisite(s):			Syllabus version: 1.0			

Course Objectives

1. Formulate and articulate a personal point of view about the meaning of leadership and teamwork, and why they are important.
2. Explore and appreciate the scope of leadership and teamwork in one's day-to-day life.
3. Understand the concepts of effective leadership and teamwork in organizations.
4. Identify and assess the skills and motivations associated with effective leadership and teamwork

Course Outcomes

CO1. Understand the basic qualities of a lifelong learner and the process and elements of inquiry-based learning.

CO2. Use and Analyze learning tools needed to be a lifelong learner of diverse subjects and self-driven, goal-oriented learning.

CO3. Demonstrate new learning contexts and develop concepts using the understanding of the learning process and tools.

CO4. Articulate responses for group and individual work undertaken by self and by others, in execution of the project/coursework.

CO5. Practice and Employ learning methods to engage in independent and lifelong learning in the broader context.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	-	1	1	-	3	3	1	1	-	-	-
CO 2	-	-	-	-	-	1	1	-	2	3	1	1	-	-	-
CO 3	-	-	-	-	-	-	1	1	1	3	1	1	-	-	-
CO 4	-	-	-	-	-	1	1	1	2	3	-	1	-	-	-

Average	**-**	**-**	**-**	**-**	**-**	.75	1	.5	2	3	.75	1	**-**	**-**	**-**

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Leadership: Introduction, Self-Awareness & Leadership Examples **5 Lecture hours**

Introduction to the Course, Importance and Its Application in Life, Self-Awareness and Leadership Examples from Different Walks of Life, Personality Assessment through (i) BIG5, (ii) MBTI.

Unit II: Defining Leaders and Leadership **3 Lecture Hours**

Syllabus Booklet: Leadership and Teamwork Defining Leaders and Leadership, Historical Perspective, Contemporary Perspective, Types of Leaders and Leadership Styles.

Unit III: Leadership Toolkit **6 Lecture Hours**

Leadership Tools - Locus of Control, Goal Setting, Time Management, Interpersonal Relationship, Role of Perception, Powerful First Impression, Body Language, Elevator Pitch, Small Talk, Constructive Criticism, Assertiveness Skills, Dealing with Difficult People.

Unit IV: What is a Team **6 Lecture Hours**

What is a Team? Why is a Team needed? 4 Phase Model of Team Formation, What to Do as a Leader and What to Do as a Team Member in Each Phase? Effective Teams and Solving Problems as a Team – Brief Introduction to The Six Thinking Hats

Unit V: Positive Leadership & Team Building Activity **5 Lecture Hours**

Positive Leadership - Communication, Appreciation, Empathy, Feedback, Leaders and Teams: Working Effectively towards Common Goals, Team Building Activity.

Unit VI: Project Work and Submission **5 Lecture Hours**

Total lecture Hours 30

Textbooks

1. Carroll, John, and Sachi Hatakenaka. "Driving Organizational Change in the Midst of Crisis." MIT Sloan Management Review 42, no. 3 (Spring 2001): 70-79.
2. P. M. Senge, The Fifth discipline fieldbook : strategies and tools for building a learning organization. Currency, Doubleday, New York: Currency, Doubleday, 1994.
3. Organizational Behavior, Stephen P. Robbins, Timothi A. Judge and Seema Sanghi, 12th ed, Prentice Hall India.
4. Organizational behavior-Human behavior at work by John W Newstrom, 12th edition,McGrawHill

Reference Books

1. S. R. Covey, The 7 habits of highly effective people: powerful lessons in personal change. New York: Simon & Schuster, 2014.
2. Katie Shonk (2018, June 19). 3 Types of Conflict and How to Address Them. Harvard Law School.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	Project	Classroom Participation	Mega Quiz	Total
Weightage (%)	50	30	20	100

Course Code	Course name	L	T	P	C
CSEG3015	**Compiler Design**	3	0	0	3
Total Units to be Covered: 5	**Total Contact Hours: 45**				
Prerequisite(s):	Automata Theory and Formal Languages Data structures, knowledge of automata theory, basic knowledge of computer architecture	**Syllabus version: 1.0**			

Course Objectives

1. To introduce the major concept areas of language translation and compiler design.
2. To enrich the knowledge in various phases of compiler and its use, code optimization techniques, machine code generation, and use of symbol table.
3. To extend the knowledge of parser by parsing LL parser and LR parser.
4. To provide practical programming skills necessary for constructing a compiler.

Course Outcomes

On completion of this course, the students will be able to

CO 1. Comprehend different phases of compiler.

CO 2. Use concepts of regular grammar to build lexical analyzer.

CO 3. Build parsers for a context free grammar.

CO 4. Synthesize syntax directed translations rules.

CO 5. Assess code and memory optimization techniques to improve the performance of a program.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	2	3	-		-	-	-	-	-	-	-	2	-	-
CO 2	2	2	3	-	1	-	-	-	-	-	-	-	2	-	-
CO 3	2	2	3	-	1	-	-	-	-	-	-	-	2	-	-
CO4	2	2	3	-	1	-	-	-	-	-	-	-	2	-	-
CO5	2	2	3	-	-	-	-	-	-	-	-	-	2	-	-

Average	2	2	3	-	.6	-	-	-	-	-	-	-	2	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction **10 Lecture Hours**

Compiler, Phases and Passes, Bootstrapping, Finite State Machines and Regular Expressions and their Applications to Lexical Analysis, Implementation of Lexical Analyzers, Lexical Analyzer Generator, LEX, Formal Grammars and their Applications to Syntax Analysis, BNF Notation, Ambiguity, YACC. The Syntactic Specification of Programming Languages: Context Free Grammars, Derivation and Parse Tree, Capabilities of CFG.

Unit II: Basic Parsing Techniques **10 Lecture Hours**

Parsers, Shift Reduce Parsing, Operator Precedence Parsing, Top Down Parsing, Predictive Parsing, Automatic Construction of Efficient Parsers: LR Parsers, The Canonical Collection of LR(0) items, Constructing SLR Parsing Tables, Constructing Canonical LR Parsing Tables, Constructing LALR Parsing Tables, Using Ambiguous Grammars, An Automatic Parser Generator, Implementation of LR Parsing Tables, Constructing LALR set of items

Unit III: Syntax-Directed Translation **10 Lecture Hours**

Syntax Directed Translation Schemes, Implementation of Syntax Directed Translators, Intermediate Code, Postfix Notation, Parse Tree & Syntax Tree, Three Address Code, Quadruples & Triples, Translation of Assignment Statements, Boolean Expressions, Statements that alters the Flow of Control, Postfix Translation, Translation with a Top Down Parser, More about Translation: Array Reference in Arithmetic Expressions, Procedure Calls, Declaration, and Case Statements.

Unit IV: Symbol Table **10 Lecture Hours**

Data Structures for Symbol Tables, Representing Score Information, Run Time Administration: Implementation of Simple Stack Allocation Scheme, Storage Allocation in Block Structures Language, Error Detection and Recovery: Lexical Phase Error, Syntactic Phase Errors, Semantic Phase Errors.

Unit V: Introduction to Code Optimization **5 Lecture Hours**

Loop Optimization, the DAG Representation of Basic Blocks, Value Number and Algebraic Laws, Global Data-Flow Analysis

Total lecture Hours 45

Textbooks

1. Alfred V. Aho, Ravi Sethi, and Jeffrey D. Ullman, "Compilers- Principles, Techniques, and Tools", 2nd Edition, Pearson Education Asia, 2013.
2. Robin Hunter, "The Essence of Compiler", 2nd Edition, Pearson Publication, 2004.

Reference Books

1. Randy Allen, and Ken Kennedy, "Optimizing Compilers for Modern Architectures: A Dependence-based Approach", Morgan Kaufmann Publishers, 2002.
2. Steven S. Muchnick, "Advanced Compiler Design and Implementation", Morgan Kaufmann Publishers - Elsevier Science, India, Indian Reprint 2003.
3. Keith D Cooper, and Linda Torczon, "Engineering a Compiler", Morgan Kaufmann Publishers Elsevier Science, 2004.
4. Charles N. Fischer, and Richard. J. LeBlanc, "Crafting a Compiler with C", Pearson Education, 2008.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSEG3057	Statistics and Data Analysis		3	0	0	3
Total Units to be Covered:		**Total Contact Hours:**				
Prerequisite(s):			**Syllabus version: 1.0**			

Course Code	Course name		L	T	P	C
PROJ3154	Minor Project		0	0	0	5
Total Units to be Covered:		Total Contact Hours:				
Prerequisite(s):			Syllabus version: 1.0			

Course Objectives

The objective of Minor Project is to develop a software application showcasing algorithm design and its subsequent implementations to solve the real-world problems.

Course Outcomes

On completion of this course, the students will be able to

CO1. Apply concepts of Data Structures, Algorithm design and Procedural Programming in the software application.

CO2. Use knowledge of Software engineering, computer networks, operating systems and domain of specialization to formulate and implement the problem statement.

CO3. Create a report capturing the entire lifecycle of project carried out in semester.

CO4. Develop a working software to department that meets the approved objectives and justifies the title of the project.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	2	3	2	2	-	-	1	1	1	2	2	3	2	-
CO 2	2	2	3	2	2	-	-	1	1	1	2	2	3	2	-
CO 3	2	2	3	2	2	-	-	1	1	1	2	2	2	-	-
CO 4	2	2	3	2	2	-	-	1	1	1	2	2	2	-	-
Average	2	2	3	2	2	-	-	1	1	1	2	2	2.5	2	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

This course needs no curated course content.

Textbooks

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA (Synopsis)	MID SEM	End Sem	Total
Weightage (%)	25	25	50	100

Course Code	Course name	L	T	P	C
EMPL004	EDGE – Advance Communication II	1	0	0	0
Total Units to be Covered:		Total Contact Hours:			
Prerequisite(s):		Syllabus version: 1.0			

SEMESTER VII

Course Code	Course name		L	T	P	C
PROJ4145	Capstone Project - Phase-1		0	0	0	5
Total Units to be Covered:		Total Contact Hours:				
Prerequisite(s):	Minor Project		Syllabus version: 1.0			

Course Objectives

As entity of university curriculum, it is mandatory for pupil to undertake a real world project. Aim of the Capstone Project is to groom pupil's knowledge for solving technical problems through well-structured project to build holistic competency. Capstone Project -I is to be carried out by all students compulsorily to practice the theoretical concepts learnt in three years of program. The objective of Capstone Project -I is to create a software application showcasing knowledge of software engineering, software design, software architecture, domain of specialization and its subsequent implementation in any programming language. Moreover, thorough knowledge of algorithmic Efficiency and in-depth study of literature, along with acquaintance in integrated development environment in necessary.

Course Outcomes

On completion of this course, the students will be able to

CO1. Apply domain specific knowledge for software development to solve an industry oriented problems.

CO2. Use knowledge of software engineering, software design, software architecture and components to formulate and implement the solution.

CO3. Design a report capturing the entire lifecycle of project carried out in the semester.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	2	1	2	1	-	-	3	1	1	2	2	3	3	3

CO 2	2	2	1	2	1	-	-	3	1	1	2	2	3	3	3
CO 3	2	2	1	2	1	-	-	3	3	3	2	2	2	3	3
Average	2	2	1	2	1	-	-	3	2	1.6	2	2	2.6	3	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

This course needs no curated course content.

Textbooks

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA (Synopsis)	MID SEM	End Sem	Total
Weightage (%)	25	25	50	100

Course Code	Course name	L	T	P	C
SIIB4102	Summer Internship	0	0	0	1
Total Units to be Covered:		Total Contact Hours:			
Prerequisite(s):		Syllabus version: 1.0			

Course Objectives

The objective is to gain the exposure of industrial practices.

Course Outcomes

Students will be able to
CO1: Understand the industrial processes and practices.

CO2. Apply the knowledge to develop, manage and implement engineering solutions within the IT industry.

CO3. Learn and understand various skills along with professional ethics practiced by the industry.

CO4. Communicate and present the technical knowledge effectively.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	3	-	-	-	2	2	-	-	-	-	3	3	2	3
CO 2	-	3	-	-	-	-	-	-	-	-	-	-	3	-	2
CO 3	3	3	-	-	-	2	2	3	3	-	-	-	3	3	2
CO4	-	-	-	-	-	2	-	-	3	-	-	1	-	2	2
Average	1.5	2.25	-	-	-	1.5	1	.75	1.5	-	-	1	2.25	1.75	2.25

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	**Report Submission/ Presentation/ Q&A**
Weightage (%)	100

Detailed breakup of Internal Assessment

SEMESTER VIII

Course Code	Course name		L	T	P	C
PROJ4146	Capstone Project - Phase-2		0	0	0	5
Total Units to be Covered:		Total Contact Hours:				
Prerequisite(s):		Syllabus version: 1.0				

Course Objectives

Course Outcomes

On completion of this course, the students will be able to

CO 1. Apply domain specific knowledge for software development to solve an industry oriented problems.

CO 2. Use knowledge of software engineering, software design, software architecture and components to formulate and implement the solution.

CO 3. Design a report capturing the entire lifecycle of project carried out in the semester.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	2	1	2	1	-	-	3	1	1	2	2	3	3	3
CO 2	2	2	1	2	1	-	-	3	1	1	2	2	3	3	3
CO 3	2	2	1	2	1	-	-	3	3	3	2	2	2	3	3
CO4	2	2	1	2	1	-	-	3	3	1	2	2	2	3	3
Average	2	2	1	2	1	-	-	3	2	1.5	2	2	2.5	3	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

Syllabus

This course needs no curated course content.

Textbooks

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA (Synopsis)	MID SEM	End Sem	Total
Weightage (%)	20	30	50	100

Course Code	Course name		L	T	P	C
CSEG4038	IT Ethical Practices		3	0	0	3
Total Units to be Covered: 6		Total Contact Hours: 45				
Prerequisite(s):		Syllabus version:				

Course Objectives

1. Define ethics and its relevance to the field of IT.
2. Identify ethical principles and frameworks applicable to IT practices.
3. Recognize and analyze ethical dilemmas in the context of AI, Cybersecurity, Secure Coding, IPR and Copyright, and Metaverse
4. Apply ethical principles to decision-making in IT projects and practices.
5. Communicate effectively about ethical issues in IT.

Course Outcomes

Understand the meaning of ethics and its significance in the context of information technology (IT).

CO1. Apply ethical principles and frameworks, such as the ACM Code of Ethics and Professional Conduct, to guide decision-making in IT scenarios.

CO2. Evaluate ethical dilemmas arising from emerging technologies such as Artificial Intelligence (AI), Cybersecurity, Secure Coding, IPR and Copyright, and Metaverse.

CO3. Integrate ethical considerations into the planning, development, implementation, and evaluation of IT projects and practices.

CO4. Communicate about ethical issues clearly and persuasively in IT, both orally and in writing.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PS01	PS02	PS03
CO1	-	-	-	-	-	2	-	-	-	-	-	-	-
CO2	1	1	-	-	-	-	-	-	-	-	-	-	-

CO3	1	1	2	-	-	1	-	-	2	-	-	-	-
CO4	-	-	-	-	-	1	-	1	-	-	-	-	-
Average	1	1	2	-	-	1.33	-	1	2	-	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**
3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to IT Ethics **5 Lecture Hours**

Defining ethics and its role in IT, Ethical principles and frameworks, Ethical dilemmas in IT.

Unit II: Ethical Considerations in Artificial Intelligence (AI) **6 Lecture Hours**

AI and its impact on society, Bias and fairness in AI, Privacy and data protection in AI, Human autonomy, and control in AI.

Unit III: Ethical Issues in Cybersecurity **8 Lecture Hours**

Cybersecurity threats and vulnerabilities, Protecting privacy and confidentiality in cyberspace, Ethical hacking and vulnerability disclosure, Cyber warfare and cybercrime.

Unit IV: Secure Coding Practices **8 Lecture Hours**

Software security principles and vulnerabilities, Secure coding techniques and best practices, Ethical implications of software vulnerabilities, Security testing and vulnerability assessment.

Unit V: Intellectual Property Rights (IPR) and Copyright in IT **10 Lecture Hours**

Understanding IPR and copyright laws, Protecting intellectual property in IT, Ethical considerations in copyright and intellectual property, Open-source software, and licensing

Unit VI: Ethical Issues in the Metaverse **10 Lecture Hours**

Defining the Metaverse and its potential impact, Privacy and security concerns in the Metaverse, Ethical considerations in avatar representation and interactions, Virtual economies and ethical dilemmas.

Total lecture Hours 47

Textbooks

1. Boddington, Paula. Towards a code of ethics for artificial intelligence. Cham: Springer, 2017.
2. Ethics for the Information Age by Michael J. Quinn
3. Christen, M., Gordijn, B. and Loi, M., 2020. The ethics of cybersecurity (p. 384). Springer Nature.
4. LeBlanc, D. and Howard, M., 2002. Writing secure code. Pearson Education.
5. Rockman, H.B., 2004. Intellectual property law for engineers and scientists. John Wiley & Sons.
6. Manjikian, M., 2017. Cybersecurity ethics: an introduction. Routledge.

Reference Books

1. Gold, Tandy. *Ethics in IT outsourcing*. CRC Press, 2012.
2. Koehn, Daryl. *The ground of professional ethics*. Routledge, 2006.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA(Synopsis Evaluation)	Mid Sem	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Specialization Tracks (Program Electives)

1. Artificial Intelligence and Machine Learning Track

Course Code	Course name		L	T	P	C
CSAI2016P	Applied Machine Learning		4	0	0	4
Total Units to be Covered: 07		Total Contact Hours: 60				
Prerequisite(s):	Basic knowledge of mathematics, problem solving and artificial Intelligence		Syllabus version: 1.0			

Course Objectives

1. Understand the core concepts and techniques of machine learning and artificial intelligence.
2. Develop machine learning models using popular libraries and frameworks.
3. Evaluate the performance of machine learning models using appropriate metrics.
4. Apply machine learning to various real-world problems and domains.

Course Outcomes

On completion of this course, the students will be able to

1. Recall and define key machine learning concepts, terminologies, and algorithms.
2. Describe the differences between supervised, unsupervised, and reinforcement learning.
3. Apply data preprocessing techniques to clean, transform, and prepare datasets for machine learning.
4. Apply, compare, and contrast the strengths and weaknesses of different machine learning algorithms.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	1	1	-	-	-	-	-	-	-	-	-	1	-	3
CO 2	1	1	1	-	-	-	-	-	-	-	-	-	1	-	3
CO 3	1	1	1	-	-	-	-	-	-	-	-	-	1	-	3
CO4	1	1	1	-	-	-	-	-	-	-	-	-	2	-	3

Average	1	1	1	-	-	-	-	-	-	-	-	-	1.25	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction **3 Lecture Hours**

Overview of machine learning and its applications, Types of machine learning: supervised, unsupervised, reinforcement, Python and libraries for machine learning (e.g., NumPy, Pandas, scikit-learn)

Unit II: Loss functions **3 Lecture Hours**

Mean Squared Error (MSE), Mean Absolute Error (MAE), Huber Loss, Binary Cross-Entropy Loss (Log Loss), Categorical Cross-Entropy Loss, Sparse Categorical Cross-Entropy Loss, Hinge Loss (SVM Loss), Triplet Loss

Unit III: Optimizer function **6 Lecture Hours**

Stochastic gradient descent, Mini-Batch Gradient Descent, Momentum, Adaptive gradient algorithm (Adagrad), Adam (Adaptive Moment Estimation), RMSprop (Root Mean Square Propagation), Adadelta

Unit IV: Data Preprocessing **10 Lecture Hours**

Data Cleaning: handling Missing Data, Handling Outlier, Data Transformation: Feature Scaling, Feature Encoding, Feature Engineering, Data Reduction: Dimensionality reduction technique, feature selection, Data Splitting: Cross validation techniques, Handling imbalanced data: Oversampling techniques, under sampling techniques.

Unit V: Regression **12 Lecture Hours**

Introduction to Regression, Regression examples, Regression models, Steps in regression analysis, Linear regression, Simple linear regression, Mathematical proof of linear regression, Least squares estimation, Least squares regression-Line of best fit, Illustration, Direct regression method, Maximum likelihood estimation, Coefficient of determination (R-squared), Checking model adequacy, Over-fitting, Detecting over-fit models: Cross validation, Logistic regression, Mathematical proof of logistic regression, multiple linear regression, Multiple linear regression model building, Mathematical proof of Multiple linear regression model, Interpretation of multiple linear regression coefficients-Partial regression coefficients, Standardized regression coefficients, Missing data, Validation of multiple regression model, regularization, ridge and lasso regularization.

Unit VI: Classification **14 Lecture Hours**

Introduction, ML classifier, Classification and general approach, Classification algorithms, Instance based learning, K-Nearest neighbour, Decision trees, Attribute selection measure: Information gain, ID3 algorithm, Converting a tree to rules, Bayesian algorithms, Ensemble, Ensemble of classifiers, Bagging, Boosting, Random forests, Neural networks, Activation functions, Feedforward neural network, Multi-layer perceptron, Back propagation algorithm, Recurrent or feedback architecture, Perceptron rule, Multilayer networks and back propagation algorithm, Support vector machine, Classification model evaluation and selection, ROC curves, AUC curves.

Unit VII: Clustering Techniques **12 Lecture Hours**

Introduction to Clustering, Clustering algorithms, Statistics associated with cluster analysis, General applications of clustering, Clustering as a pre-processing tool, Similarity and dissimilarity between objects, Type of data in clustering analysis, Binary variables, Nominal variables, Ordinal variables, Cluster centroid and distances, Hierarchical clustering, Hierarchical Agglomerative Clustering (HAC), Hierarchical Agglomerative Clustering: Linkage method, Hierarchical Agglomerative Clustering: Variance and Centroid method, Cluster distance measures, agglomerative clustering, Distance between two clusters, Hierarchical clustering: Time and Space requirements, K - means clustering, The K-medoids clustering method, CLARA (Clustering Large Applications), Density based clustering methods, DBSCAN.

Total lecture Hours 60

Textbooks

1. Introduction to Machine Learning with Python" by Andreas C. Müller and Sarah Guido

2. Pattern Recognition and Machine Learning" by Christopher M. Bishop

Reference Books

1. Machine Learning: A Probabilistic Perspective" by Kevin P. Murphy

2. Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSAI2116P	Applied Machine Learning Lab	0	0	2	1
Total Units to be Covered: 14		Total Contact Hours: 30			
Prerequisite(s):	Mathematics, Python Programming Language	Syllabus version: 1.0			

Course Objectives

The objectives of a machine learning lab course are to provide students with hands-on experience in applying machine learning concepts, techniques, and tools to real-world data. The lab component complements the theoretical understanding gained in a machine learning lecture course.

Course Outcomes

1. Design and implement custom machine learning solutions for real-world problems.
2. Evaluate the performance of machine learning models through critical analysis of various metrics and model variations.
3. Apply machine learning techniques to diverse datasets and domains, demonstrating practical problem-solving skills.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	-	-	-	-	-	-	-	-	-	-	-	-	3
CO 2	-	-	-	1	1	-	-	1	1	-	-	-	-	-	3
CO 3	-	1	1	-	-	1	-	-	-	2	-	1	-	-	3
Average	.3	.3	.3	.3	.3	.3	-	.3	.3	.67	0	.3	0	0	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Below is a list of small machine learning-based projects suitable for this lab work. Through these projects students are expected to implement the concepts of data preprocessing and machine learning algorithms. These projects cover various machine learning techniques and can serve as valuable learning experiences:

Experiment 1 Predicting Housing Prices: Develop a regression model to predict house prices based on features like location, size, and amenities.

Experiment 2 Iris Flower Classification: Use the Iris dataset to build a classification model that predicts the species of iris flowers.

Experiment 3 Handwritten Digit Recognition: Implement a digit recognition system using the MNIST dataset and a neural network.

Experiment 4 Breast Cancer Diagnosis: Develop a breast cancer classification model using medical imaging data (e.g., mammograms).

Experiment 5 Sentiment Analysis: Create a sentiment analysis tool that classifies text reviews as positive or negative using natural language processing (NLP) techniques.

Experiment 6 Spam Email Detection: Build a spam email filter using text classification algorithms.

Experiment 7 Predicting Stock Prices: Develop a time series prediction model to forecast stock prices.

Experiment 8 Credit Risk Assessment: Build a credit scoring model to assess the creditworthiness of applicants using historical financial data.

Experiment 9 Recommendation System: Create a movie or book recommendation system based on user behavior data (collaborative or content-based).

Experiment 10 Anomaly Detection: Implement an anomaly detection system for detecting outliers in data (e.g., fraud detection).

Experiment 11 Customer Churn Prediction: Develop a model to predict customer churn in a subscription-based business.

Experiment 12 Fake News Detection: Create a model to classify news articles as real or fake based on their content.

Experiment 13 Disease Diagnosis from Medical Images: Use medical imaging data (e.g., X-rays) to diagnose diseases or conditions.

Experiment 14 Traffic Sign Recognition: Build a model that can recognize and classify traffic signs in images or video streams.

Total Lab hours 30

Textbooks

1. Andreas C. Müller and Sarah Guido, "Introduction to Machine Learning with Python", Shroff/O'Reilly, 2016.
2. Christopher M. Bishop , "Pattern Recognition and Machine Learning", Springer, 2016.

Reference Books

1. Kevin P. Murphy, "Machine Learning: A Probabilistic Perspective", MIT Press, 2012.
2. Sebastian Raschka and Vahid Mirjalili, "Python Machine Learning", 2nd Edition, Packt Publishing, 2017.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	**Quiz & Viva**	**Performance & Lab Report**
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSAI3025P	**Deep Learning**	4	0	0	4
Total Units to be Covered: 05	**Total Contact Hours: 60**				
Prerequisite(s):	Basic knowledge of mathematics and machine learning	**Syllabus version: 1.0**			

Course Objectives

1. Understand the basic concepts of neural network.
2. Understand the fundamental concepts, usage and impact of neural network, deep learning algorithms in various domains.
3. Discuss various deep-learning algorithms to solve real life problems.

Course Outcomes

On completion of this course, the students will be able to

CO1. To know the concepts of neural networks.

CO2. Discuss the deep learning concepts corresponding to different applications.

CO3. Comprehend the contemporary techniques in deep learning.

CO4. Analyse the concept of convolutional neural networks, recurrent neural networks, generative. deep learning and its usage.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	1	-	-	-	-	-	-	-	-	-	-	-	3
CO 2	1	-	1	-	-	-	-	-	-	-	-	-	-	-	3
CO 3	1	-	1	-	1	-	-	-	-	-	-	-	-	-	3
CO4	1	-	2	-	1	-	-	-	-	-	-	-	-	-	3
Average	1	-	1.25	-	0.5	-	-	-	-	-	-	-	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Fundamentals of Neural Networks **10 Lecture Hours**

History of Deep Learning, McCulloch Pitts Neuron, Multilayer Perceptrons (MLPs), Representation Power of MLPs, Sigmoid Neurons, Feed Forward Neural Networks, Back propagation, Gradient Descent (GD), Momentum Based GD, Nesterov Accelerated GD, Stochastic GD, Singular Value Decomposition, Parameters v/s Hyper-parameters

Unit II: The Math behind Neural Networks **12 Lecture Hours**

Computation in neural network, The matrix magic, Visualizing deep learning, The elephant in the room, Programmatic expression of deep learning's math constructs, Operations with the tensors, Array broadcasting, Scalar product/Inner product of tensors, Morphing shapes of tensors, Gradient calculation.

Unit III: Neural Networks and Deep Learning Basics **13 Lecture Hours**

Hebbian learning, Hebbian learning modifications: Mathematical models, Competitive learning, Error-correction learning, Boltzmann learning, Learning tasks: Pattern association, Learning tasks: Pattern recognition and function approximation, Neurons, weights, biases, transfer functions and cost/loss functions, Generalization and Overfitting, GD and Optimizers, Parameter Initialization and Hyperparameter Tuning, Data and Batch Normalization and Regularization methods, Data preparation and label preparation, Data Augmentation, CNN, Visualization of 2D convolution, Visualization of 3D convolution, Deep Neural Networks

Unit IV: Deep Learning Model **12 Lecture Hours**

Image Classification, Image segmentation, Semantic Segmentation, Instance based segmentation, Object detection, R-CNN, Fast R-CNN, Faster R-CNN, YOLO, ResNet, GoogleNet, RatinaNet,

Unit V: Advanced Deep Learning **13 Lecture Hours**

Introduction to DL packages/ Important architectures, Deep RNNs, Recursive neural networks, Step function, Tanh function, Recurrent Neural Networks, LSTM, Generative Modelling using Deep networks, Variational Auto Encoders (VAE), Latent space, Generative Adversarial Networks (GAN's), Transformers, Variational Auto Encoders (VAE)

Total lecture Hours 60

Textbooks

1."Introduction to Machine Learning with Python" by Andreas C. Müller and Sarah Guido

2. "Pattern Recognition and Machine Learning" by Christopher M. Bishop

Reference Books

1."Machine Learning: A Probabilistic Perspective" by Kevin P. Murphy

2. "Python Machine Learning" by Sebastian Raschka and Vahid Mirjalili

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSAI3125P	Deep Learning Lab		0	0	2	1
Total Units to be Covered: 11		Total Contact Hours: 30				
Prerequisite(s):	Mathematics, Python Programming Language, Machine Learning		Syllabus version: 1.0			

Course Objectives

1. Understand the basic concepts of neural network
2. Understand the fundamental concepts, usage and impact of deep learning algorithms in various domain.
3. Discuss various deep-learning algorithms to solve real life problems dependent on image data.

Course Outcomes

On completion of this course, the students will be able to

CO1. To implement the basic deep learning frameworks

CO2. Implement the deep learning concepts corresponding to different datasets.

CO3. Implement the convolutional neural networks, recurrent neural networks, generative deep learning for object detection and classification

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	1	1	-	-	-	-	-	1	-	-	-	-	-	3
CO 2	1	1	1	-	-	-	-	-	1	-	-	-	-	-	3
CO 3	1	1	1	-	1	-	-	-	1	-	2	2	-	2	3
Average	1	1	1	-	0.3	-	-	-	1	-	.6	.6	-	.6	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment1 Introduction to Deep Learning: Implement a basic neural network using a deep learning framework like TensorFlow or PyTorch for a simple classification task.

Experiment2 Convolutional Neural Networks (CNNs): Build a CNN architecture for image classification on a dataset like CIFAR-10 or MNIST.

Experiment3 Transfer Learning: Fine-tune a pre-trained CNN model (e.g., VGG16, ResNet) on a custom dataset to solve a related task.

Experiment4 Recurrent Neural Networks (RNNs): Create an RNN model for sequence data like text generation or sentiment analysis.

Experiment5 Object Detection: Use a pre-trained object detection model (e.g., YOLO or Faster R-CNN) to identify objects in images or video streams.

Experiment6 Automatic Image Captioning with Keras

Experiment7 Facial Recognition

Experiment8 Digit Recognition

Experiment9 Hand Movement Recognition

Experiment10 Create synthetic datasets with the help of GANs.

Experiment11 11 Use of transfer learning for image classification.

Total Lab hours 30

Textbooks

1. Introduction to Deep Learning (Neural Networks) (IBM ICE Publication).

2. Bengio Y, Goodfellow I, Courville A. Deep learning. Cambridge, MA, USA: MIT press; 2017.

Reference Books

1. Ravichandiran S. Hands-On Deep Learning Algorithms with Python: Master deep learning algorithms with extensive math by implementing them using TensorFlow. Packt Publishing Ltd; 2019 Jul 25.

2. Shanmugamani R. Deep Learning for Computer Vision: Expert techniques to train advanced neural networks using TensorFlow and Keras. Packt Publishing Ltd; 2018 Jan 23.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSAI3026P	Pattern and Visual Recognition	4	0	0	4
Total Units to be Covered: 05	Total Contact Hours: 60				
Prerequisite(s):	Mathematics, Data Mining, Machine Learning	Syllabus version: 1.0			

Course Objectives

1. To understand fundamentals of pattern and its computational significance.
2. To study various approaches through which pattern recognition can be carried out

Course Outcomes

After completion of course, students would be able to:

CO1 Understand basic mathematical and statistical techniques commonly used in pattern recognition.

CO2 Understand and apply various pre-processing algorithms.

CO3 Apply a variety of pattern recognition algorithms.

CO4 Apply various algorithms for image classification.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	-	-	2	-	-	-	-	-	-	-	-	-	3
CO 2	1	-	-	-	2	-	-	-	-	-	-	-	-	-	3
CO 3	1	-	-	-	2	-	-	-	-	-	-	-	-	-	3
CO4	1	-	-	-	2	-	-	-	-	-	-	-	-	-	3
Average	1	-	-	-	2	-	-	-	-	-	-	-	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

Syllabus

The course is aimed at specific problem, pattern recognition that is often encountered in image processing applications. Various methods that are available under statistical and learning approaches through which pattern analysis especially from recognition and detection purpose is addressed in this course.

Unit I: Image representation and analysis **10 Lecture Hours**

Introduction to Digital Image Processing, Numerical representation of images, Image augmentation, enhancement, processing, color transforms, geometric transforms, feature recognition and extraction, Eigen values and eigen vectors, Rank of matrix and SVD

Unit II: Feature Selection and extraction **8 Lecture Hours**

Problem statement and uses, Branch and bound algorithm, Sequential forward and backward selection, Cauchy Schwartz inequality, Feature selection criteria function: Probabilistic separability based and Interclass distance based, Feature Extraction: principles

Unit III: Machine Learning Approaches for **20 Lecture Hours**

Pattern Recognition

Neural networks, how neural networks learn? Neural networks examples, Neural networks use cases, Kernel methods, Sparse kernel machines use cases, Graphical models, Mixture models and EM, Bayesian networks: Directed graphical models, Conditional probability distributions, Potential functions, Conditional independences, Sampling methods for pattern recognition, Continuous latent variables, Combining models for pattern recognition, Hidden Markov models, Markov chain Monte Carlo, The K-means algorithm, Applications of K-means.

Unit IV: Visual Recognition **12 Lecture Hours**

Human visual recognition system, Recognition methods: Low-level modelling (e.g. features), Mid-level abstraction (e.g. segmentation), High-level reasoning (e.g. scene understanding); Detection/Segmentation methods; Context and scenes, Importance and saliency, Large-scale search and recognition, Egocentric vision, systems, Human-in-the-loop interactive systems, 3D scene understanding.

Unit V: Recent advancements in Pattern Recognition **10 Lecture Hours**

Comparison between performance of classifiers, Basics of statistics, covariance and their properties, Data condensation, feature clustering, Data visualization, Probability density estimation, Visualization and Aggregation, FCM and soft-computing techniques, Examples of real-life datasets.

Total lecture Hours 60

Textbooks

1. Pattern and Anomaly Detection (IBM ICE Publications)

2. Szeliski R. Computer vision: algorithms and applications. Springer Nature; 2022 Jan

3. Bishop CM. Pattern Recognition and Machine Learning by Christopher M. Bishop. Springer Science+ Business Media, LLC; 2006.

Reference Books

1. Richard O. Duda, Peter E. Hard, David G. Stork, Pattern Recognition, 2nd, Wiley, 2021.

2. Prince SJ. Computer vision: models, learning, and inference. Cambridge University Press; 2012 Jun 18.

3. Theodoridis S, Koutroumbas K. Pattern recognition. Elsevier; 2006 Apr 7.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSAI3126P	**Pattern and Visual Recognition Lab**		0	0	2	1
Total Units to be Covered: 11		**Total Contact Hours: 30**				
Prerequisite(s):	Programming skills in python, Machine Learning, Mathematics	**Syllabus version: 1.0**				

Course Objectives

1. To gain knowledge on the process of Pattern and Visual Recognition.
2. To learn about the implementation of pattern and object detection.
3. To understand the analysis and validation of models over the datasets.

Course Outcomes

At the end of this course student should be able

CO1: Understand the concept of a pattern and the basic approach to the development of pattern recognition and machine intelligence algorithms.

CO2: Analyze the statistical approaches and apply patter recognition techniques to detectand characterize patterns in real-world data.

CO3: Comprehend the contemporary techniques in machine learning for pattern and visual recognition. to real-world problems such as face, nuclei, real world objects etc.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	1	2	-	-	-	-	-	-	-	-	1	-	-	3
CO 2	1	1	2	1	1	-	-	-	-	-	-	1	-	-	3
CO 3	1	1	2	1	1	-	-	-	-	-	-	1	-	-	3
Average	1	1	2	1	1	-	-	-	-	-	-	1	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment No 1 Installing Anaconda and setup up the environment, installation of supporting packages, Linear Regression

Experiment No 2 Linear Regression and Non-Linear Regression, Linear and Non-Linear Curve fitting for regression type of problem, Error correction methods, MSE and MAE

Experiment No 3 Logistic Regression and Sigmoid function for classification-based problem.TPR, FPR, TNR, FNR, Recall, Precision, Sensitivity, ROC-AUC curve analysis

Experiment No 4 Un-Supervised Learning, k-Means Clustering & Elbow Rule

Experiment No 5 Image Segmentation: Explore techniques like thresholding, region-based segmentation, and edge-based segmentation to partition an image into meaningful regions.

Experiment No 6 Semantic Segmentation using deep learning models

Experiment No 7 Instance segmentation using deep learning models

Experiment No 8 Build a CNN architecture for image classification on a dataset like CIFAR-10 or MNIST.

Experiment No 9 Isolated Random Forest Analyse and identify patterns in data to predict and take action in biological images and data to analyse DNA, RNA, and nuclei patterns to study evolution and identify patterns in gene or cell expression data to uncover new biomarkers

Experiment No 10 Isolated Random Forest Analyse and identify patterns in data to predict and take action in biological images and data to analyse DNA, RNA, and nuclei patterns to study evolution and identify patterns in gene or cell expression data to uncover new biomarkers

Experiment No 11 Neural network for Pattern and hand Movement Recognition

Total Lab hours 15

Textbooks

1. Book provided by IBM- Data Mining & Prediction Modeling (Course code GAI06SG1 V1.0)

Reference Books

3. Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig, c 1995 Prentice-Hall, Inc

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSEG4034P	**Computational Linguistics and Natural Language Processing**	4	0	0	4
Total Units to be Covered: 05	**Total Contact Hours: 60**				
Prerequisite(s):	a. Basic Arithmetic, Statistics & Probability b. Data Structures & Algorithm c. Knowledge of Automata Theory & Compiler Design	**Syllabus version: 1.0**			

Course Objectives

1. The objectives of this course are:
2. To introduce the concept of Natural Language Understanding & Natural Language Generation.
3. To develop the concept of statistical and probabilistic approach of language modelling.
4. To extend the knowledge of Large Language Model.
5. To enrich the knowledge with different corpuses and different tools being used for machine translation.
6. To provide programming skills necessary for processing natural language.

Course Outcomes

On completion of this course, the students will be able to

CO1: Understand the techniques in NLP.

CO2: To understand the Large Language Model

CO3: To comprehend the natural language generation.

CO4: To understand the machine translation & information retrieval techniques.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	-	1	1	-	-	-	-	-	-	-	-	-	3
CO 2	1	-	-	1	1	-	-	-	-	-	-	-	-	-	3

CO 3	1	2	-	2	1	-	-	-	-	-	-	-	-	-	3
CO4	1	-	-	3	1	-	-	-	-	-	-	-	-	-	3
Average	1	.5	-	1.75	1	-	-	-	-	-	-	-	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Classical Approaches of NLP **12 Lecture Hours**

Introduction, Classical approaches to natural language processing , Understanding linguistics, Level 1: Morphology, Level 2: Syntax, Level 3: Semantics, Level 4: Pragmatics, Understanding linguistics, Traditional approach, Example: Automatic summarization using NLP, Drawbacks, Text processing, Ambiguities and computational challenges in processing various natural languages, Introduction to Real life applications of NLP such as spell and grammar checkers, information extraction, question answering, and machine translation.

Unit II: Empirical & Statistical Approaches **12 Lecture Hours**

Corpus creation, Corpus linguistics, Types of corpora, Lexicographical implementations in corpora, Timeline of corpus linguistics, Usage areas of corpora, Traits of a good text corpus, Annotations in text corpus, NLP task-specific training corpora, Treebank annotation, Usage of annotations and corpora, Kinds of annotations, Tree banks and its construction, Need for tree bank, Types of tree bank corpus, Ambiguity in language, Segmentation, Stemming, Tokenization, Representation of word, Sentence, Word embedding, Word Senses, Linguistic Structure: Dependency Parsing. Fundamental statistical techniques, Problems of the traditional approach, how statistics helps, Problems of the traditional approach and how statistics helps, Hidden Markov model, Maximum entropy Markov model, Conditional random field model, Support vector machine, N-GRAM, Perplexity, POS Tagging, Word sense disambiguation, POS tag and Hidden Markov model, POS tagging using HMM, Viterbi algorithm, Recurrent Neural network, Vanishing Gradients and exploding gradients. Parsing, Statistical parsing, Approaches to parsing, Statistical approach, Lexicalized statistical parsing, Top-down parsing, Bottom-up parsing, Left corner parsing method, Statistical parsing: Probabilistic parser,

Unit III: Language Modelling **12 Lecture Hours**

Word similarity and text similarity, Text similarity methods, Jaccard similarity, K-means, Cosine similarity, Word Mover's distance, Word sense disambiguation, Complications in WSD, Methods in WSD, Evaluation of WSD, the role of language models, estimating parameters and smoothing. Evaluating language models, LSTM (Long short-term memory), GRU (Gated recurrent Unit), Part of speech tagging, BERT, XLnet, 1D-CNN for NLP, Sub-word Models, Contextual Representations, Transformers, Self-Attention for Generative Models.

Unit IV: Machine Translation **12 Lecture Hours**

Machine translation, Rule-based machine translation, Statically Machine Translation, Neural Machine Translation, Seq2Seq Modelling, Attention, Question Answering Bot, Natural Language Generation, Neural Machine Translation, Case studies on Amazon Alexa, Google Assistant, Microsoft Cortona etc.

Unit V: Applications of Natural Language Processing **12 Lecture Hours**

Text Summarization, Document Summarization, Sentiment Analysis, Question Answering system, Sarcasm Detection, Hostile detection Information retrieval in NLP, Image caption generation, Intelligent Tutoring System

Total lecture Hours 60

Textbooks

1. Speech and Language Processing "An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition" Second Edition Daniel Jurafsky Stanford University James H. Martin University of Colorado at Boulder

Reference Books

1. "Information Retrieval: Algorithms and Heuristics" by David A. Grossman, Ophir Frieder, Academic Press.

2. Natural Language Processing A paninian perspective, Prentice Hall, New Delhi, 1994 Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSEG4134P	**Computational Linguistics and Natural Language Processing Lab**	0	0	2	1
Total Units to be Covered: 10		**Total Contact Hours: 30**			
Prerequisite(s):	a. Basic Arithmetic, Statistics & Probability b. Data Structures & Algorithm c. Knowledge of Automata Theory & Compiler Design d. Python Programming e. Machine Learning	**Syllabus version: 1.0**			

Course Objectives

The objectives of this course are:

1. To introduce the concept of Natural Language Understanding & Natural Language Generation.
2. To develop the concept of statistical and probabilistic approach of language modelling.
3. To extend the knowledge of Large Language Model.
4. To enrich the knowledge with different corpuses and different tools being used for machine translation.
5. To provide programming skills necessary for processing natural language.

Course Outcomes

On completion of this course, the students will be able to

CO1: Understand the techniques in NLP.

CO2: To understand and execute the Large Language Model

CO3: To comprehend and execute the natural language generation.

CO4: To implement the machine translation for various application.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	-	1	1	-	-	-	-	-	-	-	-	-	3
CO 2	1	-	-	1	1	-	-	-	-	-	-	-	-	-	3
CO 3	1	2	-	2	1	-	-	-	-	-	-	-	-	-	3
CO4	1	-	-	3	1	-	-	-	-	-	-	-	-	-	3

Average	1	.5	-	1.75	1	-	-	-	-	-	-	-	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

EXPERIMENT-1:

Title: Installing various packages required for analytics in python and write code in Python for following program

1. Write code to load CSV file containing information about employee of a company in python and draw graph showing average salary department wise.

EXPERIMENT-2:

Title: Text Retrieval

1- Connect to Twitter account and Extract first 100 tweets from it in a file.
2- Study and Implementation of Processing text (Word and Sentence Tokenization)

EXPERIMENT-3:
Title: Processing Data

1. Python code to read a text document and perform basic pre-processing techniques on the text like tokenization, stop-word-removal, lemmatization etc.
2. Study and Implementation of Morphological analysis.

EXPERIMENT-4:
Title: Do text mining on extract data and Accessing text corpus

1. Calculate word count of a given specific document and show top 10 frequent words with their frequency and Create world cloud and show graphically.
2. Study and Implementation of NER (Name Entity Recognition)

EXPERIMENT-5:

Title: POS-Tagging and Tagging and Parsing

1. Categorizing and tagging words in Twitter Data.
2. Study and implementation of POS Tagging and Chunking in a sentence.

EXPERIMENT-6:

Title: Language Processor

1. Implement N–Gram Language Mode and Smoothing.

EXPERIMENT-7:

Title: Do sentimental analysis

1. Analysis of Sentiment and Subjectivity
2. Implement sentimental analysis on IMDB Movie Reviews Dataset.

EXPERIMENT-8:

Title: Do Text Summarization

1. Analysing Meaning of Sentences
2. Implement Text Summarization on IMDB Movie Reviews Dataset.

EXPERIMENT-9 & 10:

Title: Mini-Project on NLP

Implement Mini-Project on NLP applications.

Total Lab hours 15

Textbooks

1. Speech and Language Processing "An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition" Second Edition Daniel Jurafsky Stanford University James H. Martin University of Colorado at Boulder

Reference Books

1. "Information Retrieval: Algorithms and Heuristics" by David A. Grossman, Ophir Frieder, Academic Press.

2. Natural Language Processing A paninian perspective, Prentice Hall, New Delhi, 1994 Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSAI4013P	Algorithm for Intelligent Systems and Robotics	3	0	0	3
Total Units to be Covered: 06	Total Contact Hours: 45				
Prerequisite(s):	Fundamental of AI, machine learning, kinematics, statistics, and mechanics	Syllabus version: 1.0			

Course Objectives

1. To understand the concepts, usage, and impact of intelligent systems in various part of real life like in military, health care, transportation etc.
2. To understand the concepts, usage, and impact of smarter robots in every core of industry

Course Outcomes

On completion of this course, the students will be able to

CO1 To discuss the basic concepts of algorithms for intelligent systems and robotics.

CO2 To use of different types of sensors, tactile, proximity, range etc.

CO3 To discuss basic understanding of intelligent machines supported by kinematics and Mechanics, use of ROS

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	2	1	2	1	-	-	-	-	-	-	-	2	-	2
CO 2	2	2	1	2	2	-	-	-	-	-	-	-	2	-	3
CO 3	2	2	1	2	2	-	-	-	-	-	-	-	2	-	2
Average	2	2	1	2	1.66	-	-	-	-	-	-	-	2	-	2.33

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: System Modeling **8 Lecture Hours**

Biological and Cognitive Paradigms for Robot Design, Declarative-Procedural-Reflexive Hierarchy for Decision Making and Control, Articulated Robots, Joint-Link (Denavit-Hartenberg) Transformations, Mobile Ground Robots, Uninhabited Ground Robots, Intelligent Agents

Unit II: Control System Principles **8 Lecture Hours**

Open- and Closed-Loop Control, Time-Domain and Frequency-Domain Analysis, Optimality and Constraints, Stability and Performance, Adaptation, Control Actuation, Closed-Form and Probabilistic Path Planning

Unit III: Fundamentals of Robotics **7 Lecture Hours**

Introduction, Robot Hardware, Robotic Perception, Planning to Move, Planning Uncertain Movements, Moving, Robotic Software Architectures, Application Domains

Unit IV: Computing, Measurement, State, and **7 Lecture Hours**

Parameter Estimatio

Sensors and Sensing, Formal and Fuzzy Logic, Turing Machines and Concepts of Machine Learning, Analog and Digital Systems, Probability and Error Models, Sensor-Based Estimation, Extended Kalman and Particle Filters, Simultaneous Location and Mapping (SLAM)

Unit V: Knowledge in Learning in Intelligent Systems **8 Lecture Hours**

Logical Formulation of Learning, Knowledge in Learning, Explanation-Based Learning,Learning Using Relevant Information, Inductive Logic Programming

Unit VI: Introduction to ROS **7 Lecture Hours**

Introduction to Robot Operating System (ROS), Application of ROS, Installation steps of ROS, Use of Turtlesim, ROS Gazebo full simulation

Total lecture Hours 45

Textbooks

1. Algorithms for Intelligent Systems and Robotics (IBM ICE Publications)

2. Correll N, Hayes B, Heckman C, Roncone A. Introduction to autonomous robots: mechanisms, sensors, actuators, and algorithms. Mit Press; 2022 Dec 20.

Reference Books

1. J. Craig, "Introduction to Robotics Mechanics and Control", Pearson, 2018.
2. B. Ripley, "Pattern Recognition and Neural Networks", Cambridge University Press, 1996.
3. http://www.cs.rpi.edu/~trink/Courses/AlgorithmicRobotics/fall2017/agitr-letter.pdf
4. Jason M. O'Kane, A Gentle Introduction to ROS, independently published, Updated on 2016-09-06 to version 2.1.3. (Electronic copies freely available from https://www.cse.sc.edu/~jokane/agitr/)

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSAI4113P	Algorithm for Intelligent Systems and Robotics Lab	0	0	2	1
Total Units to be Covered: 12	Total Contact Hours:30				
Prerequisite(s):	Basic programming skills in python, artificial intelligence, and machine learning	Syllabus version: 1.0			

Course Objectives

1. To gain knowledge on the process of system design and AI problems and techniques
2. To learn about the kinematic and dynamic control concepts that focuses on intelligent gripping systems.
3. To understand the operating system concepts for robotics (ROS) and the basic concept of Speech Recognition and Synthesis

Course Outcomes

At the end of this course student should be able

CO1. Implement basic algorithms of machine learning for robotics and system modelling.
CO2. Demonstrate the components of an intelligent robotic system and robot operating system (ROS).
CO3. Deploy algorithms for intelligent systems and robotics to Navigation, SLAM (Simultaneous Localizing and Mapping), Speech Recognition and Synthesis.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	1	1	-	-	-	-	-	-	-	-	-	2	-	3
CO 2	1	1	1	2	3	-	-	-	-	-	-	-	2	-	2
CO 3	1	1	1	2	3	-	-	-	-	-	-	-	2	-	3
Average	1	1	1	2	3	-	-	-	-	-	-	-	2	-	2.67

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment No 1 Time Domain and frequency Domain representation using Python, Pendulum Simulation using Python: example of a system

Experiment No 2 8 queens problem using Python

Experiment No 3 Search Algorithms (BFS, DFS) using python

Experiment No 4 Hill Climbing using python

Experiment No 5 Reinforcement learning using python

Experiment No 6 Simple Neural Network concept using python

Experiment No 7 Compare various learning strategies for MLP classifier

Experiment No 8 Kalman Filtering using python

Experiment No 9 Installing ROS and other packages, basic programs, Testing the Simulator

Experiment No 10 Monitoring Robot motion using Simulator

Experiment No 11 Teleoperating the simulated Robot, Avoiding Simulated obstacles

Experiment No 12 Multiple Turtle bot Simulation, Speech related experiment

Total Lab hours 30

Textbooks

1. Book provided by IBM- Data Mining & Prediction Modeling (Course code GAI06SG1 V1.0)

2. Peter C. Robotics, vision and control: fundamental algorithms in MATLAB.

Reference Books

1. Frank C. Modern Robotics-Mechanics, Planning, and Control. Cambridge University Press; 2017.

2. Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig, c 1995 Prentice-Hall, Inc.

3. An Introduction to Robotics, Dr. Bob Williams, williar4@ohio.edu, Mechanical Engineering, Ohio University, EE/ME 4290/5290 Mechanics and Control of Robotic Manipulators, © 2019 Dr. Bob Productions

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	**Quiz & Viva**	**Performance & Lab Report**
Weightage (%)	50 %	50 %

2. DevOps Track

Course Code	Course name	L	T	P	C
CSDV2009P	DevOps Fundamentals and SCM	4	0	0	4
Total Units to be Covered: 06		Total Contact Hours: 60			
Prerequisite(s):	Elementary knowledge of DevOps	Syllabus version: 1.0			

Course Objectives

1. Learn about the DevOps lifecycle and its stages, including continuous integration, continuous delivery, and continuous deployment.

2. Gain practical knowledge of version control systems and configuration.

3. Understand the importance of security in DevOps and learn how to integrate security practices into the CI/CD pipeline.

4. Learn the principles and techniques of demand forecasting and source code management planning to effectively match with needs.

5. Gain insights into sustainable source code management practices, to create long-term value.

Course Outcomes

On completion of this course, the students will be able to

CO1: Understand the principles and benefits of DevOps and its role in software development and deployment.

CO2: Develop an understanding of compliance requirements and how to incorporate them into DevOps processes.

CO3: Develop a comprehensive understanding of source code management concepts, including its objectives and components.

CO4: Understanding of management principles and techniques, including source code planning and optimization.

CO5: Gain an understanding of source code management practices and their impact on the teams' environment.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	3	-	-	-	1	-	-	-	-	-	-
CO 2	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO 3	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-
CO4	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO5	-	-	2		2	-	-	-	2	-	-	2	-	-	-
Average	-	-	0.4		2	-	-	-	0.6	-	-	0.4	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to DevOps **12 Lecture Hours**

Definition and principles of DevOps, Benefits of DevOps adoption, DevOps culture and mindset, DevOps Lifecycle, Overview of the DevOps lifecycle stages, Continuous Integration (CI) and Continuous Delivery (CD), Automated testing and deployment, Version Control and Configuration Management, Introduction to version control systems (e.g., Git), Branching and merging strategies, Configuration management tools (e.g., Ansible, Puppet)

Unit II: Continuous Integration and Build Automation **12 Lecture Hours**

Introduction to IaC concepts and benefits, Infrastructure provisioning tools (e.g., Terraform, CloudFormation), Managing infrastructure configurations, setting up CI/CD pipelines, Building and packaging applications, Automated testing and code quality checks, Continuous Deployment and Release Management, Strategies for releasing software updates, Managing deployment environments, Release orchestration and rollback strategies

Unit III: Monitoring and Security **6 Lecture Hours**

Importance of monitoring in DevOps, implementing application and infrastructure monitoring, Log aggregation and analysis, Security and Compliance, DevOps security principles and practices, Implementing security controls in CI/CD pipelines, Compliance considerations in DevOps.

Unit IV: SCM and GIT Fundamentals **12 Lecture Hours**

Overview of source code management and its role in software development, Version control systems and their benefits, Introduction to centralized and distributed SCM, Understanding repositories, commits, and revisions, Exploring the concepts of working directory, staging area, and remote repositories, Introduction to branching and tagging, Introduction to Git and its key features, Setting up a Git repository, Committing changes and viewing history, Working with remote repositories, Cloning, pushing, and pulling changes, Collaboration workflows and managing multiple contributors

Unit V: Advanced GIT and Subversion (SVN) **1 2 Lecture Hours**

Branching strategies and best practices, merging changes, and resolving conflicts, Rebasing and cherry-picking, Git hooks and customizing workflows, managing large projects with submodules and subtrees, Git internals and understanding object storage, Overview of Subversion, and its architecture, setting up a Subversion repository, committing changes and viewing history with SVN, Branching and merging in Subversion, Repository organization and access control, Handling conflicts in Subversion

Unit VI: SCM: Systems, Practices, and Integration **6 Lecture Hours**

Evaluating different SCM systems (Git, Subversion, Mercurial, etc.), Strengths and weaknesses of each system, Choosing the right SCM for a given project, best practices for SCM in a team environment, Integrating SCM with development workflows and tools (IDEs, build systems, etc.), Continuous integration and deployment with SCM.

Total lecture Hours 60

Textbooks

1. Gene Kim, Kevin Behr, and George Spafford, "The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win", 5th Edition, Revolution Press, 2018.

2. Jez Humble and David Farley, “Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation", Addison-Wesley Professional, 2010.
3. Emma Jane, “GIT for Teams”, O'Reilly Media, 2015.

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

<table>
<tr><th>Course Code</th><th colspan="2">Course name</th><th>L</th><th>T</th><th>P</th><th>C</th></tr>
<tr><td>CSDV2109P</td><td colspan="2">DevOps Fundamentals and SCM LAB</td><td>0</td><td>0</td><td>2</td><td>1</td></tr>
<tr><td colspan="2">Total Units to be Covered: 08</td><td colspan="5">Total Contact Hours: 30</td></tr>
<tr><td>Prerequisite(s):</td><td colspan="2">Elementary knowledge of DevOps</td><td colspan="4">Syllabus version: 1.0</td></tr>
</table>

Course Objectives

1. Learn about the DevOps lifecycle and its stages, including continuous integration, continuous delivery, and continuous deployment.

2. Gain practical knowledge of version control systems and configuration.

3. Understand the importance of security in DevOps and learn how to integrate security practices into the CI/CD pipeline.

4. Learn the principles and techniques of demand forecasting and source code management planning to effectively match with needs.

5. Gain insights into sustainable source code management practices, to create long-term value.

Course Outcomes

On completion of this course, the students will be able to

CO1: Understand the principles and benefits of DevOps and its role in software development and deployment.

CO2: Develop an understanding of compliance requirements and how to incorporate them into DevOps processes.

CO3: Develop a comprehensive understanding of source code management concepts, including its objectives and components.

CO4: Understanding of management principles and techniques, including source code planning and optimization.

CO5: Gain an understanding of source code management practices and their impact on teams environment.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	3	-	-	-	1	-	-	-	-	-	-
CO 2	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO 3	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-
CO4	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO5	-	-	2	-	2	-	-	-	2	-	-	2	-	-	-
Average	-	-	0.4	-	2	-	-	-	0.6	-	-	0.4	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Creating a Local Git Repository and Executing Git common Commands on It.

Experiment 2: Create a GitHub Repository and Push/Pull/Clone to/from the Local repository.

Experiment 3: Fork a Public Repository to create a Pull Request.

Experiment 4: Git Revert, Git Rebase, and Git Reset Execution.

Experiment 5: Comparison of GitHub, GitLab, and Gutbucket SCM.

Experiment 6: Create an account on AWS and create ECR.

Experiment 7: GitHub Actions for workflow execution.

Experiment 8: Create a CI/CD pipeline using scratch for deploying applications (using Git and Jenkins)

Total Lab hours 30

Textbooks

1. Gene Kim, Kevin Behr, and George Spafford, "The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win", 5th Edition, Revolution Press.
2. Jez Humble and David Farley, “Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation", Addison-Wesley Professional, 2010.
3. “GIT for Teams” by Emma Jane, O’Reilly, 2015.

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	**Quiz & Viva**	**Performance & Lab Report**
Weightage (%)	50 %	50 %

Course Code	Course name		L	T	P	C
CSDV3022P	**DevSecOps: Integrating security into DevOps practices**		4	0	0	4
Total Units to be Covered: 05		**Total Contact Hours: 60**				
Prerequisite(s):	DevOps Fundamentals and SCM		**Syllabus version: 1.0**			

Course Objectives

1. Implement security measures within the DevOps pipeline to identify and remediate vulnerabilities effectively.
2. Foster a culture of collaboration among development, operations, and security teams to ensure continuous security monitoring and response.
3. Automate security testing and compliance checks to maintain a proactive security posture.
4. Apply best practices to secure code, infrastructure, and deployments, meeting compliance standards and minimizing risks.

Course Outcomes

On completion of this course, the students will be able to:

CO1. Recall fundamental DevSecOps principles and terminology to discuss the integration of security within DevOps workflows.

CO2. Demonstrate the ability to implement security measures within CI/CD pipelines and IaC templates to identify and address vulnerabilities.

CO3. Evaluate security scan results, assess risks, and propose improvements for secure software development and deployment.

CO4. Design and execute a collaborative incident response plan, fostering a culture of security awareness and teamwork within DevOps environments.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	2	-	-	3				1	-	-	-	-	-	-
CO 2	-	3	-	-	2		3			-	-	-	-	-	-

CO 3	-	2	-	-	1	-	-	-	-	2	-	-	-	-	-
CO4	-	-	1	-	2	-	-	-	-	-	-	-	-	-	-
CO5	-	-	2	-	2	-	-	-	2	-	-	2	-	-	-
Average	0.6	1.4	0.6	-	2	-	0.6	-	0.6	0.4	-	0.4	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

Syllabus

Unit I: Securing DevOps **12 Lecture Hours**

The DevOps Approach (Continuous integration, Continuous delivery, Infrastructure as a service, Culture, and trust), Security in DevOps, Continuous Security (Test driven security, Monitoring and responding to attacks, Assessing risks, and maturing Security).

Unit II: Protecting Web Applications **12 Lecture Hours**

Securing and testing web Apps, Website attacks, and content security (Cross-site scripting, content security policy, cross-site request forgery, clickjacking, and Iframes protection), Methods for authenticating users (HTTP basic authentication, Password Management, Identity Providers, Sessions and cookie security, Testing Authentication), Managing Dependencies (Node.js package management, Python requirements)

Unit III: Securing Delivery Pipeline **12 Lecture Hours**

Access control to code management infrastructure (managing permissions in GitHub Organization, Managing permissions between GitHub and CircleCI, Signing commits and Tags with GIT), Access control for container storage (Managing permissions between DockerHub and CircleCI, Signing containers with Docker content trust), Access control for infrastructure management (Managing permissions using AWS roles and policies, Distributing secrets to production system)

Unit IV: Maturing DevOps Security: Assessing Risks **12 Lecture Hours**

What is Risk Management? The CIA triad, Establishing the top threats to an organization, Quantifying the impact of risk, Identifying threats and measuring Vulnerabilities, Rapid Risk assessment, recording and tracking risks.

Unit V: Maturing DevOps Security: Testing and Continuous Security **12 Lecture Hours**

Testing Security: Maintaining Security, auditing internal Applications and Services, Red teams and External Pen Testing, and Bug Bounty Programs.

Continuous Security: Practice and repetition: 10,000 hours of Security, Integrating Security into DevOps, Preparing for Worst, Driving the Change.

Total lecture Hours 60

Textbooks

1. Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance" by Tim Mather, Subra Kumaraswamy, and Shahed Latif.
2. Cloud Computing: Principles and Paradigms by Rajkumar Buyya, James Broberg, and Andrzej Goscinski
3. Cloud Computing: Theory and Practice, by Dan C. Marinescu.

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%

Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSDV3122P	DevSecOps: Integrating Security in DevOps Practices Lab		0	0	2	1
Total Units to be Covered: 10			**Total Contact Hours: 30**			
Prerequisite(s):	DevOps Fundamentals and SCM			**Syllabus version: 1.0**		

Course Objectives

1. To equip students with the skills to effectively set up a cloud application development platform, showcasing their ability to configure and deploy the necessary infrastructure and tools.

2. To demonstrate the capability to leverage advanced computing capabilities for the development and deployment of cloud applications,

3. To analyze the implementation processes and challenges involved in deploying cloud applications across different domains.

Course Outcomes

At the end of this course student should be able to

CO1: Develop a basic CI/CD pipeline that incorporates security checks and integrates version control

CO2: Integrate security tools, such as static code analysis and vulnerability scanning, into the CI/CD pipeline.

CO3: Design and participate in a collaborative incident response simulation involving all teams to improve communication and cooperation.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	2	-	-	-	-	-	-	-	-	1	1	2	3
CO 2	1	1	2	-	2	-	-	-	-	-	-	1	1	2	3
CO 3	1	1	2	1	2	-	-	-	-	-	-	1	1	2	3
Average	1	0.6	2	0.3	2	-	-	-	-	-	-	1	1	2	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Setting Up a Version Control System

- Objective: Configure and use Git for version control.
- Install Git on your system.
- Create a new Git repository.
- Add files, commit changes, and create branches.
- Push changes to a remote repository.

Experiment 2: CI/CD Pipeline Setup

- Objective: Build a basic CI/CD pipeline.
- Set up a CI server (e.g., Jenkins).
- Configure a simple pipeline that builds and deploys a sample application.

Experiment 3: **Security Scanning in CI**

- Objective: Implement security checks in the CI pipeline.
- Integrate a static code analysis tool (e.g., SonarQube).
- Configure the pipeline to perform security scans on code commits.

Experiment 4: Infrastructure as Code (IaC)

- Objective: Create and manage infrastructure using IaC.
- Choose an IaC tool (e.g., Terraform).

- Define infrastructure components as code.
- Provision and manage resources on a cloud provider (e.g., AWS).

Experiment 5: Securing IaC

- Objective: Apply security best practices to IaC templates.
- Implement security controls in IaC scripts.
- Scan IaC templates for security vulnerabilities.

Experiment 6: Automated Vulnerability Scanning

- Objective: Set up automated vulnerability scanning.
- Configure a container scanning tool (e.g., Clair) in the CI/CD pipeline.
- Scan Docker images for vulnerabilities.

Experiment 7: Security Testing and Automation

- Objective: Automate security testing of applications.
- TIntegrate a dynamic application security testing (DAST) tool (e.g., OWASP ZAP) into the pipeline.
- Perform automated security testing on a web application.

Experiment 8: Security Incident Response Simulation

- Objective: Simulate a security incident and practice response.
- Create a security incident scenario (e.g., a data breach).
- Develop an incident response plan.
- Simulate and respond to the incident.

Experiment 9: Compliance as Code

- Objective: Implement compliance checks in the pipeline.
- Choose a compliance checking tool (e.g., InSpec).
- Write compliance checks as code.
- Integrate compliance checks into the CI/CD pipeline.

Experiment 10: DevSecOps Culture and Collaboration

- Objective: Promote collaboration between development, operations, and security teams.

- Collaborative incident response exercise involving all teams.
- Identify areas for improved communication and cooperation.

Total Lab hours 30

Textbooks

1. Cloud Application Development Lab Manual (IBM ICE Publications)
2. Expert AWS Development by Atul V. Mistry, Packt Publishing, March 2018

Reference Books

1. Jenkins: The Definitive Guide: Continuous Integration for the Masses by John Smart, 1st Edition, O'Reilly Media, 2011

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSDV3019P	**Container Orchestration and Security**	4	0	0	4
Total Units to be Covered: 05	**Total Contact Hours: 60**				
Prerequisite(s):	DevOps Fundamentals and SCM, Basic Understanding of Cloud Computing	**Syllabus version: 1.0**			

Course Objectives

The Container Orchestration with Kubernetes and Security course will help the students to grasp the key skills, technology, and concepts that a Kubernetes administrator needs to know. Plan to oversee containerized workloads and administrations with industrial Organizations utilizing the Kubernetes Preparing course. It covers all the aspects, including application lifecycle management, installation, configuration and validation, networking, scheduling, security, cluster maintenance, core concepts, Azure Kubernetes, storage, and more.

Course Outcomes

On completion of this course, the students will be able to

CO1: Understand the significance of container over virtual machines.

CO2: Design the container based deployment of an application on container based platform.

CO3: Implement a Kubernetes Cluster and deploy a Dockerized application on a clouds based Kubernetes cluster.

CO4: Test and Integrate security protocols in containersed application for any vulnerability.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	2	-	-	3	-	-	-	1	-	-	-	-	-	-
CO 2	-	3	-	-	2		3	-	-	-	-	-	-	-	-
CO 3	-	2	-	-	1	-	-	-	-	2	-	-	-	-	-
CO4	-	-	1	-	2	-	-	-	-	-	-	-	-	-	-
CO5	-	-	2	-	2	-	-	-	2	-	-	2	-	-	-

Average	0.6	1.4	0.6	-	2	-	0.6	-	0.6	0.4	-	0.4	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Application Containerization **8 Lecture Hours**

Understanding Containers, Challenges in the Software Industry, Problems in Software Industry Before Containers, Put that in Container! Solution by containers in the Software Industry, Virtualization, Hypervisor, Scope of Virtualization, Containers vs Virtual Machines, Understanding Containers, Containerisation Platform, Runtime and Images, Container Platform, Container Runtime, The Chroot System, FreeBSD Jails, Linux Containers (LXC), Docker

Unit II: Introduction to Containerization and different environments **10 Lecture Hours**

Docker architecture, Docker Daemon (Container Platform), Docker Rest API, Understanding Different environments: (Dev, QA and Prod), Overcoming issues with different environments, Development Environment, Testing Environment, Staging Environment, Production Environment, Virtual machines for dev/deployments, Containers for dev/deployments, Advantages and drawbacks of containerization

Unit III: Docker Fundamentals and Internals **8 Lecture Hours**

Docker container states, docker image vs docker containers, docker image creation using docker commit & Dockerfile, Dockerfile important keywords, docker tags, persistent storage use-case, docker volumes, docker networks, creating custom networks in docker, docker registry, docker inbuilt security concepts (namespaces, cgroups)

Unit IV: Orchestration Tools **12 Lecture Hours**

What is orchestration, Need of orchestration, Case study: Need of Orchestration , Need of Orchestration: Container and Microservices, Docker Swarm and Kubernetes, Architecture, AWS (ECS,EKS), AWS Elastic Container Services Architecture, Azure

Kubernetes Services, OpenShift, KUBERNETES ON CLOUD, Monitoring of container, How to monitor

Unit V: Container Security **7 Lecture Hours**

Docker vs Vagrant, Docker Challenges Revisited, Vulnerabilities in images (Public and Private), Denial of service attacks, Privilege escalation methods in Docker, Security misconfigurations, Container Security, Content Trust and Integrity checks, Capabilities and namespaces in Docker, Segregating Networks, Kernel Hardening using SecComp and AppArmor, Static Analysis of container(Docker) images, Dynamic Analysis of container hosts and daemons.

Total lecture Hours 45

Textbooks

1. Liz Rice,, "Container Security Fundamental Technology Concepts that Protect Containerized Applications, O'Reilly Media, 2020.
2. José Manuel Ortega, "CandelDevOps and Containers Security and Monitoring in Docker Containers", O'Reilly, 2020.
3. Gigi Sayfan, "Mastering Kubernetes", 4th Edition, Packt Publishing, 2020.

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSDV3119P	Container Orchestration and Security Lab	0	0	2	1
Total Units to be Covered: 14	Total Contact Hours: 30				
Prerequisite(s):	DevOps Fundamentals and SCM, Basic Understanding of Cloud Computing	Syllabus version: 1.0			

Course Objectives

The student should be able to understand and apply concepts of horizontal scaling techniques and differentiate it from vertical scaling techniques.

Course Outcomes

On completion of this course, the students will be able to

CO1. Develop and Compare containers with virtual machines.

CO2. Build Docker Images for Docker container.

CO3. Implement Docker Swarm using Orchestration Tools for a large production environment.

CO4. Setup and Use Kubernets Single and Multi-Node cluster and deploy application in different Pods.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	1	3	2	2	-	-	-	2	-	-	1	1	1	3
CO 2	1	2	2	2	3	-	-	-	3	-	-	-	2	2	3
CO 3	2	3	3	3	3	2	-	-	3	-	-	-	2	2	3
CO 4	2	3	3	3	3	2	-	-	3	-	-	-	2	2	3
Average	1.5	2.25	2.75	2.5	2.75	2	-	-	2.75	-	-	1	1.75	1.75	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

List of Experiments

Experiment1 Installing Vagrant & Creating basic vagrant box using VirtualBox virtualization.

Experiment2 Understanding vagrant file - Configuration - CPU, RAM, Storage, Provisioning (Shell Script).

Experiment3 Docker Machine - Installation , configuration, creating machines (on VirtualBox).

Experiment4 Docker - Installation, Configuration, Running Images.

Experiment5 Dockerfile - Containerizing application, Building Images, Tagging, Publishing.

Experiment6 Docker - Volumes, Env, Monitoring (Docker stats).

Experiment7 DTR - Docker Hub, Private Registries, Publishing images.

Experiment8 Docker Compose - Installation, Creating Compose files, Running Images using docker-compose.

Experiment9 Running Multi-Container applications using docker compose and on Swarm.

Experiment10 Kubernetes -Minikube installation and fundamentals.

Experiment11 Deploying Pods and Services on minikube.

Experiment12 Build Docker Image using .Dockerignore file.

Experiment13 Prepare and Implement Docker Container Restart Policy

Experiment14 Working with Metadata, Log File using Docker

Total Lab hours 30

Textbooks

1. Application containerization by Xebia. Available at https://xebia.com/blog/what-are-containers/

Reference Books

1. Jarosław Krochmalski, “Developing with Docker”, Packt Publishing, 2016.
2. Adrian Mouat, “Orchestrating, clustering, and managing containers”, O'Reilly Media, Inc., 2016.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSDV4009P	CICD Pipeline and Security	4	0	0	4
Total Units to be Covered: 06	Total Contact Hours: 60				
Prerequisite(s):	Basic Programming Skills (Linux commands) and Version Control System	Syllabus version: 1.0			

Course Objectives

The course objective of "CICD Pipeline and Security" is to provide students with a comprehensive understanding of Continuous Integration and Continuous Deployment (CICD) principles, practices, and security considerations. The course aims to equip students with the knowledge and skills to design, configure, and implement efficient and reliable CICD pipelines. It focuses on integrating security practices into the CICD process, including automated security testing, vulnerability scanning, code analysis, and secure deployment practices. The course also covers DevSecOps principles, emphasizing collaboration and the integration of security throughout the software development and operations lifecycle. Students will learn about various tools and technologies used in CICD pipelines, as well as deployment strategies, automated testing, monitoring techniques, and industry best practices. The ultimate goal is to enable students to build secure and efficient CICD pipelines that ensure the continuous delivery of high-quality software.

Course Outcomes

On completion of this course, the students will be able to

CO1. Explore the CICD toolchain, including version control, build automation, and containerization by understanding the core concepts and benefits of CICD pipelines.

CO2. Integrate security practices into CICD pipelines, including automated security testing and vulnerability scanning.

CO3. Design and implement scalable and reliable CICD workflows.

CO4. Apply DevSecOps principles to foster collaboration and security throughout the software development process.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	3	-	-	-	-	-	3	3	-	2	-
CO 2	-	-	-	-	3	-	-	-	-	-	2	3	-	2	-
CO 3	-	-	-	-	3	-	-	-	-	-	2	3	-	2	-
CO4	-	-	-	-	3	-	-	-	-	-	2	3	-	2	-
Average	-	-	-	-	3	-	-	-	-	-	2.25	3	-	2	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to CICD Pipelines and Security **10 Lecture Hours**

Overview of CICD principles and benefits, Introduction to security considerations in CICD pipelines, Understanding the software development lifecycle and the role of CICD

Unit II: Version Control and Source Code Management **10 Lecture Hours**

Introduction to version control systems (e.g., Git, SVN), Branching strategies and best practices, Integrating version control into CICD pipelines

Unit III: Security Integration in CICD Pipelines **10 Lecture Hours**

Identifying security vulnerabilities in software development, Secure coding practices and code analysis tools, Integrating security practices into CICD pipelines

Unit IV: Automated Testing and Quality Assurance **10 Lecture Hours**

Different types of automated testing (e.g., unit testing, integration testing), Implementing test automation in CICD pipelines, Continuous quality assurance and code quality monitoring

Unit V: Deployment Strategies and Orchestration **10 Lecture Hours**

Overview of deployment strategies (e.g., blue-green, canary, rolling deployments), Infrastructure as Code (IaC) and configuration management, Deployment orchestration and containerization (e.g., Docker, Kubernetes)

Unit VI: **10 Lecture Hours**

Security controls and practices in CICD pipelines, Vulnerability scanning and management, Compliance monitoring and reporting in CICD pipelines.

Total lecture Hours 60

Textbooks

1. Jez Humble and David Farley, "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation", Addison-Wesley, 2010.
2. Julien Vehent, "Secure DevOps: A Practical Guide to Securing Your Software Delivery Pipeline", Manning, 2018.
3. Heather Adkins, Betsy Beyer, Paul Blankinship, and Piotr Lewandowski, “Building Secure and Reliable Systems: Best Practices for Designing, Implementing, and Maintaining Systems", O′Reilly, 2020.

Reference Books

1. Gene Kim, Jez Humble, Patrick Debois, and John Willis, "DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations", It Revolution Press, 2016.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSDV4109P	**CICD Pipeline and Security Lab**		0	0	2	1
Total Units to be Covered: 12		**Total Contact Hours: 30**				
Prerequisite(s):	Basic Programming Skills (Linux commands) and Version Control System	**Syllabus version: 1.0**				

Course Objectives

The course objective of "CICD Pipeline and Security" is to provide students with a comprehensive understanding of Continuous Integration and Continuous Deployment (CICD) principles, practices, and security considerations. The course aims to equip students with the knowledge and skills to design, configure, and implement efficient and reliable CICD pipelines. It focuses on integrating security practices into the CICD process, including automated security testing, vulnerability scanning, code analysis, and secure deployment practices. The course also covers DevSecOps principles, emphasizing collaboration and the integration of security throughout the software development and operations lifecycle. Students will learn about various tools and technologies used in CICD pipelines, as well as deployment strategies, automated testing, monitoring techniques, and industry best practices. The ultimate goal is to enable students to build secure and efficient CICD pipelines that ensure the continuous delivery of high-quality software.

Course Outcomes

At the end of this course student should be able to learn:

CO.1. Interpret advantages of using continuous integration and continuous development in Agile.

CO.2. Explain anatomy of continuous delivery pipeline to automate the testing within minimum constraints.

CO.3. Outline continuous integration by using various tools for continuous integration and automation.

CO.4 Understand static code analysis like data flow analysis, taint analysis, lexical analysis.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	3	-	-	-	-	-	3	3	-	2	-
CO 2	-	-	-	-	3	-	-	-	-	-	2	3	-	2	-
CO 3	-	-	-	-	3	-	-	-	-	-	2	3	-	2	-
CO4	-	-	-	-	3	-	-	-	-	-	2	3	-	2	-
Average	-	-	-	-	3	-	-	-	-	-	2.25	3	-	2	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment No 1 Installation of Jenkins and Execution of a simple Job in Jenkins

Experiment No 2 Jenkins Integration with GitHub

Experiment No 3 Jenkins Integration with GitHub and Maven

Experiment No 4 Static Code Analysis using SonarQube

Experiment No 5 Jenkins Integration with Sonarqube

Experiment No 6 Create Pipeline using Jenkinsfile

Experiment No 7 Create Pipeline using Blue Ocean Plugin

Experiment No 8 Implementing Master/Slave Architecture in Jenkins

Experiment No 9 Uploading Artifacts on Nexus Server using Command Line

Experiment No 10 Nexus Integration with Jenkins

Experiment No 11 Integration of Docker with Jenkins to generate an image of generated build

Experiment No 12 Deployment of Docker Image on Cloud/ Local server (Nexus) using Jenkins

Total Lab hours 30

Textbooks

1. Jez Humble and David Farley, "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation", Addison-Wesley, 2010.
2. Julien Vehent, "Secure DevOps: A Practical Guide to Securing Your Software Delivery Pipeline", Manning, 2018.

3. Heather Adkins, Betsy Beyer, Paul Blankinship, and Piotr Lewandowski, “Building Secure and Reliable Systems: Best Practices for Designing, Implementing, and Maintaining Systems", O′Reilly, 2020.

Reference Books

1. Gene Kim, Jez Humble, Patrick Debois, and John Willis, "DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations", It Revolution Press, 2016.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name		L	T	P	C
CSDV4010P	System Provisioning and Monitoring		3	0	0	3
Total Units to be Covered: 09		Total Contact Hours: 45				
Prerequisite(s):	Cloud Computing, Container Orchestration	Syllabus version: 1.0				

Course Objectives

The course on System Provisioning and Monitoring aims to provide students with a comprehensive understanding of system provisioning methodologies, monitoring fundamentals, and relevant tools and technologies. The objective is to equip students with the knowledge and skills to automate system provisioning processes, design effective monitoring architectures, optimize system performance, and ensure security monitoring. By the end of the course, students should be able to proficiently provision systems using infrastructure-as-code techniques, implement scalable and reliable monitoring systems, identify and resolve performance bottlenecks, and incorporate security monitoring into their overall monitoring strategy. Additionally, they should be aware of emerging trends in system provisioning and monitoring, enabling them to adapt to evolving technologies and practices in the field.

Course Outcomes

On completion of this course, the students will be able to

CO1. Equipped with the knowledge and skills to design and implement robust monitoring architectures.

CO2. Effectively provision systems using automated methodologies and tools. They will understand infrastructure-as-code principles and be capable of provisioning physical and virtual servers, configuring networks, and deploying resources in a reliable and efficient manner.

CO3. Ability to identify and address performance bottlenecks in IT systems. They will understand techniques for capacity planning, resource allocation, load balancing, and optimization to improve system performance and ensure efficient resource utilization.

CO4. Integrate security monitoring practices into the overall monitoring strategy. They will understand the importance of monitoring security events, performing intrusion detection, conducting vulnerability assessments, and responding to security incidents to ensure the protection and integrity of IT systems.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	3	-	-	-	-	-	3	3	-	2	-
CO 2	-	-	-	-	3	-	-	-	-	-	2	3	-	2	-
CO 3	-	-	-	-	3	-	-	-	-	-	2	3	-	2	-
CO4	-	-	-	-	3	-	-	-	-	-	2	3	-	2	-
Average	-	-	-	-	3	-	-	-	-	-	2.25	3	-	2	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to System Provisioning and Monitoring **6 Lecture Hours**

Overview of system provisioning and monitoring concepts, Importance of system provisioning and monitoring in IT infrastructure management, System provisioning methodologies and best practices ,Automated provisioning tools and techniques, Infrastructure as Code (IaC) and configuration management, Provisioning physical and virtual servers, Network provisioning and configuration.

Unit II: Monitoring Fundamentals **6 Lecture Hours**

Fundamentals of system monitoring, Types of monitoring: active monitoring, passive monitoring, and synthetic monitoring, Metrics, logs, and events: collection, analysis, and interpretation, Key performance indicators (KPIs) and service level agreements (SLAs).

Unit III: Monitoring Tools and Technologies **6 Lecture Hours**

Monitoring tool categories: agent-based, agentless, and hybrid, Open-source and commercial monitoring tools, Infrastructure monitoring: server monitoring, network monitoring, and storage monitoring, Application monitoring: performance monitoring, availability monitoring, and error monitoring, Cloud monitoring: monitoring cloud resources and services.

Unit IV: Monitoring Architecture and Design **6 Lecture Hours**

Monitoring architecture components: agents, collectors, dashboards, and data repositories, Scalability and high availability considerations, Data visualization and reporting, Alerting and notification mechanisms, Integrating monitoring systems with incident management and ticketing systems.

Unit V: Performance Tuning and Optimization **4 Lecture Hours**

Identifying performance bottlenecks and issues, Performance metrics and benchmarks, Capacity planning and resource allocation, Load balancing and optimization techniques.

Unit VI: Security Monitoring **4 Lecture Hours**

Security event monitoring and log analysis ,Intrusion detection and prevention systems (IDPS), Vulnerability scanning and assessment, Security incident response and forensic analysis.

Unit VII: Application Performance Monitoring (APM) **4 Lecture Hours**

End-user experience monitoring, Application code profiling and performance optimization, Transaction tracing and request monitoring, Database performance monitoring.

Unit VIII: Cloud Monitoring and DevOps **4 Lecture Hours**

Monitoring cloud infrastructure and services, Continuous monitoring in DevOps practices, Infrastructure automation and monitoring as code, Containerization and container monitoring.

Unit IX: Emerging Trends and Future Directions **5 Lecture Hours**

Machine learning and artificial intelligence in system monitoring, Internet of Things (IoT) and monitoring challenges, Serverless computing and monitoring considerations, Proactive monitoring and predictive analytics.

Total lecture Hours 45

Textbooks

1. Nuno Ferreira, "Provisioning, Deployment, and Operation of High-Density Cloud Services",

2. Slawek Ligus, "Effective Monitoring and Alerting: For Web Operations", O′Reilly, 2012.
3. Harjot Gill, "Cloud Monitoring and Operations",
4. John Allspaw, "The Art of Capacity Planning: Scaling Web Resources",

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSDV4110P	System Provisioning and Monitoring Lab	0	0	2	1
Total Units to be Covered: 10	Total Contact Hours: 30				
Prerequisite(s):	Elementary knowledge of DevOps	Syllabus version: 1.0			

Course Objectives

To provide students with hands-on experience in provisioning systems, designing effective monitoring solutions, optimizing system performance, and incorporating security monitoring into their strategies. The labs are designed to align with the course objectives and equip students with practical skills relevant to system provisioning and monitoring in real-world scenarios.

Course Outcomes

On completion of this course, the students will be able to

CO1: Identify critical system metrics, develop monitoring plans, and create dashboards and alerts to ensure the reliable and proactive monitoring of systems.

CO2: Design effective monitoring architectures by identifying critical system metrics, developing monitoring plans, and creating dashboards to ensure the reliable and proactive monitoring of systems.

CO3: Analyze system performance metrics, identify performance bottlenecks, and implement optimizations to improve system performance.

CO4: Apply security monitoring into their overall monitoring strategies.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	3	-	-	-	1	-	-	-	-	-	-
CO 2	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO 3	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-
CO4	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO5	-	-	2	-	2	-	-	-	2	-	-	2	-	-	-

Average	**-**	**-**	0.4	**-**	2	**-**	**-**	**-**	0.6	**-**	**-**	0.4	**-**	**-**	**-**

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment : Setting Up the Development Environment (Java, Spring Boot, IDE, Docker, etc.)

Experiment 1: Infrastructure as Code (IaC) Basics

- Objective: Introduce students to IaC concepts.
- Tasks:
    - Choose an IaC tool (e.g., Terraform).
    - Write a simple IaC script to provision a virtual machine on a cloud platform.

Experiment 2: Advanced IaC Techniques

- Objective: Explore advanced IaC capabilities.
- Tasks:
    - Provision multiple virtual machines with interdependencies.
    - Use IaC to configure network settings and security groups.

Experiment 3: Continuous Integration for IaC

- Objective: Implement CI for IaC scripts.
- Tasks:
    - Set up a CI/CD pipeline (e.g., Jenkins) for IaC.

- Automate IaC testing and deployment.

Experiment 4: Designing Effective Monitoring

- Objective: Teach principles of effective monitoring design.
- Tasks:
    - Identify key metrics and performance indicators for a system.
    - Develop a monitoring plan for a sample application.

Experiment 5: Implementing Monitoring Solutions

- Objective: Set up monitoring tools and services.
- Tasks:
    - Install and configure monitoring agents (e.g., Prometheus, Grafana).
    - Create dashboards and alerts for a monitored system.

Experiment 6: Performance Optimization

- Objective: Identify and address performance bottlenecks.
- Tasks:
    - Analyze system performance metrics to detect bottlenecks.
    - Implement optimizations to improve system performance.

Experiment 7: ScaLab Exercisele Monitoring Architectures

- Objective: Design scalable monitoring systems.
- Tasks:
    - Explore techniques for scaling monitoring infrastructure.
    - Set up distributed monitoring and data aggregation.

Experiment 8: Security Monitoring

- Objective: Integrate security monitoring into the overall strategy.

- Tasks:
    - Configure security monitoring tools (e.g., intrusion detection systems).
    - Create alerts for security-related events and incidents.

Experiment 9: Incident Response Simulation

- Objective: Simulate a security incident and practice incident response.
- Tasks:
    - Develop an incident response plan for a security scenario.
    - Execute the plan and analyze the effectiveness of the response.

Experiment 10: Emerging Trends and Future Technologies

- Objective: Explore emerging technologies and trends in system provisioning and monitoring.
- Tasks:
    - Research and present on a current trend or technology (e.g., containerization, serverless computing) in the field.

Total Lab hours 30

Textbooks

1. "The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win" by Gene Kim, Kevin Behr, and George Spafford, 5th Edition, Revolution Press.
2. "GIT for Teams" by Emma Jane, O'Reilly.

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

3. Cloud Computing And Virtualization Technology Track

Course Code	Course name	L	T	P	C
CSVT2010P	Cloud Computing Fundamentals	4	0	0	4
Total Units to be Covered: 06	Total Contact Hours: 60				
Prerequisite(s):	Computer Organization and Architecture, Operating System,	Syllabus version: 1.0			

Course Objectives

1. To develop a strong foundation in virtualization and cloud computing, including their key concepts, principles, benefits, and distinguishing characteristics.
2. To critically evaluate the role of Hypervisors in virtualization and compare and contrast different types of virtualization, such as server, storage, and network virtualization, highlighting their unique features and use cases.
3. To analyze the diverse deployment and delivery models in cloud computing, assessing their advantages, disadvantages, and suitability for various scenarios, enabling informed decision-making for cloud adoption.
4. To assess and analyze different types of cloud workloads, considering their specific characteristics and suitability for cloud environments.

Course Outcomes

CO1. Understand the fundamental concepts and principles of virtualization and cloud computing, including their benefits and characteristics.
CO2. Evaluate the role of Hypervisors in virtualization and compare different types of virtualization, such as server, storage, and network virtualization.
CO3. Analyze the various deployment and delivery models in cloud computing and assess their advantages, disadvantages, and suitability for different scenarios.
CO4. Assess and analyze different types of cloud workloads considering their characteristics and suitability for cloud environments.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	-	-	1	-	-	-	-	-	-	1	1	-	3
CO 2	1	-	-	-	1	-	-	-	-	-	-	1	1	-	3
CO 3	1	-	-	-	1	-	-	-	-	-	-	1	1	-	3
CO4	1	-	-	-	1	-	-	-	-	-	-	1	1	-	3
Average	1	-	-	-	1	-	-	-	-	-	-	1	1	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to Virtualization **10 Lecture Hours**

Traditional IT Infrastructures, Shortcomings of physical infrastructures, Benefits and characteristics of Virtualization, Comparison of traditional IT infrastructures with virtualized infrastructures, Implementing Virtualization, Triggers for virtualization, Preparation for virtualization: Server selection, Server sizing, Server criticality, Provisioning, Proximity and Locality, Transition tools for virtualization, Cost Savings, A typical hardware/software server stack, Logical Equivalence, Pre and Post Virtualization Server Stacks, Types of Virtualization: Area and technology based classification, History of Virtualization, Time-sharing systems, IBM Mainframe Virtualization, IBM PowerVM Virtualization, Extending Virtualization to x86, Hardware support for x86 Virtualization, Impact of Virtualization: Cost Impact, Manageability Impact (CAPEX and OPEX).

Unit II: Server and Storage Virtualization **10 Lecture Hours**

Types of Server Virtualization: Process Level and system Level, Process Level: Emulation, High-Level VM, Multiprogramming, System Level: Hardware assisted Virtualization, Full Virtualization, Para Virtualization, Partial virtualization, Hybrid Virtualization; Simulation

Hypervisors: Machine Reference model, Ring levels on x86 processors, Types of Hypervisors: Type 1 and Type II, Hypervisor modules, Goldberg and Popek Criteria's

Other Types of Virtualization: Storage Virtualization, its benefits and types: Host level storage virtualization, Host based mirroring, Storage level virtualization, Network based storage Virtualization; Desktop Virtualization: its working, benefits, constraints and types, Application Server Virtualization

Case Studies**:** Xen Hypervisor, Vmware and Microsoft Hyper-V

Unit III: Network and Application Virtualization 8 Lecture Hours

Network Virtualization: Network virtualization overview, Virtual Private Network (VPN): VPN working, VPN types; Virtual LAN (VLAN), Advantages of VLAN

Operating System Level Virtualization: Programming language-level virtualization, Application-level virtualization. Application virtualization overview: Challenges in using applications in traditional install, use and update model, Solution for challenges, Architecture, Benefits of application virtualization. Containerization: application Containerization, Benefits, types, virtualization vs containerization

Unit IV: Introduction to Cloud Computing 12 Lecture Hours

Distributed vs Parallel computing, Elements of parallel and distributed computing, Approaches to parallel computing, Levels of parallelism.

Cloud Computing: Virtualization and Cloud, overlapping of virtualization and cloud, Areas and relative savings, Cloud definitions, Vision of cloud computing, Cloud computing value: Business and technological value, Market view point, Changes for provider, Cloud and end user, Advantages and Disadvantages of cloud computing model. Distributed computing: Client server, Peer to Peer, Grid vs cloud, Cluster vs cloud.

Central ideas to cloud computing: Web 2.0, Utility Computing, Service Oriented Architecture, Service level Agreements Characteristics of Cloud Computing: Scalability and Elasticity, Availability and Reliability, Manageability and Interoperability, Performance & Optimization, Accessibility and Portability, Cloud Open challenges. Anatomy of a cloud: Cloud components, Cloud computing solution components, Service Catalog, User self-service portal, Service request management, Provisioning, Optimized infrastructure, Chargeback

Unit V: **Cloud Computing Service and Deployment Models 12 Lecture Hours**

Cloud Service Models**:** Infrastructure as a service (IaaS), Platform as a service (PaaS), Software as a service (SaaS), Pure IaaS, Pure PaaS and Pure SaaS, IaaS: IaaS features, Examples, Layers: Physical Infrastructure Layer, Infrastructure Management Software, Web Based Management Interface, Cases where IaaS suitable or not suitable,PaaS: PaaS features, Examples, Layers: Physical Infrastructure Layer, PaaS Core Middleware, Web Based Management Interface, Cases where PaaS suitable or not suitable, SaaS: SaaS features, Examples, Cases where IaaS suitable or not suitable

Case Studies**:** AWS EC2, Salesforce and Social Media

Cloud Deployment Models: Private cloud, Public cloud, Hybrid cloud, Community Cloud, Pros and cons of each deployment model, Cloud deployment decision factors, Time to deploy,

Public cloud: Examples, Important points about public cloud, Factor Matrix, Public cloud advantages, Public cloud disadvantages, Private cloud – Scenario, Key observations from scenario, Factor matrix, Private cloud advantages, Private cloud disadvantages, Hybrid cloud scenarios-Observations from Hybrid scenario, Factor Matrix, Hybrid cloud advantages, Hybrid cloud disadvantages

Multi cloud: Value and benefits of multicloud, Multicloud management

Unit VI: **Cloud Workloads 8 Lecture Hours**

Cloud Workloads Overview, Workload characterization, Factors that influence cloud workload, Workloads analysis hierarchy for cloud suitability considering deployment models, Types of workloads: Scientific data intensive workloads, Business and Consumer Applications, Productivity, Social Networking, Media application, Multiplayer Online Gaming.

Total lecture Hours 60

Textbooks

1. Buyya Vecchiola, and Selvi, “Mastering Cloud Computing”, First Edition, McGraw Hill Education, 2017.

Reference Books

1. Rajkumar Buyya, James Broberg, and Andrzej M. Goscinski, “Cloud Computing: Principles and Paradigms”, Wiley, 2013.
2. Kailash Jayaswal, Jagannath Kallakurchi, Donald J. Houde, Dr. Deven Shah, and Kogent Learning Solutions, “Cloud Computing Black Book”, Dreamtech Press, 2014.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSVT2109P	Cloud Computing Fundamentals Lab	0	0	2	1
Total Units to be Covered: 12	Total Contact Hours: 30				
Prerequisite(s):	Computer System Architecture, Operating System	Syllabus version: 1.0			

Course Objectives

1. To equip students with the practical skills necessary for creating and managing virtual machines (VMs) across diverse virtualization platforms, enabling effective infrastructure deployment.
2. To empower students to apply their knowledge of virtualization and cloud computing in real-world scenarios, demonstrating their ability to leverage cloud technologies for developing applications on public cloud platforms
3. To foster proficiency in implementing server and storage virtualization on public cloud platforms, enabling students to optimize resource utilization and drive scalable infrastructure solutions through analysis and evaluation.

Course Outcomes

CO1: Demonstrate proficiency in creating and managing virtual machines (VMs) using various virtualization platforms.
CO2: Apply their knowledge of virtualization and cloud computing to develop an application using public cloud, showcasing their ability to leverage cloud technologies.
CO3: Exhibit proficiency in implementing server and storage virtualization on public cloud platforms.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	-	-	2	-	-	-	-	-	-	-	-	-	3
CO 2	1	-	-	-	2	-	-	-	-	-	-	-	-	-	3
CO 3	1	-	-	-	2	-	-	-	-	-	-	-	-	-	3
Average	1	-	-	-	2	-	-	-	-	-	-	-	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment No. 1: Install and configure hosted hypervisor (Type II): Oracle VM VirtualBox

Experiment No. 2: Create a Window 7 and Ubuntu VM over Oracle VM VirtualBox

Experiment No. 3: Install and configure bare metal hypervisor (Type I): VMware ESXI

Experiment No. 4: Creating a virtual machine using QEMU/KVM on Ubuntu.

Experiment No. 5: Creating a virtual machine using QEMU/KVM on CentOS.

Experiment No. 6: A brief introduction to AWS.

Experiment No. 7: Introduction to AWS EC2 instance and create an EC2 instance.

Experiment No. 8: Attaching an EBS Volume to EC2 instance.

Experiment No. 9: Create a S3 bucket and upload an object in it.

Experiment No. 10: Launch Linux EC2 instance through AWS CLI

Experiment No. 11: Connect S3 bucket with an EC2 instance.

Experiment No. 12: Run a Serverless "Hello, World!" with AWS Lambda.

Total Lab hours 30

Textbooks

1. Mark Wilkins, "Learning Amazon Web Services (AWS): A Hands-On Guide to the Fundamentals of AWS Cloud", First Edition, Pearson Education, 2019.
2. Buyya, Vecchiola, Selvi, "Mastering Cloud Computing", First Edition, McGraw Hill Education, 2017.

Reference Books

1. Rajkumar Buyya, James Broberg, and Andrzej M. Goscinski, "Cloud Computing: Principles and Paradigms", Wiley, 2013.

2. Kailash Jayaswal, Jagannath Kallakurchi, Donald J. Houde, Dr. Deven Shah, and Kogent Learning Solutions, “Cloud Computing Black Book”, Dreamtech Press, 2014.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

<table>
<tr><th>Course Code</th><th colspan="2">Course name</th><th>L</th><th>T</th><th>P</th><th>C</th></tr>
<tr><td>CSVT3029P</td><td colspan="2">Cloud Computing Architecture and Deployment Models</td><td>4</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td colspan="2">Total Units to be Covered: 05</td><td colspan="5">Total Contact Hours: 60</td></tr>
<tr><td>Prerequisite(s):</td><td colspan="2">Cloud Computing Fundamentals</td><td colspan="4">Syllabus version: 1.0</td></tr>
</table>

Course Objectives

1. To understand the characteristics, benefits, and limitations of cloud service models (IaaS, PaaS, SaaS) and evaluate their suitability for different organizational needs.
2. To examine the deployment models (public, private, hybrid) of cloud computing and analyze their advantages, limitations, and considerations for effective cloud infrastructure implementation.
3. To explore cloud computing reference architectures NIST, IBM & AWS and gain a comprehensive understanding of their objectives, components, and control mechanisms for designing robust cloud solutions.
4. To analyze and evaluate fundamental and advanced cloud architectures, including workload distribution, scalability, resource pooling, and redundancy, to design optimized cloud environments.

Course Outcomes

On completion of this course, the students will be able to

CO1: Analyze and compare the key characteristics, benefits, and limitations of IaaS, PaaS, and SaaS cloud service models to effectively assess their suitability for different organizational needs.

CO2: Analyze and assess the cloud deployment models (Public, Private, Hybrid) demonstrating a deep understanding of their advantages, limitations, and challenges to make informed decisions regarding the selection and management.

CO3: Analyze and evaluate NIST, IBM's and AWS cloud computing reference architectures to make informed decisions about their adoption and implementation in diverse usage scenarios.

CO4: Ability to apply fundamental as well as advanced cloud computing architectures in various business scenarios.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	1	-	1	2	-	-	-	-	-	-	-	1	1	3
CO 2	1	1	-	1	2	-	-	-	-	-	-	-	1	1	3
CO 3	1	1	-	1	2	-	-	-	-	-	-	-	1	1	3
CO4	1	1	-	1	2	-	-	-	-	-	-	-	1	1	3
Average	1	1	-	1	2	-	-	-	-	-	-	-	1	1	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Service Models (IaaS, PaaS and SaaS) **13 Lecture Hours**

Infrastructure as a Service (IaaS): Characteristics of IaaS, Comparing ISPs and IaaS, IaaS case studies, IaaS enabling technology, the trusted cloud, IaaS as the best/not best option, PaaS: Platform as a Service.
Platform as a Service: PaaS characteristics, Integrated lifecycle platforms, Anchored lifecycle platforms, Enabling technologies as a platform, Case studies: Integrated lifecycle platform, PaaS as the best/ not best option.
Software as a Service (SaaS): SaaS origin, Evolution of SaaS: Salesforce.com's approach, Characteristics of Software as a Service (SaaS), SaaS economics and the ecosystem, Types of SaaS platforms, SaaS: Providers, Collaboration as a service, Enabling and management tools as a service, Monitoring and management tools as a service, SaaS as the best/not best option.

Unit II: Deployment Models (Pubic, Private, Hybrid) **12 Lecture Hours**

Private Cloud Deployment: Private Cloud, Illustration of Private Cloud, Advantages of Private Cloud, Limitations of Private Cloud, Service Management, Journey into Private Cloud, Planning and Strategy, Standardization, Virtualization, Automation, Cloud, Case study – VMware vCloud, Case Study – IBM SmartCloud Entry, Private cloud.

Public Cloud Deployment:: Public Cloud, Illustration of Public Cloud, Why Public Cloud, Advantages of Public Cloud, Limitations of Public Cloud: Low degree of security and control, Lack of control on infrastructure, Configuration, Network latency and

accessibility concerns, Highest long term cost; Public v/s Private: Journey into Public Cloud, Revisit the idea of adopting public cloud: Cloud vendor selection, migrating to Cloud, Cloud vendor selection, SLA – Service Level Agreements, Credits/Compensation terms, Credit process, Disaster recovery plan, Exclusions, Security and Privacy, Periodic upgrade and maintenance, Data location and Jurisdiction, Pricing and Measurability, Interoperability and Lock-in, Exit process/Termination policies, Proven track record; Public cloud vendors and Case studies: AWS, Microsoft Azure, Google GCP
Hybrid Cloud deployment: Hybrid Cloud, Why Hybrid Cloud, Illustration of Hybrid Cloud, Advantages of Hybrid Cloud, Challenges of Hybrid Cloud, Develop and manage hybrid workloads, Developing applications for hybrid cloud, Develop applications using PaaS, Managing hybrid workloads, Journey into Hybrid Cloud

OpenStack: Introduction, OpenStack Architecture, IBM SoftLayer, IBM Bluemix - Benefits of IBM Bluemix, -More Bluemix features, -Bluemix architecture.

Unit III: Cloud Computing Reference Architectures (NIST & IBM) 10 Lecture Hours

NIST Cloud Computing Reference Architecture (CCRA): Objectives of NIST, The conceptual reference model, Example: Usage scenarios, Cloud consumer, Cloud provider, Cloud auditor, Cloud broker, Cloud carrier, Scope of control between provider and consumer, CCRA: Architectural components, Service orchestration, Cloud service management, Business support, Provisioning and configuration Portability and interoperability, Security, Privacy, Cloud taxonomy.

IBM's CCRA: IBM CCRA roles, Cloud service consumer, Cloud service provider, Cloud services, Infrastructure, Common Cloud Management Platform (CCMP), CCMP supports any level of virtualization, Business Support Services (BSS), Operational Support Services (OSS), Security, resilience, performance and consumption, Cloud service creator: Service development tools, IBM CCRA versions or CCRA evolution, Adoption patterns, Adoption pattern in CCRA 3.0, Examples of cloud services.

Unit IV: Fundamental and Advanced Cloud Architectures 15 Lecture Hours

Fundamental Cloud Architectures: Workload distribution architecture, Resource pooling architecture, Dynamic scalability architecture, Elastic resource capacity architecture, Service load balancing architecture, Cloud bursting architecture, Elastic disk provisioning architecture, Redundant storage architecture.

Advanced Cloud Architectures: Overview of the advanced cloud architecture, Hypervisor clustering architecture, Load balanced virtual server instances architecture, Non-disruptive service relocation architecture, Zero downtime architecture, Cloud balancing architecture, Resource reservation architecture, Dynamic failure detection and recovery, Bare-metal provisioning architecture, Rapid provisioning architecture, Components that can comprise the system, Automated administration pattern, Storage workload management architecture, Live VM migration.

Unit V: Cloud Computing Reference Architecture (CCRA) – AWS 10 Lecture Hours

What is amazon web services, Features of AWS, Web application hosting, Content and media serving, Large scale computing and huge data sets, Disaster recovery for local applications, Media sharing, Financial service grids, Time series data processing, Backup and restore to VMware cloud on AWS, Pilot light on VMware cloud on AWS, Microsoft share point on VMware cloud on AWS, Hybrid active directory trusted domain, Hybrid active directory stretched domain, Oracle RAC on VMware cloud on AWS, Batch processing, Advertisement serving, Asynchronous online gaming, Ecommerce website: Web frontend, Ecommerce website: Checkout service, Marketing and recommendations, Fault tolerance and high availability, File synchronization service, Amazon services, Amazon Simple Storage Service (S3), Amazon services developer tools, Amazon services developer tools, Amazon services security, identity and compliance, Amazon service applications.

Total lecture Hours 60

Textbooks

1. Cloud Computing Architecture (IBM ICE Publication)
2. Cloud Computing Deployment Model (IBM ICE Publication)
3. Cloud Computing For Dummies Judith Hurwitz, Robin Bloor, Marcia Kaufman, Fern Halper, For Dummies, Edition 1, Nov 2009.
4. Cloud Computing: Concepts, Technology & Architecture, Thomas Erl,Pearson Education India, Jan 2014
5. OpenStack Essentials, Dan Radez, PackIT publication, publications , 2nd Edition, Jan 2016

Reference Books

1. Rajkumar Buyya, James Broberg, and Andrzej M. Goscinski, “Cloud Computing: Principles and Paradigms”, Wiley, 2013.
2. Gautam Shroff, “Enterprise Cloud Computing Technology Architecture Applications”, Cambridge, 2010.
3. Kalyn Sehrt, “About Openstack: A Comprehensive Tutorial To Revolutionize Cloud Computing Solutions”, Independently published, March 27, 2023.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSVT3129P	**Cloud Computing Architecture and Deployment Models Lab**	0	0	2	1
Total Units to be Covered: 12		**Total Contact Hours: 30**			
Prerequisite(s):	Cloud Computing Fundamentals and Cloud Computing Fundamentals Lab	**Syllabus version: 1.0**			

Course Objectives

1. To develop a comprehensive understanding of cloud computing architecture and deployment models
2. To gain practical experience in creating and managing infrastructure on cloud platforms, configuring and developing platforms as a service, and utilizing software as a service (SaaS) solutions with the help of various tools and technologies.
3. To develop practical skills in OpenStack by successfully installing and deploying virtual machines, exploring the components of OpenStack and its components.

Course Outcomes

At the end of this course student should be able to

CO1: Demonstrate proficiency in creating and managing virtual machine instances on cloud platforms (AWS EC2, Azure, and GCP) and establishing secure connections to them using appropriate key pairs and protocols.

CO2: Demonstrate advanced proficiency in AWS cloud management techniques like storage management, server less computing, auto scaling, load balancing and identity and access management for enhanced security and efficient resource utilization.

CO3: Demonstrate proficient hands-on skills in OpenStack by installing, deploying, and managing virtual machines using Nova, Cinder, Glance, and Keystone components, and utilize the monitoring tool Nagios to monitor and assess cloud environment parameters effectively.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3

CO 1	1	-	-	1	2	-	-	1	1	-	-	1	-	-	3
CO 2	1	-	-	1	2	-	-	1	1	-	-	1	-	-	3
CO 3	1	-	-	1	2	-	-	1	1	-	-	1	-	-	3
Average	1	-	-	1	2	-	-	1	1	-	-	1	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment-1: AWS EC2 (IaaS)

1.1. Creation of Windows instance on AWS EC2 and connecting to it through RDP client and .pem key pair.

1.2 Creation of Linux Instance on AWS EC2 and connecting it to through .ppk keypair using PuTTY.

Experiment-2: Azure & GCP (IaaS)

2.1. Create Windows and Ubuntu machine Virtual Machine in Azure and connect to it.

2.2. Create a Windows Virtual machine instance in GCP and connect to it.

Experiment-3: AWS Autoscaling and Load Balancers (IaaS)

3.1. Autoscaling can be performed on EC2 instances (Scale manually & Scale on demand)

3.2. Use Application Load Balancer in AWS

Experiment-4: AWS Storage (IaaS)

4.1. EBS Volume creation and attaching it to EC2 instance (Windows as well as Linux instances)

4.2. Creating a S3 bucket

Experiment-5: AWS Storage (IaaS)

5.1 Changing S3 bucket access permissions, applying replication and lifecycle rules.

5.2. Host a static website on S3

Experiment-6: Identity and Access Management /Security in AWS Cloud Environment

6.1. Create user, role and policies and access it through Management console and CLI.

6.2. Show how to use MFA in IAM

6.3. Adding a WAF to a Web application & its Load balancer and blocking access

Experiment-7: Serverless Computing (AWS Lambda)

7.1. Creating, starting and stopping EC2 instances using AWS Lambda.

Experiment-8: AWS ElasticBeanstalk (PaaS)

8.1 Create and manage a LAMP stack using Amazon EC2 Instance and host an html file on it (Iaas+PaaS).

8.1. Deploy an application on Elastic Beanstalk

Experiment-9: Openstack Installation and VM deployment

9.1. Learn Hands-on OpenStack Installation: Install OpenStack on single node using Packstack

9.2. Familiarize with OpenStack dashboard – Horizon: Go through each and every component of the dashboard and understand the meaning of the sections and components therein

9.3. Deploy a virtual machine instance [using OpenStack Nova]: Create, Deploy, Test and Destroy a VM instance using OpenStack Nova component via Horizon

Experiment-10: Openstack

10.1. Deploy a Linux VM from an ISO image [OpenStack – Nova, Cinder and Glance]: Create an image in OpenStack Glance, Create a VM with a volume and deploy the image

10.2. Deploy a VM from an image snapshot [OpenStack – Nova, Cinder and Glance]: Create a VM, take a snapshot of the VM and store the VM image in Glance and create a new VM image in Nova from the created snapshot

10.3. User and Project Management [OpenStack – Horizon advanced, Keystone]: Explore PM functions in OpenStack and try basic settings

Experiment-11: Openstack and components

11.1. Common cloud management tasks [OpenStack – Horizon, Keystone] Work with quotas in a Project – assign, increase, decrease quotas for compute and storage

11.2. Adding a new compute node [OpenStack – Nova advanced] Add a new compute note in Nova and create VMs on the new new node

Experiment-12: Overview of Nagios

12.1 Explore monitoring tool Nagios and check the parameters it can monitor

Total Lab hours 30

Textbooks

1. IBM Material Cloud Deployment Model Lab
2. Theo H King, "The Ultimate Guide From Beginners To Advanced For The Amazon Web Services", Independently Published, 2020.
3. Kevin Jackson, Cody Bunch, and Egle Sigler, "OpenStack Cloud Computing Cookbook", 3rd Edition, Packt, 2015.

Reference Books

1. John Culkin, and Mike Zazon, "AWS Cookbook", 1st Edition, O'Reilly Media, 2021.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	**Quiz & Viva**	**Performance & Lab Report**
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSDV3018P	Containerization and DevOps	4	0	0	4
Total Units to be Covered: 06	Total Contact Hours: 60				
Prerequisite(s):	Cloud Computing Fundamentals, Cloud Computing Architecture and Deployment Models	Syllabus version: 1.0			

Course Objectives

The objective of this course are

1. To understand Containerization with Docker, emphasizing automation and orchestration.
2. To master DevOps fundamentals for collaborative and automated software delivery.
3. To gain practical knowledge of DevOps applications in real-world scenarios.
4. To explore DevOps principles, lifecycles, and essential tools for efficient software development and deployment.

Course Outcomes

On completion of this course, the students will be able to

CO1: Comprehensive understanding of containerization and the principles of DevOps.

CO2: Apply Docker fundamentals by creating and managing Docker containers, demonstrating comprehension and application of containerization concepts.

CO3: Demonstrate a deep understanding of container orchestration and apply it practically by creating and managing services using container orchestration platforms.

CO4: Assess DevOps adoption strategies, analyze its impact on business, and synthesize cross-functional team dynamics and tool selection within the DevOps context.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	1	-	-	2	-	-	-	-	-	-	1	-	-	3
CO 2	-	1	-	-	2	-	-	-	-	-	-	1	2	1	3
CO 3	-	1	-	-	2	-	-	-	-	-	-	1	2	1	3

CO4	**-**	1	**-**	**-**	2	**-**	**-**	**-**	**-**	**-**	**-**	1	1	1	3
Average	**-**	1	0.75	**-**	2	**-**	**-**	**-**	**-**	**-**	**-**	1	1.25	0.75	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to Containerization and DevOps **8 Lecture Hours**

Virtualization and its types-Server Virtualization, Operating System Virtualization

Containers & DevOps: Introduction to Containers, Understanding DevOps Principles and Practices, Benefits of Containerization in DevOps, Docker vs. Virtualization, Different vendors for containers in the market,

Docker: Introduction to Docker: What is Docker and What isn't Docker?, Overview of Docker editions, Installation of the Docker engine, Docker terminology, Docker community, Docker community edition, Docker enterprise edition, Build Kit features, Docker compose, Docker architecture-Docker host, Docker Daemon, Docker Hub, Docker API, Docker Objects, Docker and DevOps tools

Software Designing Architectural Approaches: Server based and Serverless architecture-Monolithic, Service Oriented Architecture, Microservices, Comparison of different architectural approaches, Docker and Microservices-apart and together

Unit II: Docker Fundamentals **12 Lecture Hours**

Basic Commands: Docker container Lifecycle commands, Docker desktop , Checking docker version, Running your first NGINX application, Docker repository, Docker tags, Docker TAG examples, Docker TAG scenario, Tagging scheme, Docker images, Layers, Docker file, Docker file instructions, Managing containers and images, Creating Docker Images, Dockerfile Best Practices, Pushing Images to Docker Hub, Running your own Docker container.h

Data Management in Docker: Persisting data in docker, Approaches: Volumes, bind mount, Differences, Volumes-creation, listing, --mount flag, -v flag, removing, inspecting bind mount- creation, --mount flag, -v flag, removing, tmpfs mount, Use case scenarios volumes, bind mount and tmpfs mount.

Networking in Docker: Networking in Docker, Docker network drivers- bridge, host, overlay

ipvlan, macvlan, Publishing ports, IP address and host name, DNS services, Network Drivers use case summary, creating and removing a user-defined bridge, managing a user-defined bridge- connect a container to a user defined bridge, Disconnect a container from a user-defined bridge, Connect a container to the default bridge network.

Continuous Integration: Docker as a build environment, GitHub Actions, Building CI/CD Pipleine using Git Hub actions.

Unit III: Automation and Orchestration **12 Lecture Hours**

Container Orchestration need and Overview: Key concepts in orchestration, popular orchestra platforms: Swarm Docker, Kubernetes, Apache mesos

Docker Compose: Features, use cases, history and using Docker Compose

Docker Swarm: Feature highlights, Swarm mode key concepts-swarm, nodes, services & tasks, load balancing; Create a swarm, Add nodes to swarm, Deploy a service, Inspect the service, Scale the service, Delete the service, Apply rolling updates, Drain a node, Use swarm mode routing mesh

Kubernetes: Overview, Traditional deployment era, virtualized deployment era , container deployment era, Need of Kubernetes, what Kubernetes is not?, Kubernetes components: control plane components, Node components, Addons, Kubernetes API, Cluster Architecture-Nodes, Communication between nodes and the control Plane, Controllers, Leases, Cloud Controller Manager, Container runtime Interface, Garbage Collection, Containers: Images, Container Environment, Runtime class, Container Lifecycle hooks, Workloads- Pods and Workload Resources, Services, Load balancing and Networking, Storage, Configuration, Policies and security

Case Study: Docker Swarm vs Kubernetes

Case Study: Amazon ECS and EKS services

Unit IV: DevOps: Principles and Practices **8 Lecture Hours**

Overview, Working, Benefits, DevOps history, DevOps principles and lifecycle, DevOps practice, DevOps adoption: Deming, lean manufacturing, and Kaizen, Lean manufacturing, Lean standards of manufacturing, DevOps: IBM view, Four DevOps adoption paths- Steer adoption path, Develop and test adoption path, Collaborative development, Continuous testing, Way to deployment, Continuous customer feedback and optimization,

DevOps architecture and resilience, Cloud resiliency, DevOps resiliency, Four stages of the resilience process-Detect, DevOps style; Alert with a cloud and DevOps mindset; Respond & recover using automation and appropriate failover strategy; Refine & test, achieving incremental improvements

Unit V: DevOps Adoption and Business Patterns **8 Lecture Hours**

Business needs for DevOps , DevOps teams and cross functioning of teams, Silos in the world of software development and their role in project delivery, DevOps teams and cross functioning of teams, Application team v/s. Platform team, System admins and other stakeholders, Continuous integration vs continuous deployment vs continuous delivery, DevOps tools- Continuous development, Continuous integration, Continuous Testing, Continuous Deployment, Continuous Monitoring, Lean Thinking and Methods – Kaizen, Agile Vs DevOps, DevOps impact on developers, DevOps impact on operations, Successful DevOps adoption, Challenges of DevOps adoption, Introduction to Kanban, Types of kanban board, create a kanban board, Kanban with IBM tools. Scrum application delivery pipeline and support team, an orchestration framework for continuous delivery, Software release plan, Feedback and learning from feedback and improving the delivery, DevOps toolchain, DevSecOps, DevOps vs SRE. Select the right tool for DevOps: Docker, Kubernetes, Puppet, Ansible, Other tools, DevOps monitoring tools, Version control, and code repository

Unit VI: DevOps Tools **12 Lecture Hours**

Version Control Tools: GitHub-GitLab-BitBucket, GitHub-CLI, Desktop, Branches, forks, and Pull requests, Repositories, GitHub actions, GitHub packages, Webhooks, API

Continuous Integration Tools: Jenkins, TravisCI, Jenkins- Introduction to Jenkins, Jenkins architecture and components: Installation and setup of Jenkins, Creating and Configuring Jenkins Jobs Jenkins Pipelines Source Code Management (SCM) Integration Building and Testing with Jenkins

Infrastructure as Code (IaC) Tools: Understanding IaC, Popular IaC tools- Terraform, Ansible

Monitoring and Logging Tools: Monitoring and logging in DevOps**,** Popular monitoring and logging tools- Prometheus, ELK stack

Collaboration in a DevOps Team: Popular collaboration and communication tools- Slack, Microsoft Teams, Integrating tools into the DevOps workflow

Case study: CI/CD technique.

Total lecture Hours 60

Textbooks

1. Container Orchestration & Infrastructure Automation, IBM ICE Publications.
2. James Turnbull, “The Docker Book: Containerization is the new virtualization”, First Edition, Shroff Publishers, 2019.
3. Jason Cannon, “Docker: A Project-Based Approach to Learning”, Independently published, 2021.
4. Deepak Gaikwad, Viral Thakkar, “DevOps Tools from Practitioner's Viewpoint”, Wiley, 2019.

Reference Books

1. Ian Miell and Aidan Hobson Sayers, “Docker in Practice”, 2nd Edition , Manning Publications, 2019.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSDV3118P	Containerization and DevOps Lab	0	0	2	1
Total Units to be Covered: 12	Total Contact Hours: 30				
Prerequisite(s):	Cloud Computing Fundamentals Lab , Cloud Computing Architecture and Deployment Lab	Syllabus version: 1.0			

Course Objectives

1. To demonstrate on Containerization technology and use of Docker container automation and orchestration.
2. To gain practical knowledge and hands-on experience on DevOps tools, their roles and importance in the DevOps lifecycle, including continuous integration, infrastructure provisioning, configuration management, and code quality analysis.

Course Outcomes

At the end of this course student should be able to

CO1. Demonstrate the usage of Docker Containers.

CO2. Demonstrate orchestration of Containers images and microservices.

CO3. Apply DevOps tools to effectively implement DevOps practices, including continuous integration, infrastructure provisioning, configuration management, and code quality analysis.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	-	1	2	-	-	1	1	-	-	1	-	-	3
CO 2	1	-	-	1	2	-	-	1	1	-	-	1	-	-	3
CO 3	1	-	-	1	2	-	-	1	1	-	-	1	-	-	3
Average	1	-	-	1	2	-	-	1	1	-	-	1	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Introduction to Dockers

a. Setup & Running a Container
b. Performance evaluation of Virtual box machine deployment and Dockers containers deployment.

Experiment 2: Docker Installation, configuration, Running Images

Experiment 3: Deploying web applications with Docker

Experiment 4: Dockerfile - Containerizing application, Building Images, Tagging, Publishing

Experiment 5: Docker - Volumes, Env, Monitoring (Docker stats), Docker Networks

Experiment 6: Docker Compose - Installation, Creating Compose files, Running Images using Docker-compose

Experiment 7: Create a CI CD pipeline for deploying web application using jenkins

Experiment 8: Ansible

Experiment 9: Chef

Experiment 10: Install, setup and run SonarQube for a local scan.

Experiment 11: Orchestration using Docker compose on multi container applications.

Experiment 12: Study and Analyse container orchestration using Kubernetes.

Total Lab hours 30

Text Books

1. IBM ICE Publications
2. The Docker Book: Containerization is the new virtualization by James Turnbull, Shroff Publishers, First Edition, 2019
3. Docker: A Project-Based Approach to Learning by Jason Cannon, **September 2021**
4. DevOps Tools from Practitioner's Viewpoint, by Deepak Gaikwad, Viral Thakkar, Wiley, Jan 2019

Reference Books

1. Ian Miell, and Aidan Hobson Sayers, "Docker in Practice", 2nd Edition , Manning Publications, 2019.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSVT4018P	**Cloud Application Development**	4	0	0	4
Total Units to be Covered: 05		**Total Contact Hours: 60**			
Prerequisite(s):	Object Oriented Programming, Python Programming, Cloud Computing Architecture and Deployment Models, Containerization and DevOps	**Syllabus version: 1.0**			

Course Objectives

1. Student should be able to get solid theoretical background of high computing paradigms.
2. Students get adequate conceptual understanding of cloud application development platform and framework.
3. Student can explore cloud applications and identify applications suitable for implementation on cloud.

Course Outcomes

On completion of this course, the students will be able to

CO1: Correlate the underlying technologies for cloud application development with basic programming skills

CO2: Apply the concepts of high throughput, data intensive applications and Task programming

CO3: Discover various cloud computing platforms for application development

CO4: Analyse the use of automation in cloud application development and understand the real time use cases

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	2	-	-	-	-	-	-	-	-	1	1	2	3
CO 2	1	1	2	-	2	-	-	-	-	-	-	1	1	2	3
CO 3	1	1	2	1	2	-	-	-	-	-	-	1	1	2	3
CO4	1	1	2	1		-	-	-	-	-	-	-	1	2	3

Average	1	0.6	2	0.5	2	-	-	-	-	-	-	0.75	1	2	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

Syllabus

Unit I: Basics of Cloud Computing and introduction to MapReduce programming for application development **12 Lecture Hours**

Business case for implementing cloud application, Requirements collection for cloud application development, Cloud service models and deployment models, Open challenges in Cloud Computing: Cloud inter-operability and standards, scalability and fault tolerance, security, trust and privacy, Data intensive computing: MapReduce programming, what is data-intensive computing? Characterizing data-intensive computations, Data intensive research issues, Historical perspective, The early age: High-speed wide area networking, The early age, Data grids, Data clouds and big data,

Unit II: Understanding basic Task Programming skills and MapReduce Programming **12 Lecture Hours**

Understanding the basics of HTML, javascript, Client-side web scripting: DOM and AJAX, Server-side applications: node.js, Introduction to task programming, High-throughput computing: Task programming, Task computing, Task-based application models, MPI reference scenario, MPI program structure, Workflow applications with task dependencies, Workflow technologies abstract model of workflow system, Cloud application platform task-based programming, Task programming model scenario, iTask and cloud application platform task, iTask interface, Wrapping an iTask into an cloud application platform instance, Controlling task execution, Dynamic task submission. Databases and data-intensive computing, Technologies for data intensive computing, High performance distributed file systems and storage clouds, Not only SQL (NoSQL) systems, Prominent implementations supporting data: Intensive applications, Amazon dynamo architecture, Google bigtable architecture, Apache Cassandra, Hadoop HBase, Programming platforms, The MapReduce programming model, MapReduce computation workflow, Google MapReduce infrastructure overview, Variations and extensions of MapReduce

Unit III: Application Development: Google and Microsoft Azure 12 Lecture Hours

Accessing the clouds: Web application vs Cloud Application, Frameworks: Model View Controller (MVC), Struts, Spring, Maven, Gradle. Cloud platforms in Industry – Google AppEngine (serverless platform for build scalable web and mobile back ends in any programming language), Microsoft Azure, Openshift (building containerized applications), CloudFoundry (provides a highly efficient, modern model for cloud native application delivery on top of Kubernetes)

Unit IV: Application Development on AWS and DevOps tools 12 Lecture Hours

Best practices in architecture cloud applications in AWS cloud, Elastic Beanstalk (Deploy and Provision web applications), Amazon Simple Queue Service (SQS), RabbitMQ, Cloud applications: Amazon Simple Notification Service (Amazon SNS), multi-player online game hosting on cloud resources, building content delivery networks using clouds. Puppet and Chef – steps for automation: Introduction, files and packages, services and subscriptions, exec and notify, facts, conditional statements and logging, configuration management tools, automation, configuration orchestration

Unit V: APIs for cloud application development and 12 Lecture Hours

real time use cases

RESTful API Usage, RESTful API Design & Implementation, Add Functionality to REST API, storing objects in the Cloud, Session management, Working with third party APIs: Overview of interconnectivity in Cloud ecosystems. Facebook API, Twitter API, Google API. Use cases of cloud applications, Scientific applications, Healthcare: ECG analysis in the cloud, Biology: Protein structure prediction, Gene expression data analysis for cancer diagnosis, GeoScience : Satellite image processing, Business and consumer applications, SalesForce.com, Productivity, DropBox, Google Docs, EyeOS, Social networking : Facebook, Media applications: animoto, 3D rendering on private clouds, Video encoding on the cloud: Encoding.com, Multiplayer online gaming scalable processing of logs for network games.

Total lecture Hours 60

Textbooks

3. David E. Y. Sarna, "Implementing and Developing Cloud Computing Applications", CRC Press, 2010.

4. Shagun Bakliwal, "Hands-on Application Development using Spring Boot: Building Modern Cloud Native Applications by Learning RESTFul API, Microservices, CRUD Operations, Unit Testing, and Deployment", 1st Edition, BPB Publications, 2021.

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSVT4118P	Cloud Application Development Lab		0	0	2	1
Total Units to be Covered: 12		Total Contact Hours: 30				
Prerequisite(s):	Object Oriented Programming, Python Programming, Cloud Computing architecture and Deployment Models, DevOps ad Containerization	Syllabus version: 1.0				

Course Objectives

1. To equip students with the skills to effectively set up a cloud application development platform, showcasing their ability to configure and deploy the necessary infrastructure and tools.
2. To demonstrate the capability to leverage advanced computing capabilities for the development and deployment of cloud applications,
3. To analyze the implementation processes and challenges involved in deploying cloud applications across different domains.

Course Outcomes

At the end of this course student should be able to

CO1: Demonstrate proficiency in setting up a cloud application development platform, showcasing the ability to configure and deploy the necessary infrastructure and tools for cloud application development.

CO2: Demonstrate ability to leverage advanced computing capabilities to develop and deploy cloud applications.

CO3: Analyzing the implementation processes and criticalities involved in the deployment of cloud applications across various domains.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	2	-	-	-	-	-	-	-	-	1	1	2	3
CO 2	1	1	2	-	2	-	-	-	-	-	-	1	1	2	3

CO 3	1	1	2	1	2	**-**	**-**	**-**	**-**	**-**	**-**	1	1	2	3
Average	1	0.67	2	0.33	1.33	**-**	**-**	**-**	**-**	**-**	**-**	1	1	2	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Installation of Single Node Hadoop Cluster on Ubuntu and run simple program like wordcount.

Experiment 2: Install Google App Engine. Create Hello World App and other Simple Web Applications.

Experiment 3: Build Apps at Scale with Google App Engine.

Experiment 4: Setup and Configure Cloud application development platform (Like Amazon application development platform).

Experiment 5: Using cloud Platform API develop and deploy prototype application.

Experiment 6: Generate high volume workloads of applications on cloud platform

Experiment 7: Data Intensive Computing: Map-Reduce Programming implement a sample Map-Reduce procedure using cloud platform

Experiment 8: Developing Message-Based Applications with RabbitMQ

Experiment 9: Implement data analytics for some application using cloud platform

Experiment 10: Implement Machine Learning model/s on cloud platform like Sage Maker, Amazon Polly etc.

Experiment 11: Install apache http server, enable and start the services using Chef Cookbook.

Experiment 12: Create a CI/CD pipeline using scratch for deploying application (using Git and Jenkins)

Total Lab hours 30

Textbooks

1. Cloud Application Development Lab Manual (IBM ICE Publications)
2. Atul V. Mistry, “Expert AWS Development”, Packt Publishing, 2018.

Reference Books

2. John Smart, “Jenkins: The Definitive Guide: Continuous Integration for the Masses”, 1st Edition, O'Reilly Media, 2011.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSVT4019P	**Cloud Computing Security and Management**	3	0	0	3
Total Units to be Covered: 06		**Total Contact Hours: 45**			
Prerequisite(s):	Data communication and Networks , Cloud Computing Architecture and Deployment Models	**Syllabus version: 1.0**			

Course Objectives

1. Develop a comprehensive understanding of cloud security fundamentals,
2. Analyze the importance of data security in the cloud and its principles,
3. Evaluate the role of IAM in cloud security and its components, goals, and principles.
4. Apply network security fundamentals and CIA principles in cloud environments.
5. Apply secure coding practices, including input validation, authentication, and error handling in cloud applications.

Course Outcomes

CO1: Assess the security challenges and vulnerabilities specific to cloud infrastructure, including physical security measures and virtualization technology
CO2: Identify and define key data security concepts and mechanisms used to protect data in transit, at rest, and during processing in cloud environments.
CO3: Implement and assess IAM components and policies in cloud environments,
CO4: Evaluate the importance of network and application security in cloud computing, demonstrating an understanding of security fundamentals

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	-	-	1	-	-	-	-	-	-	1	-	1	3
CO 2	1	-	-	-	1	-	-	-	-	-	-	1	-	1	3
CO 3	1	-	-	-	1	-	-	-	-	-	-	1	-	1	3
CO4	1	-	-	-	1	-	-	-	-	-	-	1	-	1	3
Average	1	-	-	-	1	-	-	-	-	-	-	1	-	1	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Security Overview **8 Lecture Hours**

Overview of cloud security- Security Services - Confidentiality, Integrity, Authentication, Nonrepudiation, Access Control - Basic of cryptography - Conventional and public-key cryptography, hash functions, authentication, and digital signatures.

Common security risks in cloud environments, Threat modeling in the cloud, Security considerations for different cloud deployment models, Cloud security: Physical Security, Network Security, Data Security, Identity and Access Management (IAM), Application Security, Virtualization Security, Operating System Security, Security Compliance and Governance, Cloud Access Security Broker (CASB) , Logging and Monitoring, Incident Response and Forensics, Encryption Key Management, Multi-Factor Authentication (MFA), Backup and Recovery, Security Training and Awareness, Zero Trust Security, Shared responsibility model, Cloud Security Alliance

Unit II: Infrastructure security & Virtualization Security **12 Lecture Hours**

Introduction to Physical and Infrastructure Security in the Cloud: Role of physical security in cloud computing, Key physical security components and objectives, Cloud infrastructure and data center architecture, Security challenges in shared data center environments, Data center architecture and design considerations, Redundancy and fault tolerance, Environmental controls (temperature, humidity, fire suppression), Physical access control and monitoring, Intrusion detection systems (IDS) and intrusion prevention systems (IPS), Video surveillance and access control systems, Visitor management and access logging, Data center personnel and roles (e.g., NOC, SOC), Security awareness and training for data center staff, Change management and incident response in data center operations, Asset management and tracking Data replication and backup strategies, Failover and high availability configurations, Testing and simulating disaster scenarios

Virtualization security threats and vulnerabilities, Isolation and segmentation of VMs, VM escape attacks and mitigation, Security for containerized applications

OS-Level Security: Hardening operating systems in the cloud, Patch management and vulnerability assessment, Anti-malware and intrusion detection for OS instances, OS-level access controls and permissions Secure OS Configuration: Secure boot and integrity verification, Network security configurations for OS instances, Secure shell (SSH) and remote access security, Application and service hardening OS Security Best Practices: OS security baseline standards, Security updates and patch management policies, Security information and event management (SIEM) integration, Monitoring and auditing OS-level activities

Case Studies: VM escape attack mitigation in a cloud service provider environment, OS-level security in a containerized application deployment

Unit III: Data Security **7 Lecture Hours**

Importance of data security in the cloud, Data security principles: CIA (Confidentiality, Integrity, Availability). Shared responsibility model in cloud security,

Data Classification and Protection: Data classification and sensitivity assessment, Data protection requirements (e.g., PII, PHI), Data masking and anonymization, Data retention policies and disposal Encryption Techniques: Encryption fundamentals, Data encryption at rest using disk and database encryption, Encryption in transit with TLS/SSL, Encryption for data during processing (homomorphic encryption) Compliance and Regulatory Requirements: Secure Data Sharing and Collaboration, Secure file sharing and collaboration tools, Secure data exchange with external parties, Data leakage prevention (DLP) and protection mechanisms, Implementing secure data sharing policies Data Security in Multi-Cloud Environments: Challenges and considerations for multi-cloud data security, Data replication and synchronization, Data governance in multi-cloud architectures, Data compliance frameworks (e.g., GDPR, HIPAA), Auditing and monitoring for compliance, Cloud provider compliance certifications, Privacy and data protection in the cloud

Unit IV: Network Security **7 Lecture Hours**

Importance of network security in the cloud, Network security fundamentals: CIA (Confidentiality, Integrity, Availability), Network security in a shared responsibility model

Cloud Network Architecture and Components: Cloud network models (VPC, VNet), Subnetting and IP addressing in the cloud, Load balancers and content delivery networks (CDNs), Virtual private clouds (VPCs) and security groups Network Protocols and Encryption: Transport layer security (TLS/SSL), VPNs and encryption for data in transit, Secure tunneling protocols (IPsec), Securing DNS and BGP Network Security Controls and Policies: Firewall fundamentals and best practices, Intrusion detection and prevention systems (IDS/IPS), Network access control lists (NACLs) and security policies, Security groups and network segmentation Secure Data Transmission and VPNs: Site-to-site VPNs for hybrid cloud environments, Remote access VPNs and secure connectivity, VPN protocols (PPTP, L2TP, OpenVPN), VPN client configuration and security considerations Network Monitoring and Security Analytics: Network monitoring tools and techniques, Real-time traffic analysis and anomaly detection, Logging and alerting for network security incidents, Security information and event management (SIEM) integration Cloud Network Security Best Practices: Implementing network security policies and procedures, Security groups and firewall rule management, Network security automation and orchestration, Network security in serverless and containerized environments

Case Studies: DDoS mitigation in a cloud-hosted e-commerce platform, Securing microservices communication in a cloud-native applications

Unit V: Identity and Access Management (IAM) in the Cloud 6 Lecture Hours

The role of IAM in cloud security, IAM components and stakeholders, IAM goals and principles, IAM trends and challenges

Authentication and authorization in the cloud, Authorization models (RBAC, ABAC), Role-based access control (RBAC), Identity federation and single sign-on (SSO), IAM Technologies and Protocols: LDAP and Active Directory, OAuth 2.0 and OpenID Connect, SAML (Security Assertion Markup Language), OAuth and API security, Multi-factor authentication (MFA), IAM Policy and Governance: Creating IAM policies and procedures, IAM policy lifecycle management, Policy enforcement and compliance, IAM auditing and reporting

Case Studies: AWS IAM and Azure AD

Unit VI: Application Security 5 Lecture Hours

Importance of application security in the cloud, Application security fundamentals, OWASP Top Ten and common vulnerabilities

Secure Coding Practices: Secure coding principles and best practices, Input validation and output encoding, Authentication and authorization in cloud applications, Error handling and logging for security Data Security in Cloud Applications: Data protection and encryption, Handling sensitive data (PII, PHI), Data access controls and role-based access control (RBAC), Data leakage prevention (DLP) in applications API Security: API security considerations and challenges, OAuth 2.0 and OpenID Connect for authentication and authorization, Rate limiting and throttling, Securing API endpoints and data transmission Application Security Testing: Static application security testing (SAST), Dynamic application security testing (DAST), Interactive application security testing (IAST), Continuous integration/continuous deployment (CI/CD) pipeline security testing Cloud Application Security Best Practices: Secure application development life cycle (SDLC), Security by design in cloud-native applications, Web application firewalls (WAFs) and application security policies, Secure application deployment strategies

Case Studies: Securing a SaaS-based customer portal, API security challenges in a cloud-native fintech application

Total lecture Hours 45

Textbooks

1. Ronald L. Krutz, and Russell Dean Vines, "Cloud Security: A Comprehensive Guide to Secure Cloud Computing", Wiley Publishing, 2010.
2. Tim Mather, SubraKumaraswamy, and ShahedLatif, “Cloud Security and Privacy", O’Reilly Media, Inc., 2009.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSVT4119P	**Cloud Computing Security and Management Lab**	0	0	2	1
Total Units to be Covered: 12	**Total Contact Hours: 30**				
Prerequisite(s):	Cloud Computing Architecture and Deployment Models Lab and Cloud Application Development Lab	**Syllabus version: 1.0**			

Course Objectives

1. To empower students to proficiently implement and manage access control strategies across various cloud platforms, ensuring data and resource security.

2. To enable students to effectively design and deploy security measures such as attribute-based access control (ABAC), multi-factor authentication (MFA), and role-based access control (RBAC) to safeguard cloud environments in real-world scenarios.

Course Outcomes

On completion of this course, the students will be able to

CO1: Demonstrate the ability to design and deploy secure cloud infrastructures across various cloud computing platforms.

CO2: Evaluate and compare access control mechanisms across various cloud computing platforms.

CO3: Create and manage customer-managed policies in AWS to enforce security and compliance standards.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	-	1	2	-	-	-	1	-	-	1	-	-	3
CO 2	1	-	-	1	2	-	-	-	1	-	-	1	-	-	3
CO 3	1	-	-	1	2	-	-	-	1	-	-	1	-	-	3
Average	1	-	-	1	2	-	-	-	1	-	-	1	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Building your first Virtual Private Cloud-AWS

Experiment 2: Delegate access across AWS accounts using roles-AWS

Experiment 3: Create a customer managed policy-AWS

Experiment 4: Use attribute-based access control (ABAC)-AWS

Experiment 5: Permit users to manage their credentials and MFA settings (AWS)

Experiment 6: Implement Azure key vault

Experiment 7: Manage Access to Azure with Role-Based Access Control

Experiment 8: Manage Identity and Access- MFA, and Conditional Access (Azure)

Experiment 9: Manage Identity and Access- Implement Directory Synchronization (Azure)

Experiment 10: Smart Access Control on GCP: Web, programmatic, and command-line access

Experiment 11: Write an IAM policy by using client libraries (GCP)

Experiment 12: Using AWS WAF to control access to your content (AWS)

Total Lab hours 30

Textbooks

1. Ronald L. Krutz and Russell Dean Vines, "Cloud Security: A Comprehensive Guide to Secure Cloud Computing", Wiley Publishing Inc., 2010.
2. Tim Mather, Subra Kumaraswamy and Shahed Latif, "Cloud Security and Privacy", O'Reilly Media Inc, 2009.
3. Imad M. Abbadi, "Cloud Management and Security", Wiley Publishing Inc., 2015.

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

4 . FULL STACK DEVELOPMENT Track

Course Code	Course name		L	T	P	C
CSFS2003P	Frontend Development		4	0	0	4
Total Units to be Covered: 05		Total Contact Hours: 60				
Prerequisite(s):	Basic knowledge of HTML and CSS, Familiarity with JavaScript fundamentals	Syllabus version: 1.0				

Course Objectives

1. Understand the fundamentals of web development and front-end technologies.
2. Design and develop responsive web pages using HTML and CSS.
3. Build interactive web applications using JavaScript.
4. Utilize front-end frameworks and libraries to streamline development processes.
5. Implement best practices for accessibility and usability in web design.
6. Optimize web performance and user experience.
7. Collaborate effectively with backend developers and designers.

Course Outcomes

On completion of this course, the students will be able to

CO1: Understand Web Development Fundamentals

CO2: Create and build web pages and applications.

CO3: Utilize front-end frameworks and optimize web performance.

CO4: Apply security best practices and conduct testing.

CO5: Demonstrate project development skills.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	3	-	-	-	1	-	-	-	-	-	-

CO 2	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO 3	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-
CO4	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO5	-	-	2	-	2	-	-	-	2	-	-	2	-	-	-
Average	-	-	0.4	-	2	-	-	-	0.6	-	-	0.4	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to Frontend Development, Web Design 12 Lecture Hours

Overview of web development and frontend technologies, Introduction to development environments (IDEs, text editors, browser developer tools), HTML5 essentials: tags, attributes, semantic markup, CSS fundamentals: selectors, box model, layout techniques, Introduction to responsive web design principles, CSS media queries and viewport settings, Flexbox and CSS Grid for flexible layouts, Introduction to Bootstrap or other CSS frameworks

Unit II: JavaScript and Web Optimization 12 Lecture Hours

Introduction to JavaScript: variables, data types, control flow, Functions and scope, DOM manipulation and event handling, Introduction to jQuery or other JavaScript libraries, Working with arrays and objects, Asynchronous programming and AJAX, Introduction to ES6+ features (arrow functions, modules, etc.), Introduction to modern JavaScript frameworks (React, Vue.js, Angular), Understanding web performance metrics, Techniques for optimizing CSS and JavaScript, Asset optimization (images, fonts, etc.), Introduction to caching and CDNs

Unit III: Frontend: Building and Testing 12 Lecture Hours

Introduction to task runners (Gulp, Grunt) or bundlers (Webpack, Parcel), CSS preprocessors (Sass, Less) for enhanced styling, Introduction to version control systems (Git), Accessibility and Usability, Principles of web accessibility, Techniques for creating accessible web content, Usability best practices and user-centered design principles, Introduction to frontend testing frameworks (Jest, Mocha), Unit testing and integration testing, Testing user interfaces and interactions.

Unit IV: JSX and Redux **12 Lecture Hours**

Why JSX, Embedding JavaScript, Expression in JSX, JSX as an Expression, Nested elements in JSX, JSX, Attributes, JSX Comments, JSX Styling and representation as object, The State of the Component, Defining State, Changing the State, Props, Validation, Validators, Elements, Rendering Element, About render (), Creating React Element, Updating Element, components, Introducing Components, Types of Components, Functional Component, Functional Components as Stateless, Using Functional Component, Redux Concepts, Redux Principles, Data Flow, Actions, Functions, Reduces, Testing , Dev-Tools, React & Redux Integrate.

Unit V: Web Security **12 Lecture Hours**

Introduction to web security principles, SPA frameworks, authentication and authorization systems, API analysis, detecting frameworks and libraries, Common Security Issues (eg. Cross-Site Scripting, CSRF, XXE, Injection), Common Countermeasures (e.g., Authentication, Authorization, HTTPS), securing web applications, reviewing code for security, vulnerability discovery and management, Web Application Firewalls.

Total lecture Hours 60

Textbooks

1. Stefan Baumgartner, “Front End Tooling with Gulp”, Manning Publication.
2. “HTML5 Black Book”, Dreamtech Publications, 2016.
3. Ben Frain, “Responsive Web Design with HTML5 and CSS”, 4th Edition, Packt Publication, 2022.

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSFS2101P	**Frontend Development Lab**	0	0	2	1
Total Units to be Covered: 11	**Total Contact Hours: 30**				
Prerequisite(s):	Basic knowledge of HTML and CSS, Familiarity with JavaScript fundamentals	**Syllabus version: 1.0**			

Course Objectives

1. Understand the fundamentals of web development and front-end technologies.
2. Design and develop responsive web pages using HTML and CSS.
3. Build interactive web applications using JavaScript.
4. Utilize front-end frameworks and libraries to streamline development processes.
5. Implement best practices for accessibility and usability in web design.
6. Optimize web performance and user experience.
7. Collaborate effectively with backend developers and designers.

Course Outcomes

On completion of this course, the students will be able to

CO1: Understand Web Development Fundamentals

CO2: Create and build web pages and applications.

CO3: Utilize front-end frameworks and optimize web performance.

CO4: Apply security best practices and conduct testing.

CO5: Demonstrate project development skills.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	3	-	-	-	1	-	-	-	-	-	-
CO 2	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO 3	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-
CO4	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-

CO5	-	-	2	-	2	-	-	-	2	-	-	2	-	-	-
Average	-	-	0.4	-	2	-	-	-	0.6	-	-	0.4	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment-1: Write a program to create a simple webpage using HTML.

Experiment-2: Write a program to create a website using HTML CSS and JavaScript

Experiment-3: Write a program to build a Chat module using HTML CSS and JavaScript

Experiment-4: Write a program to create a simple calculator Application using React JS

Experiment-5: Write a program to create a voting application using React JS

Experiment-6: Write a program to create and Build a Password Strength Check using Jquery

Experiment-7: Write a program to create and Build a star rating system using Jquery

Experiment-8: Create a Simple Login form using React JS

Experiment-9: Using the CMS users must be able to design a web page using the drag and drop method

Experiment-10: Create a project on Grocery delivery application

Experiment-11: Connecting our TODO React js Project with Firebase

Total Lab hours 30

Textbooks

1. Stefan Baumgartner, “Front End Tooling with Gulp”, Manning Publication, 2016.
2. DT Editorial Services, “HTML5 Black Book”, 2nd Edition, Dreamtech Publications, 2016.
3. Ben Frain, “Responsive Web Design with HTML5 and CSS”, 4th Edition, Packt Publication, 2022.

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name		L	T	P	C
CSFS3005P	**Backend Development**		4	0	0	4
Total Units to be Covered: 05		**Total Contact Hours: 60**				
Prerequisite(s):	Basic knowledge of HTML and CSS, Familiarity with JavaScript fundamentals		**Syllabus version: 1.0**			

Course Objectives

1. Understand the fundamentals of web development and back-end technologies.
2. Understand and explore backend frameworks, databases and data modelling.
3. Design and Implement APIs and API related tasks.
4. Utilize back-end frameworks and libraries to streamline development processes.
5. Implement best practices for security, testing and debugging.

Course Outcomes

On completion of this course, the students will be able to

CO1: Understand Back End Development Fundamentals

CO2: Create and build web pages and applications.

CO3: Utilize frameworks and APIs.

CO4: Apply security best practices and conduct testing.

CO5: Demonstrate effective team collaboration skills.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	3	-	-	-	1	-	-	-	-	-	-
CO 2	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO 3	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-
CO4	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO5	-	-	2	-	2	-	-	-	2	-	-	2	-	-	-
Average	-	-	0.4	-	2	-	-	-	0.6	-	-	0.4	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to Backend Development, Server-Side Programming **12 Lecture Hours**

Overview of backend development and its importance, Client-server architecture and communication protocols, Introduction to HTTP and RESTful APIs, Introduction to backend languages and frameworks (e.g., Python, Node.js, Django, Flask), Introduction to server-side programming languages (e.g., Python, JavaScript), Basics of handling HTTP requests and responses, Managing server-side sessions and cookies, Introduction to server-side rendering and template engines.

Unit II: Data Management, API Development and Frameworks **12 Lecture Hours**

Introduction to databases (relational and NoSQL), Designing database schemas and models, Performing CRUD (Create, Read, Update, Delete) operations on databases, Query optimization and indexing, designing, and implementing RESTful APIs, Authentication and authorization mechanisms, Handling API requests and responses, API versioning and documentation.

Unit III: Frameworks and Performance Optimization **12 Lecture Hours**

Introduction to popular backend frameworks (e.g., Express.js, Django, Ruby on Rails), Working with middleware and routing, integrating third-party APIs and services, Testing and debugging backend applications, understanding security vulnerabilities and best practices, Implementing secure authentication and authorization mechanisms, Caching and performance optimization techniques, Error handling and logging.

Unit IV: Deployment and DevOps **12 Lecture Hours**

Overview of deployment environments (e.g., local, cloud, containers), Monitoring and scaling backend applications, Basics of containerization (e.g., Docker), Continuous integration and deployment (CI/CD), Managing infrastructure configurations, setting up CI/CD pipelines, Building and packaging applications, Automated testing and code quality checks, Continuous Deployment and Release Management, Strategies for releasing software updates, Managing deployment environments, Release orchestration and rollback strategies.

Unit V: Advanced Topics **12 Lecture Hours**

Real-time communication with Web-Sockets, Microservices architecture and design patterns: Overview of microservices architecture and its benefits, Service discovery and service registration, Implementing communication between microservices (REST, messaging), Serverless computing: Understanding serverless architecture and its advantages, Building serverless backend systems using platforms like AWS Lambda or Google Cloud Functions, Event-driven serverless architecture and integration with other services, Function-as-a-Service (FaaS), Performance profiling and optimization strategies.

Total lecture Hours 60

Textbooks

1. David Herron, "Node.js Web Developent" 5th Edition, Packt Publication, 2020.
2. DT Editorial Services, "HTML5 Black Book", 2nd Edition, Dreamtech Publications, 2016.
3. Sam Newman, "Building Microservices", 2nd Edition, O'Reilly Publication, 2021.

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSFS3101P	Backend Development Lab		0	0	2	1
Total Units to be Covered: 13		Total Contact Hours: 30				
Prerequisite(s):	Basic knowledge of HTML and CSS, Familiarity with JavaScript fundamentals	Syllabus version: 1.0				

Course Objectives

1. Understand the fundamentals of web development and back-end technologies.
2. Understand and explore backend frameworks, databases and data modelling.
3. Design and Implement APIs and API related tasks.
4. Utilize back-end frameworks and libraries to streamline development processes.
5. Implement best practices for security, testing and debugging.

Course Outcomes

On completion of this course, the students will be able to

CO1: Understand Back End Development Fundamentals

CO2: Create and build web pages and applications.

CO3: Utilize frameworks and APIs.

CO4: Apply security best practices and conduct testing.

CO5: Demonstrate effective team collaboration skills.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	3	-	-	-	1	-	-	-	-	-	-
CO 2	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO 3	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-
CO4	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO5	-	-	2	-	2	-	-	-	2	-	-	2	-	-	-

Average	-	-	0.4	-	2	-	-	-	0.6	-	-	0.4	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment1 Create a web page with all possible elements of HTML5

Experiment2 Create a web page with all types of Cascading style sheets

Experiment3 Create a Responsive Web page with HTML and CSS

Experiment4 Create Responsive web page with Bootstrap

Experiment5 Programs to demonstrate JavaScript array, object and functions

Experiment6 Client-Side Scripts for Form Validation using JavaScript

Experiment7 Programs to familiarise ES6 concepts

Experiment8 Programs to demonstrate DOM and event handling.

Experiment9 Programs using AJAX with HTML, XML and JSON data

Experiment10 Programs to familiarize Query.(2 program)

Experiment11 Create a website with HTML, CSS and JavaScript (implement Ajax)

Experiment12 Programs to familiarize Server-Side Scripting using Node JS

Experiment13 Programs using Mongo DB database with Node JS

Total Lab hours 30

Textbooks

1. David Herron, "Node.js Web Developent", 5th Edition, Packt Publication, 2020.
2. DT Editorial Services "HTML5 Black Book", 2nd Edition, Dreamtech Publications, 2016.
3. Sam Newman, "Building Microservices", 2nd Edition, O'Reilly Publication, 2021.

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name		L	T	P	C
CSFS3007P	Microservices and Spring-Boot		3	0	0	3
Total Units to be Covered: 05		Total Contact Hours: 45				
Prerequisite(s):	Familiarity with Java and JavaScript fundamentals	Syllabus version: 1.0				

Course Objectives

1. Develop Microservices with Spring Boot
2. Implement Service Discovery and Load Balancing
3. Implement Event-Driven Communication and develop API gateway
4. Monitor, Test and Deploy microservices effectively
5. Develop Troubleshooting and Debugging abilities.

Course Outcomes

On completion of this course, the students will be able to

CO1. Understanding service-oriented architecture and its utility

CO2. Compare microservices with SOA

CO3. Design patterns of architecture for a microservices led approach for scaling IT solutions.

CO4. Establish role of CI/CD with containers

CO5: Demonstrate project development skills.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	3	-	-	-	1	-	-	-	-	-	-
CO 2	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO 3	-	-	-	-	1	-	-	-	-	-	-	-	-	-	-
CO4	-	-	-	-	2	-	-	-	-	-	-	-	-	-	-
CO5	-	-	2	-	2	-	-	-	2	-	-	2	-	-	-
Average	-	-	0.4	-	2	-	-	-	0.6	-	-	0.4	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Microservices & Spring Boot **10 Lecture Hours**

Spring Boot Introduction, Introduction, Software Architectures, Microservices, Architecture Layers, Application Layer, Business Layer, Enterprise Layer, Need of SpringBoot and its stakeholders, Understand architectural patterns and styles, Explain monolithic architecture, strengths and limitations, Describe SOA architecture, strengths and limitations, Discuss SOA components and its principles, Identify microservices architecture, strengths and limitations,

Unit II: Microservices Components **10 Lecture Hours**

List microservices components and its principles, define domain driven design concepts, explain strategic and tactical design, discuss domain, bounded context, ubiquitous language and context mapping, Describe the building blocks of DDD, its strengths and limitations, Explain how domain driven design applies to microservices, Discuss designing services applying DDD concepts, Define service communication, Describe the API gateway

Unit III: Spring Framework **10 Lecture Hours**

Advantages of Spring Framework, Spring Modules, IoC Containers, Bean Factory, Spring Boot, Advantages of Spring Boot, Create Projects in Spring Boot, Angular Components, Angular Forms, Services & Dependency, Modules, Advanced Components, Handling Errors, SEO and Angular, Angular CLI

Unit IV: REST and Spring Boot **10 Lecture Hours**

Web Services, Terminology, SOAP, Restful Web Service, SOAP Vs RESTful Web Services, RESTful Web Services with SpringBoot, User Bean and User Service, GET Method, POST Method, Validations, Initialize REST API, Connect Angular Front with RESTful API, HTTP Services, Retrieval, Spring Security, Authentication Service, JWt Framework, JPA and Hibernate

Unit V: Implementing Microservices and Case Studies **5 Lecture Hours**

Best practices for designing and implementing microservices, single responsibility principle, asynchronous communication: loose coupling, fault tolerance, backward compatibility, organizational efficiencies, case studies of real-world microservices architectures (Netflix; Amazon; Uber), future trends and advancements in microservices.

Total lecture Hours 45

Textbooks

1. Sam Newman, "Building Microservices: Designing Fine-Grained Systems", 2nd Edition, O'Reilly Publication, 2021.
2. K. Siva Prasad Reddy, "Spring Boot: Applications and Microservices", APress Publication.

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSFS3107P	Microservices and Spring-Boot Lab		0	0	2	1
Total Units to be Covered: 10		**Total Contact Hours: 30**				
Prerequisite(s):	Elementary knowledge of DevOps		**Syllabus version:** 1.0			

Course Objectives

The student should be able to understand and apply concepts of horizontal scaling techniques and differentiate it from vertical scaling techniques.

Course Outcomes

On completion of this course, the students will be able to

CO1: Develop microservices-based applications.
CO2: Gain the skills to design and implement RESTful APIs
CO3: Understand and implement various communication patterns between microservices.
CO4: Containerize microservices using Docker and effectively deploy them.
CO5: Demonstrate proficiency in using Spring Boot Actuator and other monitoring tools.

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	-	1	1	-	-	-	-	-	-	-	-	-	3
CO 2	1	-	-	1	1	-	-	-	-	-	-	-	-	-	3
CO 3	1	2	-	2	1	-	-	-	-	-	-	-	-	-	3
CO4	1	-	-	3	1	-	-	-	-	-	-	-	-	-	3
CO5	1	-	-	1	1	-	-	-	-	-	-	-	-	-	3
Average	1	0.2	-	1.6	1	-	-	-	-	-	-	-	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Setting Up the Development Environment (Java, Spring Boot, IDE, Docker, etc.)

Experiment 2: Creating a Basic Spring Boot Application

Experiment 3: Building a RESTful API

Experiment 4: Implementing Service Discovery with Eureka

Experiment 5: Load Balancing with Ribbon

Experiment 6: Containerizing Microservices with Docker

Experiment 7: Implementing Event-Driven Communication with Spring Cloud Stream

Experiment 8: Setting Up Centralized Configuration with Spring Cloud Config

Experiment 9: Implementing API Gateway with Spring Cloud Gateway

Experiment 10: Monitoring Microservices with Spring Boot Actuator

Total Lab hours 30

Textbooks

1. K Siva Prasad Reddy and Springerlink (Online Service, Beginning Spring Boot 2 : Applications and Microservices with the Spring Framework. Berkeley, Ca: Apress, 2017.

Reference Books

1. S. Newman, Building Microservices. Beijing: O’reilly, 2015.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Detailed breakup of Internal Assessment

Course Code	Course name	L	T	P	C
CSVT4020P	**Cloud Computing and Security**	4	0	0	4
Total Units to be Covered: 05		**Total Contact Hours: 60**			
Prerequisite(s):	DevOps Fundamentals and SCM	**Syllabus version: 1.0**			

Course Objectives

This course will provide a foundational understanding of what is required to secure a cloud ecosystem, regardless of the vendor. The concepts and principles discussed will help bridge the gaps between traditional and cloud architectures while accounting for the shifting thought patterns involving enterprise risk management. Students who complete this course will enter into any organization utilizing the cloud and immediately bring value to the infrastructure and security teams.

Course Outcomes

On completion of this course, the students will be able to

CO1: Understand core cloud computing concepts and fundamental principles, including standard delivery models and service designs.

CO2: Identity and access management practices of both cloud providers and consumers.

CO3: Design the foundational security practices that are required to secure modern cloud computing infrastructures.

CO4: Implement regulatory requirements needed to secure data in the cloud and the difficulties in meeting those requirements.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	2	-	-	3	-	-	-	1	-	-	-	-	-	-
CO 2	-	3	-	-	2	-	3			-	-	-	-	-	-
CO 3	-	2		-	1	-	-	-	-	2	-	-	-	-	-

CO4	-	-	1	-	2	-	-	-	-	-	-	-	-	-	-
CO5	-	-	2	-	2	-	-	-	2	-	-	2	-	-	-
Average	0.6	1.4	0.6	-	2	-	0.6		0.6	0.4	-	0.4	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Cloud Computing Overview **12 Lecture Hours**

Origin & History of Cloud Computing, Cloud & Cloud Engineering, Evolution, Characteristics of Cloud, Cloud Computing Elements, Types of Cloud Computing, Trends in Cloud Computing, Cloud Service Providers, Traditional IT Providers, Cloud & DevOps, Benefits of Cloud Computing, Properties, Disadvantages, Amazon Web Services(AWS)

Unit II: Cloud Computing Architecture **12 Lecture Hours**

Traditional Vs Cloud Computing Architecture, How Cloud Computing Works, Networking in Cloud, Deployment Models, IaaS (Infrastructure as a Service) & its Resource Virtualization examples, PaaS (Platform as a Service), SaaS (Software as a service), SaaS (Storage as a Service), difference between IaaS, PaaS, & SaaS, Cloud Platform & Management, Service Management in Cloud Computing, Service Level Agreements (SLA's) , Scaling support in cloud / elastic nature, setting budget & notifications for budgets, Global Infrastructure of any cloud in general (Regions, Availability Zones, Data Centres)

Unit III: AWS Basic Architecture **12 Lecture Hours**

Elastic Compute Cloud (EC2), Elastic Block Storage (EBS), Elastic Load Balancing (ELB), Security Groups, Elastic Cache, Amazon RDS, Storage & Backups, Scaling, Amazon Virtual Private Cloud (VPC), Features of VPC, Subnets, Content Delivery Network (CDN using Cloud Front), Features, Cloud Watch, Beanstalk, route 53, S3, Auto Scaling Groups

Unit IV: Cloud Security **12 Lecture Hours**

Security Concepts, Security Planning, Boundaries, Identity and Access Management(IAM), Cloud Security essentials, Cloud Security Alliance, Data Security, Encryption, Isolated Access to Data, Introduction to Key Management Service (KMS) of AWS

Unit V: Role of Cloud in DevOps **12 Lecture Hours**

The Role of Cloud in DevOps, Cloud for successful Ops, Secure Cloud Platforms for DevOps, Cloud & IT Budgets, building a business case for cloud computing, Infrastructure as Code (IAC)

Total lecture Hours 60

Textbooks

1. Tim Mather, Subra Kumaraswamy, and Shahed Latif, "Cloud Security and Privacy: An Enterprise Perspective on Risks and Compliance", O'Reilly Media, 2019.
2. Rajkumar Buyya, James Broberg, and Andrzej Goscinski, "Cloud Computing: Principles and Paradigms", Wiley, 2013.
3. Dan C. Marinescu, "Cloud Computing: Theory and Practice", Morgan Kaufmann, 2013.

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSVT4120P	Cloud Computing and Security Lab	0	0	2	1
Total Units to be Covered: 12	Total Contact Hours: 30				
Prerequisite(s):	Cloud Computing Architecture and Deployment Models	Syllabus version: 1.0			

Course Objectives

1. To empower students to proficiently implement and manage access control strategies across various cloud platforms, ensuring data and resource security.

2. To enable students to effectively design and deploy security measures such as attribute-based access control (ABAC), multi-factor authentication (MFA), and role-based access control (RBAC) to safeguard cloud environments in real-world scenarios.

Course Outcomes

On completion of this course, the students will be able to

CO1: Demonstrate the ability to design and deploy secure cloud infrastructures across various cloud computing platforms.

CO2: Evaluate and compare access control mechanisms across various cloud computing platforms.

CO3: Create and manage customer-managed policies in AWS to enforce security and compliance standards.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	-	1	2	-	-	-	1	-	-	1	-	-	3
CO 2	1	-	-	1	2	-	-	-	1	-	-	1	-	-	3
CO 3	1	-	-	1	2	-	-	-	1	-	-	1	-	-	3
Average	1	-	-	1	2	-	-	-	1	-	-	1	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Building your first Virtual Private Cloud-AWS

Experiment 2: Delegate access across AWS accounts using roles-AWS

Experiment 3: Create a customer managed policy-AWS

Experiment 4: Use attribute-based access control (ABAC)-AWS

Experiment 5: Permit users to manage their credentials and MFA settings (AWS)

Experiment 6: Implement Azure key vault

Experiment 7: Manage Access to Azure with Role-Based Access Control

Experiment 8: Manage Identity and Access- MFA, and Conditional Access (Azure)

Experiment 9: Manage Identity and Access- Implement Directory Synchronization (Azure)

Experiment 10: Smart Access Control on GCP: Web, programmatic, and command-line access

Experiment 11: Write an IAM policy by using client libraries (GCP)

Experiment 12: Using AWS WAF to control access to your content (AWS)

Total Lab hours 30

Textbooks

1. R. L. Krutz, Russell Dean Vines, and J. Wiley, Cloud security : a comprehensive guide to secure cloud computing. New Delhi: Wiley, 2016.
2. T. Mather, S. Kumaraswamy, and S. Latif, Cloud Security and Privacy : an Enterprise Perspective on Risks and Compliance. Sebastopol: O'Reilly Media, Inc., 2009.

Reference Books

I. M. Abbadi, Cloud Management and Security. John Wiley & Sons, 2014.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Detailed breakup of Internal Assessment

Course Code	Course name	L	T	P	C
CSDV4012P	Container Orchestration and Security	4	0	0	4
Total Units to be Covered: 05	Total Contact Hours: 60				
Prerequisite(s):	DevOps Fundamentals and SCM, Basic Understanding of Cloud Computing	Syllabus version: 1.0			

Course Objectives

The Container Orchestration with Kubernetes and Security course will help the students to grasp the key skills, technology, and concepts that a Kubernetes administrator needs to know. Plan to oversee containerized workloads and administrations with industrial Organizations utilizing the Kubernetes Preparing course. It covers all the aspects, including application lifecycle management, installation, configuration and validation, networking, scheduling, security, cluster maintenance, core concepts, Azure Kubernetes, storage, and more.

Course Outcomes

On completion of this course, the students will be able to

CO1: Understand the significance of containers over virtual machines.

CO2: Design the container-based deployment of an application on container-based platform.

CO3: Implement a Kubernetes Cluster and deploy a Dockerized application on a cloud based Kubernetes

cluster.

CO4: Test and Integrate security protocols in containersed application for any vulnerability.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	2	-	-	3	-	-	-	1	-	-	-	-	-	-
CO 2	-	3	-	-	2	-	3	-	-	-	-	-	-	-	-
CO 3	-	2	-	-	1	-	-	-	-	2	-	-	-	-	-
CO4	-	-	1	-	2	-	-	-	-	-	-	-	-	-	-

CO5	-	-	2	-	2	-	-	-	2	-	-	2	-	-	-
Average	0.6	1.4	0.6	-	2	-	0.6	-	0.6	0.4	-	0.4	-	-	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Application Containerization **12 Lecture Hours**

Understanding Containers, Challenges in the Software Industry, Problems in Software Industry Before Containers, Put that in Container! Solution by containers in the Software Industry, Virtualization, Hypervisor, Scope of Virtualization, Containers vs Virtual Machines, Understanding Containers, Containerization Platform, Runtime and Images, Container Platform, Container Runtime, The Chroot System, FreeBSD Jails, Linux Containers (LXC), Docker

Unit II: Introduction to Containerization and different environments 12 Lecture Hours

Docker architecture, Docker Daemon (Container Platform), Docker Rest API, Understanding Different environments: (Dev, QA and Prod), Overcoming issues with different environments, Development Environment, Testing Environment, Staging Environment, Production Environment, Virtual machines for dev/deployments, Containers for dev/deployments, Advantages and drawbacks of containerization

Unit III: Docker Fundamentals and Internals **12 Lecture Hours**

Docker container states, docker image vs docker containers, docker image creation using docker commit & Dockerfile, Dockerfile important keywords, docker tags, persistent storage use-case, docker volumes, docker networks, creating custom networks in docker, docker registry, docker inbuilt security concepts (namespaces, cgroups)

Unit IV: Orchestration Tools **12 Lecture Hours**

What is orchestration, Need of orchestration, Case study: Need of Orchestration, Need of Orchestration: Container and Microservices, Docker Swarm and Kubernetes, Architecture, AWS (ECS,EKS), AWS Elastic Container Services Architecture, Azure Kubernetes Services, OpenShift, KUBERNETES ON CLOUD, Monitoring of container, How to monitor

Unit V: Container Security **12 Lecture Hours**

Docker vs Vagrant, Docker Challenges Revisited, Vulnerabilities in images (Public and Private), Denial of service attacks, Privilege escalation methods in Docker, Security misconfigurations, Container Security, Content Trust and Integrity checks, Capabilities and namespaces in Docker, Segregating Networks, Kernel Hardening using SecComp and AppArmor, Static Analysis of container (Docker) images, Dynamic Analysis of container hosts and daemons.

Total lecture Hours 60

Textbooks

4. Liz Rice, "Container Security Fundamental Technology Concepts that Protect Containerized Applications", O'Reilly Media, 2020.
5. José Manuel Ortega Candel, "DevOps and Containers Security and Monitoring in Docker Containers", BPB Publication, 2020.
6. Gigi Sayfan, "Mastering Kubernetes", 3rd Edition, Packt Publishing, 2020.

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSDV4112P	Container Orchestration and Security Lab		0	0	2	1
Total Units to be Covered: 14	Total Contact Hours: 30					
Prerequisite(s):	DevOps Fundamentals and SCM, Basic Understanding of Cloud Computing	Syllabus version: 1.0				

Course Objectives

The student should be able to understand and apply concepts of horizontal scaling techniques and differentiate it from vertical scaling techniques.

Course Outcomes

On completion of this course, the students will be able to

CO1. Develop and Compare containers with virtual machines.

CO2. Build Docker Images for Docker container.

CO3. Implement Docker Swarm using Orchestration Tools for a large production environment.

CO4. Setup and Use Kubernets Single and Multi-Node cluster and deploy application in different Pods.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	1	3	2	2	-	-	-	2	-	-	1	1	1	3
CO 2	1	2	2	2	3	-	-	-	3	-	-	-	2	2	3
CO 3	2	3	3	3	3	2	-	-	3	-	-	-	2	2	3
CO4	2	3	3	3	3	2	-	-	3	-	-	-	2	2	3
Average	1.5	2.25	2.75	2.5	2.75	2	-	-	2.75	-	-	1	1.75	1.75	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment1 Installing Vagrant & Creating basic vagrant box using VirtualBox virtualization.

Experiment2 Understanding vagrant file - Configuration - CPU, RAM, Storage, Provisioning (Shell Script).

Experiment3 Docker Machine - Installation , configuration, creating machines (on VirtualBox).

Experiment4 Docker - Installation, Configuration, Running Images.

Experiment5 Dockerfile - Containerizing application, Building Images, Tagging, Publishing.

Experiment6 Docker - Volumes, Env, Monitoring (Docker stats).

Experiment7 DTR - Docker Hub, Private Registries, Publishing images.

Experiment8 Docker Compose - Installation, Creating Compose files, Running Images using docker-compose.

Experiment9 Running Multi-Container applications using docker compose and on Swarm.

Experiment10 Kubernetes -Minikube installation and fundamentals.

Experiment11 Deploying Pods and Services on minikube.

Experiment12 Build Docker Image using .Dockerignore file.

Experiment13 Prepare and Implement Docker Container Restart Policy

Experiment14 Working with Metadata, Log File using Docker

Total Lab hours 30

Textbooks

1. Application containerization by Xebia

Reference Books

1. Jarosław Krochmalski, “Developing with Docker”, Packt Publishing, 2016.

2. Adrian Mouat, “Orchestrating, clustering, and managing containers”, O'Reilly Media, Inc., 2016

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50

5. Cyber Security and Forensics Track

Course Code	Course name		L	T	P	C
CSSF2014P	Information Technology and Cyber Security		4	0	0	4
Total Units to be Covered: 05		Total Contact Hours: 60				
Prerequisite(s):	Elementary knowledge of Computer		Syllabus version: 1.0			

Course Objectives

This course on Information Technology and Cyber Security aims to provide students with a comprehensive understanding of fundamental concepts and principles in information security, types of cybercrimes, IT security planning, audit, and compliance, network security and data privacy, as well as physical security. The course objectives include developing knowledge and skills in risk management, security controls and technologies, incident response, legal and ethical considerations, and emerging trends in information security. Students will also gain insights into cybersecurity laws, network attacks, data encryption, secure transmission protocols, and physical security measures. Practical exercises and real-world examples will enhance critical thinking, problem-solving, and communication skills, fostering professionalism and ethical responsibility in the field.

Course Outcomes

On completion of this course, the students will be able to

CO1. Understand the fundamental concepts and principles of information security, including risk management, security controls, and incident response.

CO2. Identify and analyze different types of cyber-crimes, including hacking, malware attacks, identity theft, and financial cyber-crimes, and comprehend the impact of these crimes on individuals and organizations.

CO3. Apply IT security planning, audit, and compliance practices, including security policies and standards, risk assessment, and incident response, to ensure effective management and protection of information technology resources.

CO4. Evaluate and implement network security measures, data privacy principles, and secure transmission protocols to safeguard networks, web applications, and cloud computing environments.

CO5. Recognize the importance of physical security and employ appropriate measures, such as access control systems, video surveillance, and emergency response procedures, to mitigate physical security threats and vulnerabilities.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-
CO 2	-	-	-	1	-	-	-	-	-	-	-	-		3	-
CO 3	-	-	-		-	-	-	-	-	-	-	-	1		-
CO4	-	-	-	1	-	-	-	-	-	-	-	-	-	1	2
CO5	-	-	-	-	-	-	-	-	-	-	-	-	-	1	2
Average	-	-	-	0.4	-	-	-	-	-	-	-	-	0.4	1	0.8

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Overview & Fundamentals of Information Security **12 Lecture Hours**

Introduction to Information Security, Information Security Principles, Security Threats and Vulnerabilities, Risk Management, Security Policies and Procedures, Security Controls and Technologies, Security Incident Response, Legal and Ethical Considerations, Emerging Trends in Information Security.

Unit II: Types of Cyber Crimes **12 Lecture Hours**

Introduction to Cyber Crimes, Hacking and Unauthorized Access, Malware Attacks, Identity Theft and Fraud, Data Breaches and Information Leakage, Online Harassment and Cyberbullying, Financial Cyber Crimes, Cyber Terrorism and State-sponsored Attacks, Cybersecurity Laws and International Cooperation.

Unit III: IT Security Planning, Audit & Compliance **12 Lecture Hours**

Introduction to IT Security Planning, Audit & Compliance, IT Security Planning, IT Security Policies and Standards, IT Security Governance, IT Security Audits, Compliance Management, Security Risk Assessment and Management, Security

Incident Response and Forensics, Security Awareness and Training Programs, Security Metrics and Reporting

Unit IV: Introduction to Network Security & Data Privacy 12 Lecture Hours

Introduction to Network Security & Data Privacy, Network Security Fundamentals, Types of Network Attacks, Network Security Devices and Technologies, Firewalls and Intrusion Detection Systems, Virtual Private Networks (VPNs) and Secure Remote Access, Wireless Network Security, Web Application Security, Data Privacy Principles and Regulations, Data Encryption and Cryptography, Secure Data Transmission Protocols, Data Backup and Disaster Recovery, Privacy Enhancing Technologies, Privacy Policies and Consent Management, Securing Cloud Computing Environments.

Unit V: Introduction to Physical Security 12 Lecture Hours

Introduction to Physical Security, Importance of Physical Security, Physical Security Threats and Vulnerabilities, Risk Assessment and Site Survey, Access Control Systems, Video Surveillance and CCTV Systems, Intrusion Detection and Alarm Systems, Perimeter Security and Fencing, Security Lighting and Signage, Visitor Management and Access Procedures, Locks and Keys, Security Personnel and Guard Services, Emergency Response and Crisis Management, Security Policies and Procedures, Physical Security Audits and Inspections.

Total lecture Hours 60

Textbooks

1. Duane C. Wilson, "Cybersecurity" The MIT Press, 2021.
2. David Alexander, Amanda Finch, David Sutton, and Andy Taylor, "Information Security Management Principles", 2nd Edition, BCS, The Chartered Institute for IT, 2013.

Reference Books

1. Dr. Erdal Ozkaya, "Cybersecurity: The Beginner's Guide: A comprehensive guide to getting started in cybersecurity", Packt publisher, 2019.
2. Charles J. Brooks, Christopher Grow, Philip Craig, and Donald Short, "Cybersecurity Essentials", Sybex, 2018.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSSF2114P	**Information Technology and Cyber Security Lab**	0	0	2	1
Total Units to be Covered: 11	**Total Contact Hours: 30**				
Prerequisite(s):	Elementary knowledge of Computer	**Syllabus version: 1.0**			

Course Objectives

The course objectives of the Information Technology and Cyber Security Lab are to provide students with comprehensive knowledge and practical skills in areas such as system event log analysis, email tracing, live vulnerability assessments, footprinting, search engine hacking, batch file creation, local vulnerability scanning, Kali Linux, network and host scanning, vulnerability scanning with Nessus, and phishing email analysis. Through hands-on experiments, students will develop the ability to identify and address cybersecurity threats, understand the techniques used by attackers, apply mitigation measures, and effectively secure information systems and networks, preparing them for careers in the field of cybersecurity.

Course Outcomes

On completion of this course, the students will be able to

CO1. Proficiency in system event log analysis, enabling students to effectively interpret log files generated by operating systems and applications. Students will develop the ability to identify security incidents, detect anomalies, and understand system behaviors.

CO2. Competence in tracing email sender locations, allowing students to investigate and determine the geographic origin of email communications. Students will gain the skills to analyze email headers, trace email paths, and identify potential malicious activities.

CO3. Mastery of conducting live vulnerability assessments on internet devices, equipping students with the capability to identify and evaluate security weaknesses in real-time. Students will gain hands-on experience in assessing network infrastructure and system vulnerabilities.

CO4. Proficient in basic footprinting and search engine hacking techniques, enabling students to gather information about target systems, networks, and potential vulnerabilities. Students will learn to utilize search engines effectively to identify exposed information and assess potential security risks.

CO5. Proficiency in local vulnerability scanning, utilizing tools and techniques to identify vulnerabilities within a local environment. Students will gain the skills to scan and assess the security posture of systems, networks, and applications within a controlled setting.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-
CO 2	-	-	-	1	-	-	-	-	-	-	-	-	-	3	-
CO 3	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-
CO4	-	-	-	1	-	-	-	-	-	-	-	-	-	1	2
CO5	-	-	-	-	-	-	-	-	-	-	-	-	-	1	2
Average	-	-	-	0.4	-	-	-	-	-	-	-	-	0.4	1	0.8

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1. System Event Logs

Experiment 2. Trace Email Sender Location

Experiment 3. Live Vulnerability on Internet devices

Experiment 4. Basic Footprinting

Experiment 5. Hack using Search Engines

Experiment 6. Create Irritating Batch Files

Experiment 7. Local Vulnerability Scanning

Experiment 8. Kali Linux

Experiment 9. Network & Host Scanning

Experiment 10. Vulnerability Scanning using Nessus

Experiment 11. Phishing Email Analysis.

Total Lab hours 30

Textbooks

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name		L	T	P	C
CSSF3026P	Ethical Hacking & Penetration Testing		4	0	0	4
Total Units to be Covered: 05		Total Contact Hours: 60				
Prerequisite(s):	Information Technology and Cyber Security		Syllabus version: 1.0			

Course Objectives

The objective of the course is to provide students with a comprehensive understanding of the principles and techniques involved in ethical hacking and penetration testing. The course aims to equip students with the knowledge and skills necessary to identify vulnerabilities, exploit weaknesses, and assess the security of computer systems, networks, and applications. Students will learn about various hacking methodologies, tools, and technologies used in ethical hacking and penetration testing. The course will emphasize hands-on practical exercises, simulations, and real-world scenarios to develop the ability to assess, analyze, and secure systems from potential threats and vulnerabilities. Additionally, students will gain insights into legal and ethical considerations in conducting ethical hacking activities and will be prepared to apply their knowledge to protect and secure organizations' digital assets.

Course Outcomes

On completion of this course, the students will be able to

CO1. Understand the principles, methodologies, and techniques of ethical hacking and penetration testing to identify and exploit vulnerabilities in computer systems, networks, and applications.

CO2. Demonstrate proficiency in using various tools, technologies, and frameworks commonly employed in ethical hacking and penetration testing, including reconnaissance, scanning, enumeration, and exploitation.

CO3. Develop critical thinking and problem-solving skills to effectively analyze and assess security vulnerabilities and recommend appropriate countermeasures to enhance system security.

CO4. Apply ethical and legal considerations in conducting ethical hacking activities, adhering to professional codes of conduct, and protecting sensitive information during penetration testing engagements.

CO5. Demonstrate effective communication skills in documenting and presenting findings, risks, and recommendations resulting from ethical hacking and penetration testing activities to stakeholders in a clear and concise manner.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	-	-	-	-	-	-	-	-	1	-	3
CO 2	-	-	-	1	-	-	-	-	-	-	-	-	-	3	1
CO 3	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-
CO4	-	-	-	1	-	-	-	-	-	-	-	-	-	2	-
CO5	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-
Average	-	-	-	0.4	-	-	-	-	-	-	-	-	0.4	1.4	0.8

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

Syllabus

Unit I: Introduction to Ethical Hacking **10 Lecture Hours**

Introduction to Ethical Hacking, Ethical Hacking vs. Unethical Hacking, Legal and Ethical Considerations, Phases of Ethical Hacking, Footprinting and Reconnaissance, Scanning and Enumeration, Vulnerability Assessment and Penetration Testing, System Hacking and Exploitation, Web Application Security Testing, Wireless Network Security Testing, Social Engineering Techniques, Physical Security Assessments, Security Tools, and Technologies, Reporting and Remediation.

Unit II: Setting up Ethical Hacking Labs & Targets **12 Lecture Hours**

Setting up Ethical Hacking Labs & Targets, Understanding Lab Requirements, Virtualization Technologies for Hacking Labs, Building a Virtual Lab Environment, Configuring Network Segmentation, Creating Target Machines and Vulnerable Systems, Lab Network Design Considerations, Lab Setup for Web Application Security Testing, Lab Setup for Wireless Network Security Testing, Lab Setup for System and Network Exploitation, Lab Setup for Social Engineering Exercises, Lab Maintenance and Security Best Practices, Lab Documentation and Reporting.

Unit III: Vulnerability Assessment & Pen Testing Basics **14 Lecture Hours**

Vulnerability Assessment & Pen Testing Basics, Introduction to Vulnerability Assessment, Importance and Benefits of Penetration Testing, Types of Vulnerability Assessments, Scoping and Planning a Penetration Test, Pre-engagement Activities, Information Gathering and Reconnaissance, Vulnerability Scanning and Assessment, Exploitation and Post-Exploitation Techniques, Web Application Penetration Testing, Network Penetration Testing, Wireless Network Penetration Testing, Social Engineering Testing, Report Generation and Recommendations.

Unit IV: Overview of various VA/PT Tools **12 Lecture Hours**

Overview of various VA/PT Tools, Nessus, OpenVAS, NMAP, Qualys, Burp Suite, Acunetix, Metasploit, Wireshark, OWASP ZAP, Nikto, SQLMap, DirBuster, BeEF, Hydra, John the Ripper, Aircrack-ng, THC-Hydra, Hashcat, Armitage.

Unit V: Unit-5 Scanning, Enumeration, Attacking **12 Lecture Hours**

Scanning, Enumeration, Attacking, Port Scanning Techniques, Network and Host Discovery, Vulnerability Scanning, Service Enumeration, Banner Grabbing, Operating System Fingerprinting, Web Application Scanning, DNS Enumeration, SMTP Enumeration, SMB Enumeration, SNMP Enumeration, LDAP Enumeration, Exploitation Techniques, Brute-Force Attacks, Password Cracking, Remote Code Execution, Privilege Escalation, Denial of Service Attacks.

Total lecture Hours 60

Textbooks

1. Jon Erickson, “Hacking: The Art of Exploitation”, 2nd Edition, No Starch Press, US, 2008
2. Patrick Engebretson, “The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made Easy”, 2nd Edition, Syngress, 2013.

Reference Books

1. Rafay Baloch, “Ethical Hacking and Penetration Testing Guide”, Auerbach Publications, 2014.
2. Keshav Kaushik, and Akashdeep Bhardwaj, “Perspectives on Ethical Hacking and Penetration Testing”, IGI Global, 2023.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSSF3110P	Ethical Hacking & Penetration Testing Lab		0	0	2	1
Total Units to be Covered: 10		Total Contact Hours: 30				
Prerequisite(s):	Information Technology and Cyber Security		Syllabus version: 1.0			

Course Objectives

The Ethical Hacking & Penetration Testing Lab aims to achieve several objectives that equip students with essential knowledge and practical skills in the field of ethical hacking and penetration testing. First and foremost, the course aims to provide students with a comprehensive understanding of the principles and methodologies involved in ethical hacking, emphasizing the importance of conducting authorized and legal activities to identify and address security vulnerabilities. Throughout the lab, students will develop practical skills in footprinting, which involves gathering information about target systems, networks, and potential attack vectors. They will explore various techniques and tools, including the use of SHODAN and other footprinting tools, to gather in-depth intelligence about a target's infrastructure and potential weaknesses. Google hacking techniques will also be covered to identify sensitive information and vulnerabilities exposed through misconfigurations or other security weaknesses.

The course will further focus on network scanning using NMAP, enabling students to identify open ports, active hosts, and services running on target systems. This skill is crucial in assessing the security posture of a network and detecting potential vulnerabilities. Students will gain hands-on experience with NMAP, learning how to perform comprehensive network reconnaissance. Vulnerability assessment is another vital objective of the lab. Students will utilize industry-standard tools and techniques to identify security weaknesses and evaluate their severity. They will develop the ability to provide meaningful recommendations for mitigating vulnerabilities, emphasizing the importance of effective security measures.

In addition, the lab will provide students with practical experience in setting up an Ethical Hacking & Penetration Testing (EHPT) lab environment. This includes configuring virtual machines and networking components, ensuring students have the necessary skills to create a controlled and secure testing environment for ethical hacking and penetration testing exercises. Students will also have the opportunity to set up Damn Vulnerable Web Application (DVWA), a deliberately vulnerable web application, and exploit common web application vulnerabilities such as SQL injection, cross-site scripting (XSS), and command injection. This hands-on exercise enhances students' understanding of real-world security weaknesses and their ability to identify and exploit them.

Throughout the lab, students will develop critical thinking and problem-solving skills by applying ethical hacking and penetration testing techniques in a controlled environment. They will learn to identify vulnerabilities, exploit weaknesses, and recommend effective security measures to enhance the overall security of computer networks and systems. Ethical and legal considerations in the field of ethical hacking and penetration testing will also be emphasized. Students will gain knowledge about ethical standards, privacy concerns, and the importance of conducting activities within legal boundaries. By the end of the course, students will have acquired a well-rounded skill set in ethical hacking and penetration testing. They will possess the knowledge and practical skills necessary to identify vulnerabilities, conduct penetration tests, and provide effective recommendations for enhancing cybersecurity. The hands-on approach of the lab will prepare students for real-world scenarios, enabling them to contribute to securing computer networks and systems in a professional and ethical manner.

Course Outcomes

On completion of this course, the students will be able to

CO1. Proficiency in conducting footprinting activities, including utilizing various techniques and tools to gather comprehensive information about target systems, networks, and potential attack vectors. Students will be able to assess the security posture of a target and identify potential vulnerabilities.

CO2. Mastery of network scanning techniques using NMAP, enabling students to identify open ports, active hosts, and services running on target systems. Students will possess the ability to perform comprehensive network reconnaissance, essential for detecting potential vulnerabilities and assessing network security.

CO3. Competence in vulnerability assessment, including the use of industry-standard tools and techniques to identify and evaluate security weaknesses. Students will be capable of providing meaningful recommendations for mitigating vulnerabilities and enhancing overall security.

CO4. Proficient in setting up an Ethical Hacking & Penetration Testing (EHPT) lab environment, including configuring virtual machines and networking components. Students will gain hands-on experience in creating a controlled and secure testing environment for conducting ethical hacking and penetration testing exercises.

CO5. Applied knowledge in exploiting common web application vulnerabilities by setting up and exploiting Damn Vulnerable Web Application (DVWA). Students will be able to identify and exploit vulnerabilities such as SQL injection, cross-site scripting (XSS), and command injection, enhancing their understanding of real-world security weaknesses and their ability to assess and secure web applications.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	-	-	-	-	-	-	-	-	1	-	3
CO 2	-	-	-	1	-	-	-	-	-	-	-	-	-	3	1
CO 3	-	-	-	-	-	-	-	-	-	-	-	-	1	-	-
CO4	-	-	-	1	-	-	-	-	-	-	-	-	-	2	-
CO5	-	-	-	-	-	-	-	-	-	-	-	-	-	2	-
Average	-	-	-	0.4	-	-	-	-	-	-	-	-	0.4	1.4	0.8

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

List of Experiments

Experiment 1 Footprinting Basics

Experiment 2	Footprinting using SHODAN
Experiment 3	Footprinting Tools
Experiment 4	Advanced Footprinting
Experiment 5	Google Hacking
Experiment 6	Scanning using NMAP
Experiment 7	Vulnerability Assessment
Experiment 8	EHPT Lab Setup
Experiment 9	DVWA Setup
Experiment 10	Exploit DVWA

Total Lab hours 30

Textbooks

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name		L	T	P	C
CSSF3027P	Network Security		4	0	0	4
Total Units to be Covered: 08		Total Contact Hours: 60				
Prerequisite(s):	Information Technology and Cyber Security	Syllabus version: 1.0				

Course Objectives

This course aims to provide students with a comprehensive understanding of the principles, technologies, and best practices involved in securing computer networks. The course objectives are to equip students with the necessary knowledge and skills to identify and mitigate network security threats, implement appropriate network security controls and technologies, and respond effectively to network security incidents. Students will learn about the importance and challenges of network security, the goals and objectives of network security, and the common threats and vulnerabilities faced by networks. They will explore network protocols, architecture, and models, gaining an understanding of network devices, components, and segmentation techniques. The course will cover network security technologies and tools such as firewalls, intrusion detection/prevention systems (IDS/IPS), virtual private networks (VPNs), and network access control (NAC). Students will also learn about network access control and authentication methods, wireless network security, and guest access management. Additionally, the course will cover topics like firewall and IDS/IPS systems, VPNs and secure remote access, network security best practices, incident response, and forensics in network security. By the end of the course, students will be proficient in designing, implementing, and maintaining secure network environments, effectively mitigating network security risks, and responding to network security incidents in a timely and efficient manner.

Course Outcomes

On completion of this course, the students will be able to

CO1. Understand the fundamental concepts and principles of network security, including the importance and challenges of securing computer networks, as well as the goals and objectives of network security.

CO2. Identify and assess common network threats and vulnerabilities, including malware and virus attacks, denial of service (DoS) attacks, network intrusions, and unauthorized access, and implement appropriate measures to mitigate these risks.

CO3. Demonstrate proficiency in deploying and configuring network security technologies and tools, such as firewalls, intrusion detection/prevention systems (IDS/IPS), virtual private networks (VPNs), and network access control (NAC), to protect network infrastructure and ensure secure network communications.

CO4. Apply authentication methods and access control models to manage network user authentication and authorization, including password-based, biometric, and token-based authentication, as well as role-based access control (RBAC), mandatory access control (MAC), and discretionary access control (DAC).

CO5. Develop the knowledge and skills to implement network security best practices, including the establishment of security policies and procedures, network hardening techniques, security auditing, compliance management, and incident response procedures. Students will also gain an understanding of network forensics and its role in investigating network security incidents.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	-	-	-	-	-	-	-	-	1		3
CO 2	-	-	-	1	-	-	-	-	-	-	-	-		3	2
CO 3	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-
CO4	-	-	-	1	-	-	-	-	-	-	-	-	-	2	-
CO5	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-
Average	-	-	-	0.4	-	-	-	-	-	-	-	-	0.6	1.2	1

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to Network Security **6 Lecture Hours**

Overview of Network Security, Importance and Challenges of Network Security, Goals and Objectives of Network Security, Network Security Threat Landscape, Network Security Controls and Technologies.

Unit II: Network Protocols and Architecture **8 Lecture Hours**

Introduction to Network Protocols, TCP/IP Protocol Suite, Network Architecture and Models (OSI, TCP/IP), Network Devices and Components, Network Segmentation and Zones.

Unit III: Network Threats and Vulnerabilities **8 Lecture Hours**

Types of Network Threats, Common Network Vulnerabilities, Malware and Virus Attacks, Denial of Service (DoS) Attacks, Network Intrusions and Unauthorized Access.

Unit IV: Network Security Technologies and Tools **8 Lecture Hours**

Firewalls and Intrusion Detection/Prevention Systems (IDS/IPS), Virtual Private Networks (VPNs), Network Access Control (NAC), Secure Sockets Layer/Transport Layer Security (SSL/TLS), Network Security Appliances.

Unit V: Network Access Control and Authentication **8 Lecture Hours**

Authentication Methods (Passwords, Biometrics, Tokens), Access Control Models (RBAC, MAC, DAC), Network Access Policies, Wireless Network Security, Guest Access Management.

Unit VI: Firewall and Intrusion Detection/Prevention Systems **8 Lecture Hours**

Introduction to Firewalls, Firewall Types (Packet Filtering, Stateful Inspection, Application-Level), Intrusion Detection Systems (IDS), Intrusion Prevention Systems (IPS), Firewall Rule Management.

Unit VII: Virtual Private Networks (VPNs) and Secure Remote Access **8 Lecture Hours**

VPN Fundamentals, VPN Protocols (IPSec, SSL/TLS), Site-to-Site VPNs, Remote Access VPNs, Secure Remote Access Technologies (RDP, SSH), VPN Configuration and Troubleshooting.

Unit VIII: Network Security Best Practices and Incident Response **6 Lecture Hours**

Security Policies and Procedures, Network Hardening Techniques, Security Auditing and Compliance, Incident Response and Handling, Forensics in Network Security.

Total lecture Hours 60

Textbooks

1. William Stallings, "Cryptography and Network Security - Principles and Practice", 7th Edition, Pearson, 2017.
2. William Stallings, "Network Security Essentials", 6th Edition, Pearson, 2018.

Reference Books

1. E Cole, "Network Security Bible", 2nd Edition, John Wiley & Sons Inc, 2009.
2. Atul Kahate, "Cryptography And Network Security", 4th Edition, McGraw-Hill, 2019.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSSF3127P	Network Security Lab		0	0	2	1
Total Units to be Covered: 10		Total Contact Hours: 30				
Prerequisite(s):	Information Technology and Cyber Security		Syllabus version: 1.0			

Course Objectives

The course objectives of the Network Security Lab are to provide students with comprehensive knowledge and practical skills in troubleshooting networks, analyzing network packets, tracing network queries, analyzing HTTP traffic, decrypting SSL traffic, cracking passwords of various file types, conducting TCP, ICMP, FTP, and UDP analysis, and analyzing DHCP and DNS protocols. Through hands-on experiments, students will develop the ability to identify network vulnerabilities, detect and analyze network-based attacks, and implement effective security measures to protect computer networks from potential threats, preparing them for careers in the field of network security.

Course Outcomes

On completion of this course, the students will be able to

CO1. Proficiency in troubleshooting network issues without relying on tools, enabling students to identify and resolve network problems effectively and efficiently.

CO2. Mastery of network packet sniffing, capture, and analysis techniques, equipping students with the skills to examine network traffic, detect anomalies, and identify potential security breaches.

CO3. Competence in tracing network queries and analyzing HTTP traffic, enabling students to investigate network activities, track information flow, and identify potential vulnerabilities or malicious activities.

CO4. Proficient in decrypting SSL traffic and conducting password cracking for various file types, providing students with the ability to uncover encrypted communications, assess security weaknesses, and gain unauthorized access for educational purposes.

CO5. Mastery of TCP, ICMP, FTP, UDP, DHCP, and DNS analysis, allowing students to understand the behavior of network protocols, detect anomalies, and identify potential security risks or attacks.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	-	-	-	-	-	-	-	-	1	-	3
CO 2	-	-	-	1	-	-	-	-	-	-	-	-	-	3	2
CO 3	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-
CO4	-	-	-	1	-	-	-	-	-	-	-	-	-	2	-
CO5	-	-	-	-	-	-	-	-	-	-	-	-	-	1	-
Average	-	-	-	0.4	-	-	-	-	-	-	-	-	0.6	1.2	1

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1	Troubleshoot Network without Tools
Experiment 2	Network Packet Sniffing, Capture & Analysis
Experiment 3	Trace network queries
Experiment 4	HTTP Traffic tracing
Experiment 5	Decrypt SSL Traffic
Experiment 6	Crack Passwords of ZIP, RAR, PDF, Shadow files
Experiment 7	TCP Analysis
Experiment 8	ICMP & FTP Analysis
Experiment 9	UDP Analysis
Experiment 10	DHCP & DNS Analysis

Total Lab hours 30

Textbooks

1. William Stallings, "Cryptography and Network Security - Principles and Practice", 7th Edition, Pearson, 2017.
2. William Stallings, "Network Security Essentials", 6th Edition, Pearson, 2018.

Reference Books

1. E Cole, "Network Security Bible", 2nd Edition, John Wiley & Sons Inc, 2009.
2. Atul Kahate, "Cryptography And Network Security", 4th Edition, McGraw-Hill, 2019.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name		L	T	P	C
CSSF4015P	Digital Forensic		4	0	0	4
Total Units to be Covered: 10		Total Contact Hours: 60				
Prerequisite(s):	Ethical Hacking & Penetration Testing		Syllabus version: 1.0			

Course Objectives

This course aims to provide students with a comprehensive understanding of the principles, methodologies, and techniques involved in digital forensics investigations. The course objectives are to equip students with the necessary knowledge and skills to conduct digital forensic examinations, preserve and collect digital evidence, analyze various digital artifacts, and prepare clear and concise forensic reports. Students will learn about the fundamental concepts of digital forensics, including the forensic process, evidence preservation, and acquisition techniques. They will gain expertise in analyzing file systems, operating systems, network traffic, mobile devices, and memory. The course will cover essential technical concepts such as disk imaging, hashing, metadata analysis, volatile memory forensics, network protocols, and encryption. Students will also explore the requirements for setting up a computer forensics lab, including hardware, software, storage, and lab security. Legal and ethical considerations in digital forensics, anti-forensics techniques, and report writing skills will be emphasized throughout the course. By the end of the course, students will be proficient in conducting digital investigations, handling digital evidence, and producing professional forensic reports in compliance with legal and ethical standards.

Course Outcomes

On completion of this course, the students will be able to

CO1. Understand the fundamental concepts and principles of digital forensics, including the forensic process, evidence preservation, and legal and ethical considerations, enabling students to conduct digital investigations in a professional and responsible manner.

CO2. Gain proficiency in digital evidence collection, preservation, and analysis techniques, including acquiring and imaging digital devices, analyzing file systems, operating systems, network traffic, mobile devices, and memory.

CO3. Develop skills in utilizing forensic tools and software to analyze digital artifacts, including file metadata, email communications, web browser activities, and social media interactions, in order to extract valuable information for investigative purposes.

CO4. Apply essential technical concepts in digital forensics, such as disk imaging, hashing, volatile memory forensics, network protocols, and encryption, to effectively identify and interpret digital evidence, establish timelines, and reconstruct events.

CO5. Demonstrate the ability to produce professional forensic reports, adhering to industry standards, and effectively communicate findings and conclusions in a clear and concise manner. Students will also be aware of anti-forensics techniques and be able to detect and counteract these attempts to thwart digital forensic investigations.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-
CO 2	-	-	-	1	-	-	-	-	-	-	-	-	-	3	2
CO 3	-	-	-	-	-	-	-	-	-	-	-	-	3	-	-
CO4	-	-	-	1	-	-	-	-	-	-	-	-	-	1	1
CO5	-	-	-	-	-	-	-	-	-	-	-	-	-	2	1
Average	-	-	-	0.4	-	-	-	-	-	-	-	-	1	1.2	0.8

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

Syllabus

Unit I: Introduction to Digital Forensics **6 Lecture Hours**

Digital Forensics Fundamentals, Introduction to Digital Forensics, Digital Forensics Process and Methodology, Preservation and Collection of Digital Evidence, Acquisition and Imaging Techniques, Forensic Analysis of File Systems, Forensic Analysis of Operating Systems, Network Forensics and Traffic Analysis, Mobile Device Forensics, Memory Forensics, Malware Analysis and Reverse Engineering, Steganography and Cryptography in Digital Forensics, Incident Response and Forensics, Legal and Ethical Considerations in Digital Forensics.

Unit II: Essential Technical Concepts **6 Lecture Hours**

Essential Technical Concepts for Digital Forensics, File Systems and Data Storage Concepts, Disk Imaging and Acquisition, File Metadata and Timestamp Analysis, File Carving and Data Recovery, Hashing and Digital Signatures, Volatile Memory Forensics, Network Forensics, Email Forensics, Mobile Device Forensics, Cloud Forensics, Steganography and Anti-Forensics Techniques, Forensic Analysis of Malware, Log Analysis and Event Reconstruction, Encryption and Decryption Techniques, Data Hiding and File Steganography, Network Protocols and Packet Analysis, Forensic Tools and Software, Evidence Handling and Chain of Custody, Legal and Ethical Considerations in Digital Forensics.

Unit III: Hard Disks and File Systems **6 Lecture Hours**

Hard Disks and File Systems, Introduction to Hard Disks, Hard Disk Architecture and Components, Hard Disk Interfaces (IDE, SATA, SCSI, NVMe), Hard Disk Partitioning and Formatting, File System Basics, FAT File System, NTFS File System, Ext File Systems (Ext2, Ext3, Ext4), HFS+ File System, APFS File System, File System Journaling and Recovery, Disk Partitioning Schemes (MBR, GPT), Disk Imaging and Cloning, Disk Maintenance and Optimization, Disk Encryption and Data Protection, Bad Sectors and Disk Repair, File System Forensics, Recovering Deleted Files and File Metadata, File System Analysis Tools, Disk and File System Forensics Techniques, Best Practices for Hard Disk and File System Security.

Unit IV: Requirements for a Computer Forensics Lab **6 Lecture Hours**

Requirements for a Computer Forensics Lab, Introduction to Computer Forensics Lab, Hardware Requirements for a Forensics Lab, Workstation Setup and Configuration, Storage and Backup Solutions, Network Infrastructure for a Forensics Lab, Forensic Imaging and Acquisition Tools, Forensic Analysis and Investigation Tools, Forensic Software and Applications, Virtualization for Forensic Analysis, Lab Security and Access Control, Forensic Lab Documentation and Procedures, Lab Environment and Ergonomics, Data Integrity and Chain of Custody, Lab Certification and Accreditation, Training and Skill Development, Lab Budgeting and Resource Management, Emerging Trends in Computer Forensics Labs.

Unit V: Acquiring Digital Evidence **6 Lecture Hours**

Acquiring Digital Evidence, Introduction to Digital Evidence Acquisition, Legal Considerations and Chain of Custody, Types of Digital Evidence, Volatile and Non-Volatile Data, Acquiring Data from Hard Disks and Storage Devices, Imaging and Hashing Techniques, Live Data Acquisition, Network Traffic Capture and Analysis, Acquiring Data from Mobile Devices, Forensic Acquisition of Cloud Data, Remote Acquisition Techniques, Memory Acquisition and Analysis, Data Extraction from Embedded Systems, Data Acquisition from IoT Devices, Acquisition of Social Media and Web Content, Forensic Imaging of Virtual Machines, Challenges and Best Practices in Digital Evidence Acquisition.

Unit VI: Analysis of Digital Evidence **6 Lecture Hours**

Analysis of Digital Evidence, Introduction to Digital Evidence Analysis, Preservation and Validation of Digital Evidence, Digital Forensic Tools and Software, File and Metadata Analysis, Email and Messaging Analysis, Web and Social Media Analysis, Network Traffic Analysis, Malware Analysis and Reverse Engineering, Memory Analysis, Mobile Device Analysis, Cloud Data Analysis, Multimedia and Image Forensics, Steganalysis, Database and Registry Analysis, Log File Analysis, Timeline Analysis, Link and Association Analysis, Data Recovery and Reconstruction Techniques, Reporting and Presentation of Digital Evidence, Challenges and Best Practices in Digital Evidence Analysis.

Unit VII: Windows Forensic Analysis **6 Lecture Hours**

Windows Forensic Analysis, Introduction to Windows Forensics, Windows File Systems (FAT, NTFS), Windows Registry Analysis, Windows Event Logs Analysis, User Account and Authentication Analysis, Windows Artifact Analysis (Recent Files, Jump Lists, Prefetch), Windows Timeline Analysis, Internet Browser Forensics (Internet Explorer, Microsoft Edge, Chrome, Firefox), Email Client Forensics (Outlook, Thunderbird), Windows Memory Analysis, Windows Malware Analysis, Windows Network Forensics, Windows Shellbags Analysis, Windows Link File Analysis, Windows Forensic Tools and Software, Anti-Forensics Techniques on Windows, Reporting and Presentation of Windows Forensic Findings.

Unit VIII: Web Browser and E-mail Forensics **6 Lecture Hours**

Web Browser and E-mail Forensics, Introduction to Web Browser Forensics, Web Browser Architecture and Components, Web Browser Forensic Artifacts, Internet History Analysis, Bookmark and Favorites Analysis, Cookie Analysis, Downloaded Files and Cache Analysis, Form Data and Autofill Analysis, Webmail Forensics, E-mail Header Analysis, E-mail Content and Attachments Analysis, E-mail Metadata Analysis, Web Browser and E-mail Forensic Tools, Extracting Evidence from Web Browsers and E-mail Clients, Analyzing Webmail Services (Gmail, Yahoo Mail, Outlook.com), Investigating Social Media Activities, E-mail Spoofing and Phishing Investigations, Recovering Deleted E-mails and Web Browser History, Reporting and Presentation of Web Browser and E-mail Forensic Findings.

Unit IX: E-mail Forensics **6 Lecture Hours**

E-mail Forensics, Introduction to E-mail Forensics, E-mail Headers and Metadata Analysis, E-mail Content and Attachments Analysis, E-mail Client Artifacts, E-mail Tracking and Tracing, E-mail Spoofing and Phishing Analysis, E-mail Authentication and Validation, E-mail Recovery and Reconstruction Techniques, Deleted E-mail Recovery, E-mail Encryption and Decryption, E-mail Timestamp Analysis, E-mail Forensic Tools and Software, E-mail Forensics in Webmail Services (Gmail, Yahoo Mail, Outlook.com), E-mail Forensics in Desktop Clients (Outlook, Thunderbird), E-mail Forensics in Mobile Devices, E-mail Forensics in Exchange Servers, Reporting and Presentation of E-mail Forensic Findings.

Unit X: Anti-Forensics Techniques and Report Writing 6 Lecture Hours

Anti-Forensics Techniques and Report Writing, Introduction to Anti-Forensics Techniques, Data Destruction and Deletion, Data Encryption and Steganography, Data Hiding and File Manipulation, Anti-Forensics in Network Communications, Anti-Forensics in Web Browsing and E-mails, Anti-Forensics in Mobile Devices, Anti-Forensics in Cloud Services, Anti-Forensics Countermeasures, Detecting and Overcoming Anti-Forensics Techniques, Digital Forensics Reporting, Elements of a Forensic Report, Report Structure and Formatting, Writing Clear and Concise Findings, Case Documentation and Evidence Presentation, Legal and Ethical Considerations in Report Writing, Peer Review and Quality Assurance in Report Writing.

Total lecture Hours 60

Textbooks

1. Dr. Akashdeep Bhardwaj and Keshav Kaushik, “Practical Digital Forensics”, BPB publisher, 2023.
2. J. Bayuk, “CyberForensics: Understanding information security investigations”, Springer Science & Business Media, 2010.

Reference Books

1. E. Casey, “Handbook of digital forensics and investigation”, Academic Press, 2009.
2. E. Casey, “Digital evidence and computer crime: Forensic science, computers and the internet”, Academic Press, 2011.
3. EC-Council, “Computer forensics: Investigating network intrusions and cybercrime (CHFI)”, Cengage Learning, 2016.
4. T. J. Holt, A. M. Bossler, and K. C. Seigfried-Spellar, “Cybercrime and digital forensics: An introduction”, Routledge, 2015.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSSF4115P	Digital Forensics Lab		0	0	2	1
Total Units to be Covered: 10		Total Contact Hours: 30				
Prerequisite(s):	Ethical Hacking & Penetration Testing		Syllabus version: 1.0			

Course Objectives

The course objectives of the Digital Forensics Lab are to provide students with practical experience in digital forensics by equipping them with the knowledge and skills to create forensic images, capture and analyze network packets, acquire and analyze data using forensic tools, utilize Kali Linux for forensic investigations, investigate email communications, perform memory forensics, conduct dynamic and static malware analysis, explore steganography techniques, and understand digital watermarking. Through hands-on activities and exercises, students will develop a comprehensive understanding of digital forensics principles, techniques, and tools, enabling them to investigate cybercrimes, analyze digital evidence, and contribute to the prevention and resolution of cybersecurity incidents.

Course Outcomes

On completion of this course, the students will be able to

CO1. Demonstrate proficiency in creating forensic images using FTK Imager, capturing and analyzing network packets using Wireshark, and acquiring and analyzing data using Autopsy Sleuthkit, showcasing the ability to effectively gather and preserve digital evidence for forensic investigations.

CO2. Apply advanced techniques and tools in digital forensics using Kali Linux, showcasing the ability to identify and analyze digital artifacts, recover deleted information, and investigate potential security breaches or cybercrimes.

CO3. Develop expertise in email forensics, including analyzing email headers, tracing email origins, and extracting relevant evidence, demonstrating the capability to investigate and understand the digital communication trail in forensic investigations.

CO4. Utilize memory forensics techniques to extract volatile information from computer memory, identify hidden artifacts, and analyze potential malicious activity, highlighting the ability to uncover valuable evidence in volatile digital environments.

CO5. Perform dynamic and static malware analysis, understanding the behavior and characteristics of malware, and effectively analyze malware samples in controlled environments, showcasing the capability to identify, classify, and mitigate potential cybersecurity threats.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-
CO 2	-	-	-	1	-	-	-	-	-	-	-	-	-	3	2
CO 3	-	-	-	-	-	-	-	-	-	-	-	-	3	-	-
CO4	-	-	-	1	-	-	-	-	-	-	-	-	-	1	1
CO5	-	-	-	-	-	-	-	-	-	-	-	-	-	2	1
Average	-	-	-	0.4	-	-	-	-	-	-	-	-	1	1.2	0.8

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

List of Experiments

Experiment 1 Creating a Forensic Image using FTK Imager
Experiment 2 Capturing and analyzing network packets using Wireshark
Experiment 3 Data Acquisition using Autopsy Sleuthkit
Experiment 4 Forensics using Kali Linux
Experiment 5 Email Forensics
Experiment 6 Memory Forensics
Experiment 7 Malware Analysis – Dynamic
Experiment 8 Malware Analysis – Static
Experiment 9 Steganography
Experiment 10 Watermarking.

Total Lab hours 30

Textbooks

1. Dr. Akashdeep Bhardwaj and Keshav Kaushik, “Practical Digital Forensics”, BPB publisher, 2023.
2. J. Bayuk, “CyberForensics: Understanding information security investigations”, Springer Science & Business Media, 2010.

Reference Books

1. E. Casey, “Handbook of digital forensics and investigation”, Academic Press, 2009.
2. E. Casey, “Digital evidence and computer crime: Forensic science, computers and the internet”, Academic Press, 2011.
3. EC-Council, “Computer forensics: Investigating network intrusions and cybercrime (CHFI)”, Cengage Learning, 2016.
4. T. J. Holt, A. M. Bossler, and K. C. Seigfried-Spellar, “Cybercrime and digital forensics: An introduction”, Routledge, 2015.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSSF4017P	**OS, Application & Cloud Security**	3	0	0	3
Total Units to be Covered: 05	**Total Contact Hours: 45**				
Prerequisite(s):	Ethical Hacking & Penetration Testing	**Syllabus version: 1.0**			

Course Objectives

The objective of this course is to provide students with a comprehensive understanding of the concepts, techniques, and best practices related to securing operating systems, applications, and cloud environments. The course aims to equip students with the knowledge and skills necessary to effectively administer and troubleshoot operating systems, manage user access and permissions, monitor system performance, and implement security configurations. Students will also learn about open-source intelligence (OSINT) and its methodologies, tools, and ethical considerations. The course further covers the fundamentals of database security, including authentication, access control, encryption, auditing, and incident response. Additionally, students will explore application security principles, secure software development practices, common vulnerabilities, and security testing techniques. The course will conclude with an examination of virtualization and cloud security, including virtual machine and hypervisor security, cloud deployment models, identity and access management, data security and privacy, incident response, and best practices in cloud security. By the end of the course, students will have the necessary knowledge and skills to effectively secure and protect operating systems, applications, and cloud environments against potential threats and vulnerabilities.

Course Outcomes

On completion of this course, the students will be able to

CO1. Understand the principles, methodologies, and techniques involved in administering and troubleshooting operating systems, including user management, file system management, process management, and system performance monitoring.

CO2. Develop an understanding of open-source intelligence (OSINT) methodologies, tools, and ethical considerations, enabling students to gather and analyze information from various sources for security and threat intelligence purposes.

CO3. Evaluate and implement database security measures, including authentication, access control, encryption, auditing, backup and recovery strategies, and vulnerability management, to protect sensitive data and ensure compliance with security standards and regulations.

CO4. Apply secure software development principles, including secure coding practices, threat modeling, input validation, authentication, access control, and error handling, to develop and deploy secure applications. Students will also gain proficiency in conducting security testing and code reviews to identify and remediate vulnerabilities.

CO5. Understand the fundamentals of virtualization and cloud computing, including virtual machine and hypervisor security, cloud deployment models, identity and access management, data security and privacy, and incident response, to effectively address security challenges and risks in virtualized and cloud environments.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	-	-	-	-	-	-	-	-	2	-	2
CO 2	-	-	-	1	-	-	-	-	-	-	-	-	-	3	
CO 3	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-
CO4	-	-	-	1	-	-	-	-	-	-	-	-	-	2	2
CO5	-	-	-	-	-	-	-	-	-	-	-	-	-	1	1
Average	-	-	-	0.4	-	-	-	-	-	-	-	-	0.8	1.2	1

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Administration & Troubleshooting of OS **9 Lecture Hours**

Administration & Troubleshooting of OS, User Management and Access Control, File System Management, Process Management, System Performance Monitoring, System Logs and Event Viewer, Software Installation and Updates, Patch Management, Disk Management and Storage, Network Configuration and Troubleshooting, Device Drivers and Hardware Management, Backup and Recovery Strategies, OS Security Configuration, Troubleshooting Common OS Issues, System Maintenance and Optimization.

Unit II: Open-Source Intelligence **9 Lecture Hours**

Open-Source Intelligence, Introduction to Open Source Intelligence (OSINT), OSINT Methodology and Frameworks, Information Sources and Tools, Search Engines and Advanced Search Techniques, Social Media Intelligence (SOCMINT), Online Forums and Communities, Public Records and Government Sources, Web Scraping and Data Mining, Geolocation and Mapping Tools, Metadata Analysis, Image and Video Analysis, Dark Web Intelligence, OSINT for Threat Intelligence, Ethical and Legal Considerations in OSINT.

Unit III: Database Security **9 Lecture Hours**

Database Security, Introduction to Database Security, Database Security Threats and Risks, Database Authentication and Authorization, Access Control and Privilege Management, Data Encryption and Masking, Secure Database Design and Configuration, Database Activity Monitoring and Auditing, Database Backup and Recovery, Database Patching and Vulnerability Management, Database Security Best Practices, Database Security Standards and Regulations, Database Security Testing and Assessment, Database Security Controls and Tools, Database Security Incident Response.

Unit IV: Application Security **9 Lecture Hours**

Application Security, Introduction to Application Security, Secure Software Development Lifecycle (SDLC), Threat Modeling and Risk Assessment, Secure Coding Practices, Input Validation and Data Sanitization, Authentication and Session

Management, Access Control and Authorization, Cryptography and Key Management, Error Handling and Logging, Secure Configuration Management, Secure File and Resource Handling, Database Security and SQL Injection, Cross-Site Scripting (XSS) and Cross-Site Request Forgery (CSRF), Security Testing and Code Review, Web Application Firewalls (WAF), Secure Development Frameworks and Libraries, Mobile Application Security, API Security, Secure Coding Standards and Best Practices, Application Security Tools and Technologies, Application Security Incident Response.

Unit V: Virtualization & Cloud Security **9 Lecture Hours**

Virtualization & Cloud Security, Introduction to Virtualization Technologies, Virtualization Deployment Models, Virtual Machine Security, Hypervisor Security, Virtual Network Security, Virtualization Management and Monitoring, Cloud Computing Fundamentals, Cloud Service Models (IaaS, PaaS, SaaS), Cloud Deployment Models (Public, Private, Hybrid), Cloud Security Challenges and Risks, Identity and Access Management in the Cloud, Data Security and Privacy in the Cloud, Cloud Compliance and Legal Considerations, Cloud Incident Response and Forensics, Cloud Security Best Practices.

Total lecture Hours 45

Textbooks

1. Gerardus Blokdyk, **"**Mobile Operating System Security A Complete Guide", The Art of Service - Mobile Operating System Security Publishing, 2020.

Reference Books

1. Chris Dotson, "Practical Cloud Security: A Guide for Secure Design and Deployment", O′Reilly, 2019.
2. Bryan Sullivan, "Web Application Security, A Beginner's Guide", Osborne/McGraw-Hill, 2012.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSSF4117P	**OS, Application & Cloud Security Lab**	0	0	2	1
Total Units to be Covered: 10		**Total Contact Hours: 30**			
Prerequisite(s):	Ethical Hacking & Penetration Testing	**Syllabus version: 1.0**			

Course Objectives

The course objectives of the OS, Application & Cloud Security Lab are to provide students with comprehensive knowledge and practical skills in securing operating systems, applications, and cloud environments. The lab aims to develop proficiency in hardening operating systems, assessing application security, implementing secure coding practices, conducting vulnerability scanning and patch management, deploying web application firewalls (WAFs), configuring cloud security settings, ensuring container security, managing identity and access, and deploying and configuring cloud services securely. By achieving these objectives, students will be equipped to effectively protect and defend critical components of information systems against potential cyber threats and vulnerabilities in the realm of operating systems, applications, and cloud environments.

Course Outcomes

On completion of this course, the students will be able to

CO1. Proficiency in implementing security measures to harden operating systems, ensuring the protection and resilience of critical system components against potential cyber threats and vulnerabilities.

CO2. Mastery of application security assessment techniques, allowing students to identify and mitigate security weaknesses in applications, thereby enhancing the overall security posture of software systems.

CO3. Competence in applying secure coding practices, enabling students to develop software with built-in security measures and reduce the risk of vulnerabilities that can be exploited by attackers.

CO4. Proficient in conducting vulnerability scanning and implementing effective patch management strategies, enabling students to proactively identify and remediate vulnerabilities in operating systems, applications, and associated components.

CO5. Mastery of cloud security configuration techniques, including the deployment of web application firewalls (WAFs), ensuring the secure configuration of cloud environments, container security, and the proper implementation of identity and access management (IAM) controls.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	-	-	-	-	-	-	-	-	2	-	2
CO 2	-	-	-	1	-	-	-	-	-	-	-	-	-	3	-
CO 3	-	-	-	-	-	-	-	-	-	-	-	-	2	-	-
CO4	-	-	-	1	-	-	-	-	-	-	-	-	-	2	2
CO5	-	-	-	-	-	-	-	-	-	-	-	-	-	1	1
Average	-	-	-	0.4	-	-	-	-	-	-	-	-	0.8	1.2	1

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1 Hardening an Operating System

Experiment 2 Application Security Assessment

Experiment 3 Secure Coding Practices

Experiment 4 Vulnerability Scanning and Patch Management

Experiment 5 Web Application Firewall (WAF) Implementation

Experiment 6 Cloud Security Configuration

Experiment 7 Container Security

Experiment 8 Identity and Access Management (IAM)

Experiment 9 Secure Authentication and Authorization Mechanisms

Experiment 10 Secure Deployment and Configuration of Cloud Services

Total Lab hours 30

Textbooks

Reference Books

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

6. Big Data Track

Course Code	Course name	L	T	P	C
CSBD2010P	**Big Data Overview and Ingestion**	4	0	0	4
Total Units to be Covered: 06		**Total Contact Hours: 60**			
Prerequisite(s):	Basic Programming, Basic Linux	**Syllabus version: 1.0**			

Course Objectives

1. To understand the characteristics and challenges of Big Data.

2. To get familiarized with fundamentals of Big Data Ecosystem and Hadoop Architecture

3. Enable students to acquire knowledge of Data Ingestion and different Data Ingestion architectures.

4. Learn practical knowledge of applying skills and tools to ingest different types of data.

Course Outcomes

On completion of this course, the students will be able to

CO1: Understand the concept of Data and Big Data.

CO2: Explain various components of Big Data Ecosystem.

CO3: Exploring the real-world use cases and applications of Big Data.

CO4: Understanding of management principles and techniques, of Big Data Ingestion

CO5: Infer various data ingestion architectures and apply data ingestion concepts using Sqoop & Kafka.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	-	1	-	-	-	-	-	-	-	-	-	1	-	1

CO 2	2	1	-	-	-	-	-	-	-	-	-	-	1	1	2
CO 3	2	2	1	-	-	-	-	-	-	-	-	-	2	2	2
CO4	2	2	2	-	-	-	-	-	-	-	-	-	2	2	3
CO5	2	2	2	-	-	-	-	-	-	-	-	-	1	1	2
Average	2	1.4	1.2	-	-	-	-	-	-	-	-	-	1.4	1.2	2

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to Big Data **10 Lecture Hours**

Data Growth explosion, Categories of Data, Different Data Storage mechanisms, Introduction to a flat file, tabular and relational databases, Introduction to NoSQL data stores, Characteristics of Big data, (Volume, Velocity, Variety, Value, Veracity), Information mining and benefits of big data; Risks of Big Data, Structure of Big Data;

Unit II: Real World Applications and Use cases **8 Lecture Hours**

Need for and Importance of Big Data Analytics, The Need for Standards; Case studies from various industries (e.g., finance, healthcare, e-commerce), Emerging trends and future directions in big data.

Unit III: Big Data Ecosystem **12 Lecture Hours**

Introduction to Apache Hadoop Ecosystem and its components. Hadoop File System (HDFS); Scalable Storage, Data locality, Resilience and fault tolerance. HDFS Administration; Setup of a Hadoop Cluster; Data loading into Hadoop, Map / Reduce concepts; Managing Job Execution using Yarn; Data analysis and Visualization.

Unit IV: Data Ingestion in Big data **10 Lecture Hours**

Different sources of Big data, Introduction to Data ingestion, Features and challenges in Big Data Ingestion, ETL and ELT, Data ingestion pipelines for Big Data, Ingesting batch vs streaming data.

Unit V: Batch Ingestion in Big data **10 Lecture Hours**

Techniques for ingesting batch data from various sources (e.g., files, databases), Sqoop Architecture, Sqoop Import and Export, Sqoop Job, Incremental Import and File Handling.

Unit VI: Real-Time Ingestion in Big Data **10 Lecture Hours**

Introduction to message queuing systems, Architecture of message queuing systems, Broker vs. Broker less, Distributed broker, Exchanges and Exchange Types, Distributed publish-subscribe messaging system, Kafka - Broker, Producer and consumer

Total lecture Hours 60

Textbooks

1. Big Data Overview – Xebia Course Material
2. Big Data Ingestion - Xebia Course Material

Reference Books

1. Sqoop: O'Reilly Media
2. The Big Data Ingestion: O'Reilly Media

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSBD2110P	**Big Data Overview and Ingestion Lab**	0	0	2	1
Total Units to be Covered: 10	**Total Contact Hours: 30**				
Prerequisite(s):	Basic Knowledge of Bigdata, Warehouse	**Syllabus version: 1.0**			

Course Objectives

1. Learn Hadoop setup and installation on a VM
2. To understand the basic working of Hadoop and Map Reduce programming.
3. Enable students to acquire knowledge of Data Ingestion and different Data Ingestion architectures.
4. Teach students in applying skills and tools to ingest different types of data.

Course Outcomes

On completion of this course, the students will be able to

CO1. To Setup Hadoop cluster on a VM and demonstrate basic HDFS commands.

CO2. Illustrate file transfer in Hadoop and the impact of MapReduce on the transfer.

CO3. Build the various concepts of data ingestion using Sqoop.

CO4. Demonstrate data ingestion concepts using Kafka.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	-	1	-	2	-	-	-	-	-	-	-	1	-	3
CO 2	2	1	-	-	2	-	-	-	-	-	-	-	1	1	3
CO 3	2	2	1	-	2	-	-	-	-	-	-	-	2	2	3
CO4	2	2	2	-	2	-	-	-	-	-	-	-	2	2	3
Average	2	1.25	1	-	2	-	-	-	-	-	-	-	1.5	1.25	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Installation of the Cloudera QuickStart virtual machine

1. Install Cloudera Hadoop Framework on the VM
2. Check the Hadoop tools present and running using HUE

Experiment 2: Understand Basic Linux and HDFS commands

1. Demonstrate various Linux commands
2. Demonstrate Basic HDFS commands
3. Write commands for data transfer to/from HDFS
4. Visualising the HDFS file system using HUE

Experiment 3: Importing data using Sqoop

1. Login to MySQL
2. Create a database **upes_db**
3. Create table **students** in the **upes_db** database

Rollno	Name	Marks	Subject
1	Amit	90	Physics
2	Sumit	85	Math
3	Ram	88	Computer
4	Sandeep	88	Physics
5	John	86	Math
6	Tom	85	Computer
7	Mani	89	Physics
8	Akshay	88	Math
9	Arun	91	Computer

4. Using Sqoop transfer all the data of **students** table into HDFS.
5. Using Sqoop transfer all the data of **students** table into HDFS to a specific folder **students_1**.

6. Using Sqoop transfer all the data of **students** table who are studying **Physics** subject into HDFS to a specific folder **students_2.**

7. Using Sqoop transfer all the data of **students** table who are studying **Computer** subject into HDFS to a specific folder **students_pass,** without specifying the password on the command line**.**

8. Import **students** table from MySQL as a text file to the destination folder **students_text.** Fields should be terminated by a tab character ("\t") character and lines should be terminated by newline character ("\n").

Experiment 4: Compression and Data Format data using Sqoop

1. Import **students** table from MySQL into HDFS to the destination folder **students_avro**. The file should be stored as Avro file.

2. Import **students** table from MySQL into HDFS to the destination folder **students_parquet**. The file should be stored as parquet file.

3. Import **students** table from MySQL into HDFS to the destination folder **students_sequence**. The file should be stored as a sequence file.

4. Import **students** table from MySQL into HDFS to the destination folder **students_compress**. Decrease the size occupied by the generated file using **GZip** codec.

5. Import **students** table from MySQL into HDFS to the destination folder **students_mappers**. Create a number of mapper 2 and use **snappy** compression.

6. Create a new table **library** in the **upes_db** database and import all tables from MySQL database **upes_db** into HDFS as Avro data files use compression

Id	Title	Author
1	Java Programming	James Gosling
2	DBMS	Navathe
3	C Language	Yashwant Kanetkar
4	Big Data	Tom
5	Statistics	Atul
6	Networking	William

Experiment 5: Incremental import using Sqoop

1. In Students table insert more data as below

Rollno	Name	Marks	Subject

10	Akshat	88	Physics
11	Akshay	85	Math
12	Sunil	87	Computer
13	Mandeep	84	Physics

2. Using Sqoop transfer only new rows of **students** table into HDFS to a specific folder **students_increment** and verify the content at the target directory in HDFS.
3. Remove primary key from **Library** table in the upes_db database using (alter, drop, modify) SQL commands.
4. Using Sqoop transfer all the data of the **Library** table into HDFS to a specific folder **library_noPk** without using primary key and verify the content at the target directory in HDFS.
5. Using Sqoop do the following. Read the entire steps before you create a Sqoop job.
   - create a Sqoop job Import Students table as the text file to directory **Students_job**.
   - Import all the new inserted three records to **Students** table from MySQL.
   - Run the Sqoop job so that only newly added records can be pulled from MySQL.
   - Validate to make sure that no duplicate records in **HDFS**

Experiment 6: Free form query, export and Sqoop merger

1. Create a MySQL table named **students_1** and load data from **/home/cloudera/students_1**

Validate to make sure the records have been added to the database.

2. Using Sqoop, import **students_replica** table from MYSQL into HDFS such that fields are separated by a '|' and lines are separated by '\n'. Null values are represented as -1 for numbers and "NOT-AVAILABLE" for strings. Only records with roll no greater than or equal to 1 and less than or equal to 88 should be imported and use 3 mappers for importing. The destination file should be stored as a text file to directory **/home/cloudera/students_replica**
3. Using Sqoop transfer all the data of **students** table who are studying **Physics** subject into HDFS to a specific folder **students_query,** use SQL query to import the data.

4. Using Sqoop transfer the data of **students** table into HDFS to a specific folder **students_m1** only student roll no less than or equal to 5, use SQL query to import the data.
5. Using Sqoop transfer all the data of **students** table into HDFS to a specific folder **students_m2** only student roll no greater than 5.
6. Using sqoop merge data available in **students_m1** and **students_m2** to produce a new set of files in **students_both.**

Experiment 7: Lab setup and configuration for Kakfa

1. Install JDK 8 or higher (it's a pre-requisite)
2. Download Kafka, install and configure the properties file for Kafka and Zookeeper.
3. Installing Java 8 & IntelliJ Community Edition.

Experiment 8 and 9: Kafka Producer and Consumer - Command Line

1. Create a topic name first_topic with 1-replication factor and one partition.
2. Verify if the topic is created or not.
3. Describe topic first_topic
4. Create a topic name second_topic with 3-replication factor and one partition (use configuration properties file).
5. Verify all the topics
6. Describe topic second_topic to check all ISR.
7. Send message to the first_topic.
8. In new window read the first_topic.
9. Read the first_topic from the beginning.
10. Modify the first_topic change partition to 3.
11. Delete the second_topic

Experiment 10: Kafka Producer and Consumer – Java application

1. Create a Producer application using Java to implement fire and forget method of sending messages in Kafka.
2. Create the consumer application to read the message from the Producer.
3. Create a Producer application using Java to implement Synchronous way of sending messages in Kafka.
4. Create the consumer application to read the message from the Producer.
5. Create a Producer application using Java to implement Asynchronous way of sending messages in Kafka.

6. Create the consumer application or through console window to read the message from the Producer.

Total Lab hours 30

Textbooks

1. Kathleen Ting, and Jarek Jarcec Cecho, “Apache Sqoop Cookbook”, O'Reilly Media, 2013.
2. Neha Narkhede, Gwen Shapira, and Todd Palino, “Kafka: The Definitive Guide”, O'Reilly Media, 2017.

Reference Books

1. Sqoop: O'Reilly Media
2. The Big Data Ingestion: O'Reilly Media

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name		L	T	P	C
CSBD3015P	**Big Data Storage and Analysis**		4	0	0	4
Total Units to be Covered: 05		**Total Contact Hours: 60**				
Prerequisite(s):	DBMS, DS		**Syllabus version: 1.0**			

Course Objectives

1. Enable students to acquire the knowledge of NoSQL and differentiate with RDBMS.

2. Know the internal architecture of a distributed database file system.

3. To provide students with the concept of Big Data analysis and different frameworks.

4. Teach students in applying skills and tools to manage and analyze Big data.

Course Outcomes

On completion of this course, the students will be able to

CO1. Illustrate the file system namespace, NoSQL, and Data Lake concepts.

CO2. Infer Distributed file system concepts and Data Lake.

CO3: Define the concepts of big data analysis framework.

CO4. Describe data definition, description, data munging and transformation.

CO5. Build querying, data management, data storage using Hive and Pig.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	-	1	-	-	-	-	-	-	-	-	-		1	2
CO 2	2	-	-	1	1	-	-	-	-	-	-	-	2	1	2
CO 3	2	2	-	1	2	-	-	-	-	-	-	-	2	2	3
CO4	2	2	-	2	2	-	-	-	-	-	-	-	2	2	3

CO5	-	-	2	-	1	-	-	-	-	-	-	-	-	2	2
Average	1.6	.8	.6	.8	1.2	-	-	-	-	-	-	-	1.2	1.6	2.4

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

Syllabus

Unit I: The File System Namespace **14 Lecture Hours**

Traditional file systems, File system hierarchy, Limitations of traditional file systems, Introduction to NoSQL databases, Compare and contrast with RDBMS, Document-oriented, columnar, graph-based and key value pair. Scaling using sharding. Introduction to Data Lake.

Unit II: Distributed File Systems **14 Lecture Hours**

Introduction to DFS, Client-server architecture in DFS, Cluster-based architecture in DFS, Google File System; HDFS - Block size, daemons and roles, Replication, robustness and fault tolerance, Data Disk Failure, Heartbeats and Re-Replication, High availability, Checkpointing; HBase.

Unit III: Introduction to Big Data Analysis **12 Lecture Hours**

Requirement of Big Data analysis, Exploring Big data, mining big data, Challenges in analyzing big data. Scripting like frameworks, SQL like frameworks, Programming for Big Data analysis.

Unit IV: Data Definition, Description, and Transformation **10 Lecture Hours**

Data type conversions, Tables, Partitions, Buckets, Views, Performing data munging, exploratory data analysis, Data transformation.

Unit V: Common Data Analysis Frameworks **10 Lecture Hours**

Build querying, data management, data storage using Hive, Impala, and Pig

Total lecture Hours 60

Textbooks

1. Big Data Storage– Xebia Course Material
2. Big Data Analysis- Xebia Course Material

Reference Books

1. Hadoop: The Definitive Guide, 4th Edition - O'Reilly Media
2. MongoDB: The Definitive Guide, 2nd Edition - O'Reilly Media
3. Integrated Analytics: Courtney Webster
4. Programming Hive - O'Reilly Media
5. Programming Pig, 2nd Edition - O'Reilly Media

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSBD3115P	Big Data Storage and Analysis Lab		0	0	2	1
Total Units to be Covered: 10		Total Contact Hours: 30				
Prerequisite(s):	DBMS, DS		Syllabus version: 1.0			

Course Objectives

1. Enable students to apply the knowledge of NoSQL and differentiate with RDBMS.
2. Know the internal architecture of a distributed database file system.
3. Teach students in applying skills and tools to manage and analyze Big data.

Course Outcomes

On completion of this course, the students will be able to

CO1. Design NoSQL Database using MongoDB.

CO2. Apply the distributed file system concepts in the Hadoop Distributed File System (HDFS).

CO3. Implement HBase for data storage and analysis.

CO4. Build querying, data management, data storage using Hive.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	-	-	-	-	-	-	-	-	-	-	-	2	2	3
CO 2	2	-	2	1	1	-	-	-	-	-	-	-	2	2	3
CO 3	2	2	-	1	2	-	-	-	-	-	-	-	2	2	3
CO4	2	2	-	2	2	-	-	-	-	-	-	-	2	2	3
Average	2	1	.5	. 1	1.25	-	-	-	-	-	-	-	2	2	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1 and 2: Working with NoSQL Document DB – MONGODB

i. To understand and perform the commands on MongoDB.

Experiment 3: Working with HDFS

i. Hadoop Installation and Configuration on Cloud Era
ii. To understand the basic commands of Hadoop.
iii. To understand the advance Hadoop commands.

Experiment 4: HDFS Administration and Visualisation

i. To study the HDFS Administration commands and their uses.
ii. To visualise Hadoop cluser and the HDFS using HUE interface.

Experiment 5: HBase Data Import

To understand how HBase data import is done.

Experiment 6: Explore HBase Shell

i. To study and understand how to access data in HBase Shell.
ii. Understanding how multiple operations on data-tables can be performed to get efficient data storage and flexible interaction to the clients.
   a. Create a table, add rows, add column families, replace a row, drop table
   b. retrieve a row and a cell, retrieve many rows from a given column family
   c. create a new version of a value in a cell

Experiment 7: HIVE Data Import

To understand how HIVE data import is done.

Experiment 8: Explore HIVE Shell – internal table

i. create an internal relational table
ii. find a location of internal relational table in HDFS
iii. insert a row into an internal relational table
iv. load data into an internal relational table
v. Describe and drop table

Experiment 9: Explore HIVE Shell – external table

i. create an external relational table

ii. insert a row into an external relational table

iii. load data into an external relational table

iv. Describe and drop table

Experiment 10: HIVE query language

i. Use HQL to query the relational database

ii. create and process a self-contained HQL script

Total Lab hours 30

Textbooks

1. Xebia Materials
2. Tom White, "Hadoop: The Definitive Guide", 4th Edition, O'Reilly Media, 2015.

Reference Books

1. Lars George, "HBase: The Definitive Guide", O'Reilly Media, 2011.
2. Edward Capriolo, Dean Wampler, and Jason Rutherglen, "Programming Hive", O'Reilly Media, 2012.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	**Quiz & Viva**	**Performance & Lab Report**
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSBD3016P	Big Data Processing - Disk Based and In Memory	4	0	0	4
Total Units to be Covered: 06	Total Contact Hours: 60				
Prerequisite(s):	Basics of python and java	Syllabus version: 1.0			

Course Objectives

1. Understand Hadoop MapReduce framework and the working of MapReduce on data stored in HDFS.

2. Learning YARN concepts in MapReduce.

3. To understand In-Memory computation: Spark framework.

4. To learn the concept of RDD- Resilient Distributed Dataset, data frames, data sets, transformations, and actions.

5. To understand the working of Spark in Python and Scala.

Course Outcomes

On completion of this course, the students will be able to

CO1. Elaborate distributed processing framework.

CO2. Infer Map-Reduce architecture and build basic MapReduce programs.

CO3. Combine resource management framework like YARN with MapReduce.

CO4. Discuss Spark architecture and explain the use of core Spark APIs.

CO5. Analyze typical use cases of Spark.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	2	-	1	-	-	-	-	-	-	-	-	-	-	1	2
CO 2	2	-	-	1	1	-	-	-	-	-	-	-	2	1	2
CO 3	-	-	-	-	2	-	-	-	-	-	-	-	-	1	3

CO4	-	-	-	1	2	-	-	-	-	-	-	-	-	1	3
CO5	2	-	-	1	2	-	-	-	-	-	-	-	-	1	3
Average	1.2	-	.2	.6	1.4	-	-	-	-	-	-	-	.4	1	2.6

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

Syllabus

Unit I: Performing distributed processing **10 Lecture Hours**

Scaling processing, Distributed processing architectures, Batch processing

Unit II: Implementing MapReduce **10 Lecture Hours**

Map Reduce for distributed processing, Data locality principle, Mapper and Reducer.

Unit III: Implementing YARN **10 Lecture Hours**

Introduction to YARN, Resource Manager, Node Manager, Application Master.

Unit IV: Compare and contrast Disk Based and In-Memory architecture **10 Lecture Hours**

Performance of Disk based distributed computation, Limitations of Disk-based architectures, Introduction to in-memory distributed computation, comparison of Disk-based and In-Memory processing.

Unit V: Overview of current In-Memory technologies **10 Lecture Hours**

In-Memory architecture. Example: Spark, Driver and Worker nodes, Spark Context and Executors

Unit VI: Concept of RDD and Working with Spark **10 Lecture Hours**

Introduction to RDD, Actions and transformations, RDD lineage, PySpark, Building with Spark API using Python, Building with Spark API using Scala .

Total lecture Hours 60

Textbooks

1. Big Data Storage– Xebia Course Material

2. In-Memory Computing - Xebia Course Material

Reference Books

1. Tom White, "Hadoop: The Definitive Guide", 4th Edition, O'Reilly Media, 2015.

2. Bill Chambers, and Matei Zaharia, "Spark: The Definitive Guide", O'Reilly, 2018.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSBD3116P	Big Data Processing Lab	0	0	2	1
Total Units to be Covered: 10	Total Contact Hours: 30				
Prerequisite(s):	DBMS, DS	Syllabus version: 1.0			

Course Objectives

1. To understand Hadoop MapReduce framework, Mapreduce daemons and working of MapReduce on data stored in HDFS.
2. To understand disk-based computing and In-Memory computing: Spark.
3. To understand the concept of RDD- Resilient Distributed Dataset
4. To understand the working of Spark in Python and Scala.

Course Outcomes

On completion of this course, the students will be able to

CO1. Design and solve MapReduce programs with the Hadoop Core API.

CO2. Operate on data using core Spark APIs.

CO3. Analyze large data sets using Spark SQL and DataFrames.

CO4. Apply Spark framework on typical use cases.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	1	-	-	2	-	-	-	-	-	-	-	1	1	3
CO 2	-	2	2	-	3	-	-	-	-	-	-	-	2	2	3
CO 3	-	3	3	2	3	-	-	-	-	-	-	-	2	3	3
CO4	-	3	3	3	3	-	-	-	-	-	-	-	2	3	3
Average	-	2.25	2	1.25	2.75	-	-	-	-	-	-	-	1.75	2.25	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1

i. Write down the steps required to install and configure the MapReduce on Cloudera quickstart virtual machine.
ii. Write a MapReduce job to count the numbers of words in a given file.

Experiment 2

i. Exploring YARN and HUE
ii. Write a MapReduce job that reads any text input and computes the average length of all words that start with each character

Experiment 3

i. Add a Combiner in WordCount program to reduce the amount of intermediate data sent from the Mapper to the Reduce
ii. Write an application that counts the number of times words appear next to each other.

Experiment 4

Create an application that will process a web server's access log to count the number of times gifs, jpegs, and other resources have been retrieved. The job will report three figures: the number of gif requests, number of jpeg requests, and the number of other requests

Experiment 5

i. Write a MapReduce job that reads any text input and computes the average length of all words that start with each character.
ii. Write a MapReduce job with multiple Reducers, and create a Partitioner to determine which Reducer each piece of Mapper output is sent to.

Experiment 6

i. To install spark and understand spark daemons.
ii. Work with Spark Repl and Pyspark : word count problem

Experiment 7

i. To build page rank algorithm using spark (Python)
ii. To build page rank algorithm using spark (Python)

Experiment 8. Working with Scala Programming Language

To create a project in Scala programming language: Word count problem,

Experiment 9

i. To build a simple analytical dashboard using PySpark and flask: election polling problem

ii. To work with Spark GraphX

Experiment 10

i. To build a spark application using Python and Mysql

ii. To monitor and trouble shoot spark application

Total Lab hours 30

Textbooks

1. B. Chambers and M. Zaharia, Spark: The Definitive Guide, O’Reilly. ISBN:978-93-5213-706-0
2. Hien Luu, Beginning Apache Spark 2:With Resilient Distributed Datasets, Spark SQL, Structured Streaming And Spark Machine Learning Library, APress, ISBN: 9781484235799

Reference Books

1. Sandy Ryza, Uri Laserson, Advanced Analytics with Spark: Patterns for Learning from Data at Scale 2nd Edition, Kindle Edition, O’Reilly. ISBN:978-1-491-91276-8

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name		L	T	P	C
CSBD4008P	**Stream Processing**		4	0	0	4
Total Units to be Covered: 05		**Total Contact Hours: 60**				
Prerequisite(s):	Basics of python and java		**Syllabus version: 1.0**			

Course Objectives

1. Explain a few concepts of Spark streaming
2. Describe basic and advanced sources
3. Explain how stateful operations work
4. Explain window and join operations

Course Outcomes

On completion of this course, the students will be able to

CO1. Learn Data Streaming concepts and different frameworks.

CO2. Design Spark Streaming applications using the Scala/Python programming language.

CO3. Analyze different access log data and transform streams of it.

CO4. Build streaming data through stateful and stateless streaming.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	1	-		2	-	-	1	-	-	-	-	2	1	3
CO 2	1		-	1	2	-	-	1	-	-	-	-	3	3	3
CO 3	-	1	-	1	1	-	-		-	-	-	-	2	2	1
CO4	1	-	-	2	2	-	-	1	-	-	-	-	2	1	2
Average	.5	.5	-	1	1.75	-	-	.75	-	-	-	-	2.25	1.75	2.25

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

Syllabus

Unit I: Overview of Stream Data Processing **14 Lecture Hours**

Batch versus real-time data processing, Understanding streams, Data storage layer, Input data streams, Resilient distributed datasets and discretized streams, Fault-tolerant Stream Processing. Benefits and challenges of Stream Processing, stream processing frameworks.

Unit II: Discretized Stream (Dstream) in Spark **14 Lecture Hours**

Stream Processing architecture, Programming Model – Dstream, Streaming Context, Input Dstreams and Receivers, Transformations on DStreams, Output Operations on DStreams, DataFrame and SQL Operations, MLlib Operations.

Unit III: Structure Streaming programming **14 Lecture Hours**

Stateful streaming, Stateless streaming, Basic Operations - Selection, Projection, Aggregation, Window-based Transformations, Join Operations, Streaming Deduplication, Arbitrary Stateful Computations, Streaming Queries.

Unit IV: Streaming Architectures **10 Lecture Hours**

Components of a Data Platform, Architectural Models, The Use of a Batch-Processing Component in a Streaming Application, Referential Streaming Architectures, The Lambda Architecture, The Kappa Architecture, Streaming Versus Batch Algorithms

Unit V: Use case for Stream Processing **8 Lecture Hours**

Processing Distributed Log Files in Real Time, Fraud Detection, Stock Market, etc.

Total lecture Hours 60

Textbooks

1. Stream Processing– Xebia Course Material

2. Hien Luu, “Beginning Apache Spark 2: With Resilient Distributed Datasets, Spark SQL, Structured Streaming and Spark Machine Learning Library”, Apress, 2018

Reference Books

1. Francois Garillot, and Gerard Maas, “Stream Processing with Apache Spark”, O'Reilly Media, 2019.

2. Bill Chambers, and Matei Zaharia, "Spark: The Definitive Guide", O'Reilly Media, Inc., 2018.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSBD4101P	Stream Processing Lab		0	0	2	1
Total Units to be Covered: 12		Total Contact Hours: 30				
Prerequisite(s):	DBMS, Scala, Spark		Syllabus version: 1.0			

Course Objectives

1. Learn Querying in Hadoop Cluster using Sqoop and Flume.
2. Learn to build the application for Data Stream.
3. Able to write Spark Streaming programs.

Course Outcomes

On completion of this course, the students will be able to

CO1. Developing application in Scala using Spark Streaming which will integrate with Twitter.

CO2. Developing application in Scala using Spark Streaming for analyzing streaming dataset for webpage views.

CO3. Developing application in Scala using Spark Streaming which will integrate with the Spark SQL. The application will read the data from the socket stream and count the words.

C04. Developing application in Scala for Stateful and Stateless Streaming implementation.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	1	-		2	-	-	1	-	-	-	-	2	1	
CO 2	1	-	-	1	2	-	-	1	-	-	-	-	3	3	1
CO 3	-	1	-	1	1	-	-		-	-	-	-	2	2	
CO4	1	-	-	2	2	-	-	1	-	-	-	-	2	1	1
Average	.4	.4	-	.8	1.4	-	-	.6	-	-	-	-	1.8	1.4	.4

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Spark Streaming with Twitter Data

Experiment 2: Spark Streaming with Clickstream Data

Experiment 3: Integrating Spark Streaming with Spark SQL

Experiment 4: Integrating Spark Streaming with Apache Kafka

Experiment 5: Running with Spark-submit

Experiment 6: Implementing Spark Stateless Streaming

Experiment 7: Implementing Spark Stateful Streaming

Experiment 8: Performing Window-based Transformations

Experiment 9: Working with Arbitrary Combinations of Batch and Streaming Computation

Experiment 10: Analyzing Apache Log Files with Structured Streaming

Experiment 11: Streaming Applications with Different Data Formats

Experiment 12: ETL with Streaming Applications

Total Lab hours 30

Textbooks

1. Stream Processing– Xebia Course Material
2. Hien Luu, "Beginning Apache Spark 2: With Resilient Distributed Datasets, Spark SQL, Structured Streaming And Spark Machine Learning Library", Apress, 2018.

Reference Books

1. Francois Garillot, and Gerard Maas, "Stream Processing with Apache Spark: Mastering Structured Streaming and Spark Streaming", O′Reilly, 2019.
2. Bill Chambers, and Matei Zaharia, "Spark: The Definitive Guide", O'Reilly Media, Inc., 2018.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	**Quiz & Viva**	**Performance & Lab Report**
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSBD4009P	Big Data Search and Security	3	0	0	3
Total Units to be Covered: 05		Total Contact Hours: 45			
Prerequisite(s):	Basic Knowledge of Bigdata, Warehouse	Syllabus version: 1.0			

Course Objectives

1. To understand Big Data Search tools: Elastic-search and Lucene

2. To Identify the key components of Lucene and setup a basic search solution

3. Assess threats to a production Hadoop cluster, plan and deploy defenses against these threats.

4. Set up authentication and authorization with Kerberos and Apache Sentry.

Course Outcomes

On completion of this course, the students will be able to

CO1. Discuss the big data search and analyze problems related to Elastic-search.

CO2. Explain the core components of Lucene and its architecture.

CO3. Explain data protection in distributed environments.

CO4. Discuss the need of securing distributed systems with Kerberos and Sentry.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	2	-	-	-	-	-	-	-	1	3	3
CO 2	-	-	-	1	2	-	-	-	-	-	-	-	1	3	3
CO 3	2	-	-	1	2	-	-	-	-	-	-	-	1	3	3
CO4	1	1	-	-	-	-	-	1	-	-	-	-	2	2	3
Average	.75	.25	-	.5	1.5	-	-	.25	-	-	-	-	1.25	2	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

Syllabus

Unit I: Performing exploration on massive volume of data **10 Lecture Hours**

Requirement of Big Data search, Insights using Big Data search, Text search, Big Data search tools – Elastic Search and Lucene.

Unit II: Working with Elastic Search and Lucene **10 Lecture Hours**

Building Indexes, Document, Mapping in elastic search, Data types, Nodes and shards; Architecture of Lucene, Terms: Document, Field, Term, Token, Lucene scoring, Lucene based search tools, build a sample application using Lucene.

Unit III: Securing Distributed Systems **9 Lecture Hours**

Threat Categories, Threat and Risk Assessment, Vulnerabilities

Unit IV: Using Kerberos and Sentry with distributed systems **8 Lecture Hours**

Kerberos Overview, Kerberos Trusts, Kerberos Workflow, MIT Kerberos, The Sentry Service, Sentry Privilege Models, Sentry Policy Administration

Unit V: Data protection in distributed environments **8 Lecture Hours**

Integrity of Ingested Data, Data Ingest Confidentiality, Ingest Workflows, Encryption Algorithms, Encrypting Data at Rest, Encrypting Data in Transit

Total lecture Hours 45

Textbooks

1. Big Data Search – Xebia Course Material
2. Big Data Security - Xebia Course Material

Reference Books

1. Andrea Gazzarini, “Apache Solr Essentials”, Packt Publishing Limited, 2015.
2. Clinton Gormle, “Elasticsearch: The Definitive Guide”, O'Reilly Media, 2015.

3. Ellen Friedman, and Ted Dunning, “Sharing Big Data Safely: Managing Data Security”, O'Reilly Media, 2016.

4. Davi Ottenheimer, “The Realities of Securing Big Data”, John Wiley & Sons, 2020.

5. Ben Spivey, and Joey Echeverria, “Hadoop Security: Protecting Your Big Data Platform”, O'Reilly Media, 2015.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSBD4109P	Big Data Search and Security Lab		0	0	2	1
Total Units to be Covered: 08		Total Contact Hours: 30				
Prerequisite(s):	DBMS, Scala, Spark		Syllabus version: 1.0			

Course Objectives

1. Learn Big Data Search tools: Elastic-search and Lucene
2. Identify the key components of Lucene and setup a basic search solution
3. Assess threats to a production Hadoop cluster, plan and deploy defenses against these threats.
4. Set up authentication and authorization with Kerberos and Apache Sentry.

Course Outcomes

On completion of this course, the students will be able to

CO1. Exploring the big data search tools and analyze problems related to Elastic-search.
CO2. Solve real-world Big Data Search problems using Lucene.
CO3. Learning data protection in distributed environments.
CO4. Securing distributed systems with Kerberos and Sentry.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	-	2	-	-	-	-	-	-	-	1	1	3	3
CO 2	2	2		3	-	-	-	-	-	-	-	2	2	3	3
CO 3	3	3	2	3	-	-	-	-	-	-	-	2	3	3	3
CO4	1	-	-	2	2	-	-	1	-	-	-	-	2	1	1
Average	1.75	1.25	.5	2.5	.5	-	-	.25	-	-	-	1.75	2	2.5	2.5

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment No: 01 Installation

Installing lucene and build a sample application

Experiment No: 02 Indexing

Indexing with local pdf files using SOLR

Experiment No: 03 Text search

Full text search using SOLR

Experiment No: 04 working with data structure, Functions and methods

Working with query dsl in elasticsearch

Experiment No: 05 Altering apache lucene scoring

Experiment No: 06 Access Control and Authentication Testing- kerberos

Experiment No: 07 Data Encryption and Decryption – data at rest and in transit

Experiment No: 08 Data Auditing and Monitoring

Total Lab hours 30

Textbooks

1. Big Data Search – Xebia Course Material
2. Big Data Security - Xebia Course Material

Reference Books

1. Andrea Gazzarini, “Apache Solr Essentials”, Packt Publishing Limited, 2015.
2. Clinton Gormle, “Elasticsearch: The Definitive Guide”, O'Reilly Media, 2015.
3. Ellen Friedman, and Ted Dunning, “Sharing Big Data Safely: Managing Data Security”, O'Reilly Media, 2016.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

7. Data Science Track

Course Code	Course name		L	T	P	C
CSDS2001P	**Fundamentals of Data Science**		4	0	0	4
Total Units to be Covered: 06		**Total Contact Hours: 60**				
Prerequisite(s):	Basics of mathematics, programming		**Syllabus version: 1.0**			

Course Objectives

1. To understand the concept of data science.

2. To understand techniques and methods related to the area of data science on real world applications.

Course Outcomes

After the completion of the course the students will be able to

CO1: Understand the fundamentals of data processing.
CO2: Understand and apply mathematical concepts in the field of data science.
CO3: Employ the techniques and methods related to the area of data science in a variety of applications.
CO4: Apply logical thinking to understand and solve the problem in context.
CO5: Apply the entire concept in data analysis tools.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	3	2	2	-	-	-	-	-	-	-	-	-	3	3
CO 2	-	3	2	2	-	-	-	-	-	-	-	-	-	3	3
CO 3	-	3	2	3	-	-	-	-	-	-	-	-	-	3	3
CO4	-	3	2	3	-	-	-	-	-	-	-	-	-	3	3
CO5	-	3	2	2	-	-	-	-	-	-	-	-	-	2	3
Average	-	3	2	2.4	-	-	-	-	-	-	-	-	-	2.8	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to Data Science **8 Lecture Hours**

Fundamentals of Data Science, Real World Applications, Data Science Challenges, Software Engineering for Data Science (DataOps, MLOps (intro)). Data science process roles, Stages in data science.

Defining Analytics, Types of data analytics (Descriptive, Diagnostic, Predictive, Prescriptive)

Data Science Process: CRISP-DM Methodology, SEMMA, BIG DATA LIFE CYCLE, SMAM.

Unit II: Probability and statistics for Data Science **12 Lecture Hours**

Probability: Introduction, finite sample spaces, conditional probability, independence; Random variables, distribution functions, probability mass and density functions, standard univariate discrete and continuous distributions; Mathematical expectations, moments; Random vectors, joint, marginal, and conditional distributions, independence, covariance, correlation, standard multivariate distributions, functions of random vectors; central limit theorem.

Statistics: Sampling distributions of the sample mean and the sample variance for a normal population; Point and interval estimation; Sampling distributions (Chi-square, t,F,Z), Hypothesis testing; One tailed and two-tailed tests; Analysis of variance, ANOVA, One way and two way classifications

Unit III: Data, Data Sources and Visualization **15 Lecture Hours**

Types of Data and Datasets, Data Quality, and Issues, Data Models, General Framework of Formal modeling, Association Analyses, Prediction Analyses, Data Pipelines and patterns, Data from files & working with relational databases, Diverse data sources, data warehouses, data mining, cloud, and Data lake: Characteristics, components, Data Streaming Ingestion, Batch Data Ingestion, Data Cataloging, Data Pipeline Stages (extraction, ingestion, cleaning, exploration, wrangling, versioning, Data transformation, Feature management). Data Visualization: Overview of visualization techniques for Data Exploratory analysis

Unit IV: Feature Engineering and Optimization **10 Lecture Hours**

Feature Extraction, Feature Construction, Feature Subset selection, Feature Learning, Feature Reduction (Dimensionality Reduction) Case Study involving FE tasks, and Feature Engineering techniques for text, images, audio, and video.Necessary and sufficiency conditions for optima; Gradient descent methods; Constrained optimization; Introduction to non-gradient techniques; Introduction to least squares optimization; Optimization view of machine learning.

Unit V: Supervised and unsupervised learning **10 Lecture Hours**

Introduction to Machine Learning, types, Supervised Learning: Overview, workflow, data processing, Linear Regression, Logistic Regression, Decision Trees, Random Forest, Support Vector Machines (SVM), k-Nearest Neighbors (k-NN).

Unsupervised Learning: Overview, clustering algorithms: K-Means Clustering, Hierarchical Clustering, DBSCAN, Gaussian Mixture Models (GMM),

Dimensionality Reduction: Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE)

Association Rule Mining: Apriori Algorithm, FP-Growth Algorithm, Anomaly Detection, Model Evaluation (Silhouette Score, Inertia, etc.)

Use Cases and Practical Applications

Unit VI: Data Analysis Tool **5 Lecture Hours**

Reading and getting data into R, ordered and unordered factors i.e arrays and matrices – lists and data frames, reading data from files, probability distributions statistical models in R - manipulating objects – data distribution.

Total lecture Hours 60

Textbooks

1. G. Strang, “Introduction to Linear Algebra”, 5^{th} Edition, Wellesley-Cambridge Press, USA, 2016.

4. Avrim Blum, John Hopcroft, and Ravindran Kannan, "Foundations of Data Science", 2018. Available online at: https://www.cs.cornell.edu/jeh/book.pdf.
2. D. C. Montgomery, and G. C. Runger, "Applied Statistics and Probability for Engineers", 5th Edition, John Wiley & Sons, Inc., NY, USA, 2011.
3. Nina Zumel, and John Mount, "Practical Data Science with R", Manning Publications, 2014.

Reference Books

1. Mark Gardener, "Beginning R - The Statistical Programming Language", John Wiley & Sons, Inc., 2012.
2. W. N. Venables, D. M. Smith and the R Core Team, "An Introduction to R", 2013. Available online at: https://cran.r-project.org/doc/manuals/R-intro.pdf.
3. S. Abiteboul, R. Hull, V. Vianu, "Foundations of Databases", Addison Wesley, 1995.
4. J. S. Bendat, and A. G. Piersol, "Random Data: Analysis and Measurement Procedures", 4th Edition, John Wiley & Sons, Inc., NY, USA, 2010.
5. D. C. Montgomery, and G. C. Runger, "Applied Statistics and Probability for Engineers", 5th Edition, John Wiley & Sons, Inc., NY, USA, 2011.
6. Cathy O'Neil, and Rachel Schutt, "Doing Data Science", O'Reilly Media, 2013.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSDS2101P	Fundamentals of Data Science Lab		0	0	2	1
Total Units to be Covered: 10		Total Contact Hours: 30				
Prerequisite(s):	Basics of mathematics, programming		Syllabus version: 1.0			

Course Objectives

1. Learn to collect, clean, and preprocess data from diverse sources for analysis.
2. Understand core statistical concepts to extract valuable insights from data.
3. Gain a foundational understanding of machine learning algorithms and their applications.
4. Develop coding skills to perform data analysis and visualization.

Course Outcomes

CO 1. Know the importance of data analytics in relation to various statistical measures.
CO 2. Employ statistical techniques to extract insights from data.
CO 3. Demonstrate proficiency in using R for data analysis.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	-	1	1	-	-	-	-	-	-	-	-	-	3
CO 2	1	-	-	1	1	-	-	-	-	-	-	-	-	-	3
CO 3	1	2	-	2	1	-	-	-	-	-	-	-	-	-	3
Average	1	0.67	-	1.3	1	-	-	-	-	-	-	-	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

List of Experiments

Experiment no 1	Conduct basic data exploration by calculating summary statistics, creating histograms, and generating scatterplots.
Experiment no 2	Learn data cleaning techniques, including handling missing data, outliers, and data imputation.
Experiment no 3	Perform hypothesis tests, such as t-tests or chi-squared tests, to make inferences about data.
Experiment no 4	Implement simple linear regression to analyze relationships between variables and make predictions.
Experiment no 5	Create a variety of visualizations, including bar charts, line graphs, heatmaps, and box plots.
Experiment no 6	Use clustering algorithms to group similar data points together.
Experiment no 7	Build a random forest model for more advanced classification and regression tasks.
Experiment no 8	Discover frequent item sets and association rules in transactional data.
Experiment no 9	Project 1 (Sentiment analysis)
Experiment no 10	Project 2 (Recommendation systems)

Total Lab hours 30

Textbooks

1. G. Strang, "Introduction to Linear Algebra", 5th Edition, Wellesley-Cambridge Press, USA, 2016.
2. D. C. Montgomery, and G. C. Runger, "Applied Statistics and Probability for Engineers", 5th Edition, John Wiley & Sons, Inc., NY, USA, 2011.

Reference Books

1. Mark Gardener, "Beginning R - The Statistical Programming Language", John Wiley & Sons, Inc., 2013.
2. W. N. Venables, D. M. Smith, and the R Core Team, "An Introduction to R", 2013. Available online at: https://cran.r-project.org/doc/manuals/R-intro.pdf.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSDS3001P	Data Visualization and Interpretation	4	0	0	4
Total Units to be Covered: 06	Total Contact Hours: 60				
Prerequisite(s):	Fundamentals of programming and elementary statistics	Syllabus version: 1.0			

Course Objectives

1. Develop skills to both design and critique visualizations
2. Understand why visualization is an important part of data analysis
3. Understand the various analytical testing techniques
4. Understand the type of data impacts the type of visualization
5. Exploration and exploitation of different visualization tools.

Course Outcomes

Know the importance of data analytics in relation to various statistical measures.

CO 1. Employ descriptive analytic techniques for data modelling.

CO 2. Apply the concept of predictive analytics to various applications.

CO 3. Design interactive dashboard for data visualization.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	-	1	1	-	-	-	-	-	-	-	-	-	3
CO 2	1	-	-	1	1	-	-	-	-	-	-	-	-	-	3
CO 3	1	2	-	2	1	-	-	-	-	-	-	-	-	-	3
CO4	1	-	-	3	1	-	-	-	-	-	-	-	-	-	3
Average	1	.5	-	1.75	1	-	-	-	-	-	-	-	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Exploratory data analysis and visualization **10 Lecture Hours**

Importance of data analysis, Descriptive vs predictive vs prescriptive, Overview of commonly used techniques, Univariate analysis, Distributions, Outlier detection, preprocessing of data, Histograms, Kernel density estimate plots, Box and violin plots, Regression plots, Bar charts, Classification, Association and Segmentation models using Python/R- programming.

Unit II: Exploratory Data Analytics and Charts **15 Lecture Hours**

T-Test, Bi-variate Analysis, Correlations, Crosstabs, Heatmaps, Clustered Matrices Stacked bars, Line charts, Multi-variate Analysis, Trending analysis, Geographical analysis, Maps, Scatter plot, Multi line charts, Area graph, Labs using R/Python.

Unit III: Hypothesis Testing **10 Lecture Hours**

Hypothesis Testing, T-test, Chi-square test, ANNOVA, Clustering Analysis, Cognitive Analytics, create reports based on relationships, Labs using R/Python.

Unit IV: Visualizing Business Intelligence **10 Lecture Hours**

Representing data using graphs and charts- Line chart, Bar, Box plot, Scatter chart, building visualization reports using tableau

Unit V: Dashboard and Scorecards **10 Lecture Hours**

Scorecards and Dashboard creation for data analytics, Dynamic reporting, enhance user interaction using Cognos Analytics

Unit VI: Case Studies **5 Lecture Hours**

Use case studies on real data: Stock prices prediction, Stock market analysis, Election poll analysis.

Total lecture Hours 60

Textbooks

1. Cole Nussbaumer Knaflic, “Storytelling with Data: A Data Visualization Guide for Business Professionals”, Wiley, 2015.
2. Kieran Healy, “Data Visualization: A Practical Introduction”, Princeton University Press, 2018.

Reference Books

1. Steve Wexler, Jeffrey Shaffer, and Andy Cotgreave, “The Big Book of Dashboards”, Wiley, 2017.
2. Stephen Few, “Information Dashboard Design”, Analytics Press, 2013.
3. Colin Ware, “Information Visualization: Perception for Design”, 4th Edition, Morgan Kaufmann Publishers In, 2020.
4. Kieran Healy, “Data Visualization: A Practical Introduction”, Princeton University Press, 2018.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSDS3101P	Data Visualization and Interpretation Lab		0	0	2	1
Total Units to be Covered: 10		Total Contact Hours: 30				
Prerequisite(s):	Fundamentals of programming and elementary statistics		Syllabus version: 1.0			

Course Objectives

1. Develop skills to both design and critique visualizations
2. Understand why visualization is an important part of data analysis
3. Understand the various analytical testing techniques
4. Understand the type of data impacts the type of visualization
5. Exploration and exploitation of different visualization tools.

Course Outcomes

CO1. Know the importance of data analytics in relation to various statistical measures.

CO2. Employ descriptive analytic techniques for data modelling.

CO3. Apply the concept of predictive analytics on various application.

CO4. Design interactive dashboard for data visualization.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	-	1	1	-	-	-	-	-	-	-	-	-	3
CO 2	1	-	-	1	1	-	-	-	-	-	-	-	-	-	3
CO 3	1	2	-	2	1	-	-	-	-	-	-	-	-	-	3
CO4	1	-	-	3	1	-	-	-	-	-	-	-	-	-	3
Average	1	.5	-	1.75	1	-	-	-	-	-	-	-	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment no 1	Introduction to Cognos Insight
Experiment no 2	Personal Analysis
Experiment no 3	What is analysis
Experiment no 4	Data Cube and dimension creation
Experiment no 5	Pivot table creation
Experiment no 6	Introduction to Plotly to visualize data
Experiment no 7	Identification of various types of data in real life
Experiment no 8	Analysing Panel data
Experiment no 9	Project 1 (Time series analysis)
Experiment no 10	Project 2 (Cross sectional data analysis)

Total Lab hours 30

Textbooks

1. Cole Nussbaumer Knaflic, "Storytelling with Data: A Data Visualization Guide for Business Professionals", Wiley, 2015.

2. Kieran Healy, "Data Visualization: A Practical Introduction", Princeton University Press, 2018.

Reference Books

1. Steve Wexler, Jeffrey Shaffer, and Andy Cotgreave, "The Big Book of Dashboards", Wiley, 2017.
2. Stephen Few, "Information Dashboard Design", Analytics Press, 2013.
3. Colin Ware, "Information Visualization: Perception for Design", 4^{th} Edition, Morgan Kaufmann Publishers In, 2020.
4. Kieran Healy, "Data Visualization: A Practical Introduction", Princeton University Press, 2018.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSDS3002P	**Machine Learning and Deep Learning**	4	0	0	4
Total Units to be Covered: 05		**Total Contact Hours: 60**			
Prerequisite(s):	Linear Algebra, Probability	**Syllabus version: 1.0**			

Course Objectives

1. To understand the fundamental concepts and usage of machine learning.
2. To learn ML algorithms to solve various tasks in real life.
3. To understand the need and use of Deep Learning.

Course Outcomes

On completion of this course, the students will be able to

CO 1. Understand types of Machine Learning algorithms, their usage and evaluation.

CO 2. Use Regression for univariate and multivariate analysis.

CO 3. Apply classification and clustering techniques.

CO 4. Understand Deep Learning Techniques and their applications.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	2	2	-	2	-	-	-	-	-	-	-	-	3	2
CO 2	-	2	2	-	2	-	-	-	-	-	-	-	-	3	2
CO 3	-	2	2	-	2	-	-	-	-	-	-	-	-	3	2
CO4	-	2	2	-	2	-	-	-	-	-	-	-	-	3	2
Average	-	2	2	-	2	-	-	-	-	-	-	-	-	3	2

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction **10 Lecture Hours**

The Origins of Machine Learning, Uses and Abuses of Machine Learning, Generalization, Bias and variance, Overfitting vs. Underfitting, Assessing the Success of Learning, Accuracy, Precision, Recall, F-1 Score, Machine Learning Pipeline, Features- selection, reduction, enhancement. Types of Machine Learning Algorithms- Supervised, Unsupervised, Semi-supervised, Active, Incremental, Multi-modal, and Transfer learning. Matching Data to an Appropriate Algorithm.

Unit II: Regression **10 Lecture Hours**

Regression basics: Relationship between attributes using Covariance and Correlation Relationship between multiple variables, Linear Regression, Multiple Linear Regression, Polynomial Regression. Regularization methods- Lasso, Ridge, and Elastic nets, of error measures (ROCR).

Unit III: Classification **15 Lecture Hours**

Probabilistic classifiers and Deterministic Classifiers, Binary and Multi-class Classification, Multi-label Classification, Logistic Regression, Decision Trees, Naive Bayesian Classifier, k-Nearest Neighbor Algorithm, Support Vector Machines, Linear Discriminant Analysis, Ensemble Methods: Bagging, Boosting, Random Forests, Advanced Classification Methods.

Unit IV: Clustering **10 Lecture Hours**

Unsupervised Learning, Cluster Analysis, Partitioning Methods- K-Means, K-Medoids, Spectral Clustering. Hierarchical Methods- Agglomerative, Divisive, BIRCH. Density-Based Methods- DBSCAN, OPTICS.

Unit V: Deep Learning **15 Lecture Hours**

Neural Networks- Perceptron, Back Propagation. Deep Networks- Definition, Motivation, Applications, Restricted Boltzmann Machine, Sparse Auto-encoder, Deep Belief Net, Hidden Markov Model. Convolution Neural Network (CNN)- Basic architecture, Activation functions, Pooling, Handling vanishing gradient problem, Dropout, Greedy Layer-wise Pre-training, Weight initialization methods, Batch Normalization. Different CNN Models- Alex Net, VGG Net, Google Net, Res Net, Dense Net. Graphical Model- Bayes Net, Variational Auto-encoders. Recurrent Neural Network (RNN), Gated RNN, Long short-term memory (LSTM).

Total lecture Hours 60

Textbooks

1. Tom M. Mitchell, “Machine Learning”, McGraw Hill Education, 2017.
2. C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2016.

Reference Books

1. I. Goodfellow, Y. Bengio, and A. Courville, “Deep Learning”, MIT Press, 2016.
2. Oswald Campesato, “Artificial Intelligence, Machine Learning, and Deep Learning”, Mercury Learning & Information, 2020.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSDS3102P	Machine Learning and Deep Learning Lab	0	0	2	1
Total Units to be Covered: 12	Total Contact Hours: 30				
Prerequisite(s):	Python Programming	Syllabus version: 1.0			

Course Objectives

1. To understand the workings of various ML and DL algorithms.
2. To get familiar with various libraries and tools for these algorithms.

Course Outcomes

CO1 Use and understand evaluation metrics for Machine Learning algorithms.

CO2 Apply different regression methods for prediction.

CO3 Apply suitable classification and clustering techniques to ML tasks.

CO4 Understand and apply Deep Learning techniques to solve real life

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	2	1	-	2	-	-	-	-	-	-	-	-	-	3
CO 2	-	2	1	-	2	-	-	-	-	-	-	-	-	-	3
CO 3	-	2	1	-	2	-	-	-	-	-	-	-	-	-	3
CO4	-	2	1	-	2	-	-	-	-	-	-	-	-	-	3
Average	-	2	1	-	2	-	-	-	-	-	-	-	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

List of Experiments

Experiment no 1	Write a program to find Precision, Accuracy, Recall & F1 Score.
Experiment no 2	Write a program to perform Linear Regression.
Experiment no 3	Write a program to perform Multiple Linear Regression.
Experiment no 4	Write a program to showcase Lasso, Ridge and Elastic Net Regularization.
Experiment no 5	Write a program to perform Logistic Regression.
Experiment no 6	Write a program to perform classification using k-NN.
Experiment no 7	Write a program to perform classification using Support Vector Machine.
Experiment no 8	Write a program to perform classification using Linear Discriminant Analysis.
Experiment no 9	Write a program to perform k-Means Clustering.
Experiment no 10	Write a program to build a simple neural network.
Experiment no 11	Write a program to build a Convolutional Neural Network.
Experiment no 12	Write a program to build an LSTM model.

Total Lab hours 30

Textbooks

1. Aurélien Géron, “Hands-on ML with Scikit-Learn, Keras & TensorFlow”, 3rd Edition, O’Reilly Media, 2022.
2. Shailendra Kadre, Venkata Reddy Konasani, “Machine Learning and Deep Learning Using Python and TensorFlow”, McGraw Hill, 2021.

Reference Books

1. Seth Weidman, “Deep Learning from Scratch: Building with Python from First Principles”, Shroff/O'Reilly, 2019.
2. Sebastian Raschka, Yuxi (Hayden) Liu, Vahid Mirjalili, and Dmytro Dzhulgakov, “Machine Learning with PyTorch and Scikit-Learn”, Packt Publishing, 2022.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSEG4034P	Computational Linguistic and Natural Language Processing	4	0	0	4
Total Units to be Covered: 05	Total Contact Hours: 60				
Prerequisite(s):	Machine Learning & Theory of Automata	Syllabus version: 1.0			

Course Objectives

1. To understand fundamentals of linguistics and language processing.
2. To study various approaches to construct a natural language processing system like information retrieval system can be developed

Course Outcomes

On completion of this course, the students will be able to

CO1. Explain basic concepts of natural language processing principles and approaches.

CO2. Demonstrate steps involved in development of information retrieval system with advancements.

CO3. Discuss the emerging applications of NLP.

CO4. Discuss the empirical, statistical, and classical methods in NLP and its usage.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	1	-	-	-	-	-	-	-	-	-	-	-	3
CO 2	1	-	1	-	-	-	-	-	-	-	-	-	-	-	3
CO 3	1	-	1	-	-	-	-	-	-	-	-	-	-	-	3
CO4	1	-	1	-	-	-	-	-	-	-	-	-	-	-	3
Average	1	-	1	-	-	-	-	-	-	-	-	-	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Classical Approaches of NLP **10 Lecture Hours**

Introduction, Classical approaches to natural language processing, Approaches to natural language processing, understanding linguistics, Level 1: Morphology, Level 2: Syntax, Level 3: Semantics, Level 4: Pragmatics, Understanding linguistics, Traditional approach, Example: Automatic summarization using NLP, Drawbacks, Text processing, What Is text processing? Text analysis vs. Text mining vs. Text analytics, Tools and methodologies: Statistical methods, Tools and methodologies: Text classification, Tools and methodologies: Text extraction, Tools and methodologies: Example, Scope of text analysis/processing, Importance of text analysis, Working principles of text analysis, Data gathering, Data preparation, Data preparation steps, Data analysis, Evaluation of text classification process, Text extraction, Analysis in test extraction, Evaluation of text extraction process, Text analysis APIs, Levels of NLP, Lexical analysis, Pre-processing activity, POS tagging, Syntactic parsing, Types of parsing, Derivation logic, Grammar, Semantic analysis, Semantic analysis elements, Representation in semantic analysis, Natural language generation, NLP vs NLG, History of NLG, Working principle of natural language generation, Limitations in natural language generation.

Unit II: Empirical Approaches **10 Lecture Hours**

Corpus creation, Corpus linguistics, Types of corpora, Lexicographical implementations in corpora, Timeline of corpus linguistics, Usage areas of corpora, Traits of a good text corpus, Annotations in text corpus, NLP task-specific training corpora, Data sets used for natural language processing, Treebank annotation, Linguistic description layers, Areas using text annotations, Usage of annotations and corpora, Kinds of annotations, Annotation semantic labels, Annotations in machine learning, Annotation development cycle, Model creation, Create annotations, Training and testing the algorithms, Result evaluation, Revision of the model, Tree banks and its construction, Need for tree bank, Types of tree bank corpus, Phrase structured vs dependency structured tree bank, Fundamental statistical techniques, Problems of the traditional approach, How statistics helps, Problems of the traditional approach and how statistics helps, Hidden Markov model, Maximum entropy Markov model, Conditional random field model, Support vector machine, N-GRAM, Genetic algorithm, POS Tagging, Word sense disambiguation, POS tag and Tagsets, Types of POS taggers, Markovian model, Hidden Markov model, POS tagging using HMM.

Unit III: Statistical Approaches 15 Lecture Hours

Parsing, Statistical parsing, Approaches to parsing, Statistical approach, Lexicalized statistical parsing, Top-down parsing, Bottom-up parsing, Left corner parsing method, Statistical parsing: Probabilistic parser, Multiword expressions, Features of MWE, Types of multi word expressions, Multi word verbs, Word similarity and text similarity, Text similarity methods, Jaccard similarity, K-means, Cosine similarity, Word Mover's distance, Variational auto encoders, Pre-trained sentence encoders, Bidirectional Encoder Representations from Transformers (BERT) with cosine distance, Word sense disambiguation, Complications in WSD, Methods in WSD, Evaluation of WSD, History of speech recognition technology, Working principle in voice recognition, Major leaders in speech recognition and voice assistant, Amazon Alexa, Microsoft Cortana, Google Assistant, Machine translation, Rule-based machine translation, Statistical machine translation, Rule-based MT vs. statistical MT, Working principle of SMT, Challenges with statistical machine translation.

Unit IV: Applications of Natural Language Processing 15 Lecture Hours

Information retrieval, Information retrieval in NLP, IR development, Model types, Model types: Mathematical basis model, Problems with NLP in information retrieval, NLP in information retrieval, IR evaluation metrics, Information Retrieval (IR) model and types, Design features of IR systems, Design features of IR systems, Question answering systems, QA system architecture, QA system types, Text based QA systems, Factoid question answering system, Web based question answering system, Information retrieval or information extraction based QA systems, Restricted domain question answering, Rule based question answering systems, Information extraction, Working of information extraction, Information extraction applications, Chunking, Representing chunks: Tags vs trees, Report generation, Text report specifications, Features of reports, Report generation process, Usage of NLP text in report generation, Ontology construction, Ontology classifications and process, Why ontology and its advantages, Ontology components, Levels of formality, Ontology construction approaches, Ontology construction.

Unit V: Emerging Applications of Natural Language Generation in Information Visualization, Education, and Health Care **10 Lecture Hours**

Multimedia presentation generation, Focus points to add multimedia in NLG, Text generation: Meaning representation, Text generation: Document structure design, Text generation: Linguistic style control, Document layout, Layout and meaning representation, Layout style and wording representation, Image style and meaning representation, Image and wording usage, Scripted dialogue, Language interfaces for intelligent tutoring systems, CIRCSIM-Tutor, CIRCSIM-Tutor architecture, data presentation and process cycle, AUTOTUTOR, AUTOTUTOR architecture and process, ATLAS Andes, Andes system architecture and design, Pedagogical considerations in Andes, WHY2-ATLAS, Why 2 Atlas architecture and process, Argumentation for healthcare consumers, CDS architecture and processing, NLP for CDS scope, NLP models, Building blocks of NLP - CDS, Data based evidence collection: Summarization, Applications of NLP in healthcare, Sentiment analysis and subjectivity, Difficulties in sentiment analysis, Document level sentiment classification, Sentence level sentiment classification, Lexicon, Feature-based sentiment analysis, Opinion summarization.

Total lecture Hours 60

Textbooks

1. Christopher D. Manning, and Hinrich Schutze, “Foundations of Natural Language Processing”, 6th Edition, The MIT Press Cambridge, Massachusetts London, England, 2003.

2. Daniel Jurafsky and James H. Martin “Speech and Language Processing”, 3rd Edition, Prentice Hall, 2009.

Reference Books

1. Nitin Indurkhya, and Fred J. Damerau “Handbook of Natural Language Processing”, 2nd Edition, CRC Press, 2010.

2. James Allen, “Natural Language Understanding”, 8th Edition, Pearson Publication, 2012.

3. Christopher D. Manning and Hinrich Schütze, “Foundations of Statistical Natural Language Processing”, 2nd edition, MIT Press Cambridge, MA, 2003.

4. Hobson Lane, Cole Howard, and Hannes Hapke, “Natural language processing in action”, Manning Publications, 2019.

5. Alexander Clark, Chris Fox, and Shalom Lappin, “The Handbook of Computational Linguistics and Natural Language Processing”, Wiley-Blackwell, 2012.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSEG4134P	Computational Linguistics and Natural Language Processing Lab	0	0	2	1
Total Units to be Covered: 10	Total Contact Hours: 30				
Prerequisite(s):	a. Basic Arithmetic, Statistics & Probability b. Data Structures & Algorithm c. Knowledge of Automata Theory & Compiler Design	Syllabus version: 1.0			

Course Objectives

The objectives of this course are:

1. To introduce the concept of Natural Language Understanding & Natural Language Generation.
2. To develop the concept of statistical and probabilistic approach of language modelling.
3. To extend the knowledge of Large Language Model.
4. To enrich the knowledge with different corpuses and different tools being used for machine translation.
5. To provide programming skills necessary for processing natural language.

Course Outcomes

On completion of this course, the students will be able to

CO1: Understand the techniques in NLP

CO2: To understand and execute the Large Language Model

CO3: To comprehend and execute the natural language generation

CO4: To implement the machine translation for various application

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3

CO 1	1	-	-	1	1	-	-	-	-	-	-	-	-	-	3
CO 2	1	-	-	1	1	-	-	-	-	-	-	-	-	-	3
CO 3	1	2	-	2	1	-	-	-	-	-	-	-	-	-	3
CO4	1	-	-	3	1	-	-	-	-	-	-	-	-	-	3
Average	1	.5	-	1.75	1	-	-	-	-	-	-	-	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

List of Experiments

EXPERIMENT-1:

Title: Installing various packages required for analytics in python and write code in Python for following program

2. Write code to load CSV file containing information about employee of a company in python and draw graph showing average salary department wise.

EXPERIMENT-2:

Title: Text Retrieval

3- Connect to Twitter account and Extract first 100 tweets from it in a file.
4- Study and Implementation of Processing text(Word and Sentence Tokenization)

EXPERIMENT-3:

Title: Processing Data

3. Python code to read a text document and perform basic pre-processing techniques on the text like tokenization, stop-word-removal, lemmatization etc..
4. Study and Implementation of Morphological analysis.

EXPERIMENT-4:

Title: Do text mining on extract data and Accessing text corpus

3. Calculate word count of a given specific document and Show top 10 frequent words with their frequency and Create world cloud and show graphically.
4. Study and Implementation of NER (Name Entity Recognition)

EXPERIMENT-5:

Title: POS-Tagging and Tagging and Parsing

3. Categorizing and tagging words in Twitter Data.
4. Study and implementation of POS Tagging and Chunking in a sentence

EXPERIMENT-6:

Title: Language Processor

2. Implement N–Gram Language Mode and Smoothing.

EXPERIMENT-7:

Title: Do sentimental analysis

3. Analysis of Sentiment and Subjectivity
4. Implement sentimental analysis on IMDB Movie Reviews Dataset.

EXPERIMENT-8:

Title: Do Text Summarization

3. Analysing Meaning of Sentences
4. Implement Text Summarization on IMDB Movie Reviews Dataset.

EXPERIMENT-9 & 10:

Title: Mini-Project on NLP

Implement Mini-Project on NLP applications.

Total Lab hours 30

Textbooks

1. Daniel Jurafsky, and James H. Martin, "Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition", 2nd Edition, Pearson, 2013.

Reference Books

1. David A. Grossman, and Ophir Frieder, "Information Retrieval: Algorithms and Heuristics", Springer, 2004.
2. Akshar Bharati, Vineet Chaitanya, and Rajeev Sangal, "Natural Language Processing: A paninian perspective", Prentice Hall, New Delhi, 1995.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	**Quiz & Viva**	**Performance & Lab Report**
Weightage (%)	50 %	50 %

Course Code	Course name		L	T	P	C
CSDS4001P	Generative Artificial Intelligence		3	0	0	3
Total Units to be Covered: 04	Total Contact Hours: 45					
Prerequisite(s):	Machine learning and deep learning	Syllabus version: 1.0				

Course Objectives

1. To introduce the Generative models.
2. To enrich the knowledge in various methods of GenAI and their application in different research applications.
3. To extend the knowledge of GenAI to solve the current problems of AI.
4. To provide practical programming skills necessary for constructing a GenAI.

Course Outcomes

On completion of this course, the students will be able to

CO 1. Comprehend different GenAi techniques.

CO 2. Use concepts of GenAI to different applications and areas.

CO 3. Builds new GenAI method with explainability.

CO 4. Aware from responsibilities and social impact of GenAI.

CO 5. Build GenAi methodology to improve the performance of any problem area.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	2	2	-	2	-	-	-	-	-	-	-	-	-	3
CO 2	-	2	2	-	2	-	-	-	-	-	-	-	-	-	3
CO 3	-	2	2	-	2	-	-	-	-	-	-	-	-	-	3
CO4	-	2	2	-	2	-	-	-	-	-	-	-	-	-	3
Average	-	2	2	-	2	-	-	-	-	-	-	-	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: **15 Lecture Hours**

Introduction to GenAI, Encoder-decoder, Autoencoder, Variational autoencoder, Txt2Txt translation, Txt2Img translation, Img2Txt translation, Txt2Voice translation

Unit II: **15 Lecture Hours**

Text summarization, Text translation, Question-answer bots, Generative adversarial network, Auto-regressive methods, Attention mechanism, Soft attention, Hard attention, Self-attention

Unit III: **15 Lecture Hours**

Transformers and its variants, Large language models, GPT-3, ChatGPT, Prompt engineering, Introduction to LangChain

Unit IV: **15 Lecture Hours**

Explainability, Responsibility, Ethical AI, DeepFake,

Total lecture Hours 45

Textbooks

1. I. Goodfellow, Y. Bengio, and A. Courville, “Deep Learning”, MIT Press, 2016.
2. Jakub Langr, and Vladimir Bok, “GANs in Action Deep Learning with Generative Adversarial Networks”, Manning, 2019.

Reference Books

1. Sudharsan Ravichandran, “Hands on Deep learning algorithms with python”, Packt Publishing Limited, 2019.
2. Navin K Manaswi, “Generative Adversarial Networks with Industrial Use Cases”, BPB Publisher, 2020.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSDS4101P	Generative Artificial Intelligence Lab	0	0	2	1
Total Units to be Covered: 12	Total Contact Hours: 30				
Prerequisite(s):		Syllabus version: 1.0			

Course Objectives

1. To introduce the Generative models.
2. To enrich the knowledge in various methods of GenAI and their application in different research applications.
3. To extend the knowledge of GenAI to solve the current problems of AI.
4. To provide practical programming skills necessary for constructing a GenAI.

Course Outcomes

On completion of this course, the students will be able to

CO 1. Comprehend different GenAi techniques.

CO 2. Use concepts of GenAI to different applications and areas.

CO 3. Builds new GenAI method with explainability.

CO 4. Aware from responsibilities and social impact of GenAI

CO 5. Build GenAi methodology to improve the performance of any problem area.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	2	3	-	2	-	-	-	-	-	-	-	-	-	3
CO 2	-	2	3	-	2	-	-	-	-	-	-	-	-	-	3
CO 3	-	2	3	-	2	-	-	-	-	-	-	-	-	-	3
CO4	-	2	3	-	2	-	-	-	-	-	-	-	-	-	3
Average	-	2	3	-	2	-	-	-	-	-	-	-	-	-	3

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment no 1	WAP to create an autoencoder for structured data.
Experiment no 2	WAP to create an autoencoder for Txt2Txt translation.
Experiment no 3	WAP to create variational autoencoder for Img2Txt translation.
Experiment no 4	Develop a GenAI methodology for Text summarization.
Experiment no 5	Develop GAN method to augment images.
Experiment no 6	Develop GAN variant method for language translation
Experiment no 7	WAP to use different attention mechanism for image recognition
Experiment no 8	Compare various attention mechanism performance in language translation
Experiment no 9	Develop a Transformer method for time series data
Experiment no 10	Develop transformer method to analyze signal data
Experiment no 11	Develop a DeepFake method for images
Experiment no 12	Develop a DeepFake method for video data.

Total Lab hours 30

Textbooks

1. I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning", MIT Press, 2016.

2. Rafael Valle, “Hands-On Generative Adversarial Networks with Keras”, Packt Publishing, 2019.

Reference Books

1. Sudharsan Ravichandran, “Hands on Deep learning algorithms with python”, Packt Publishing Limited, 2019.
2. Josh Kalin, “Generative Adversarial Networks Cookbook”, Packt Publishing Limited, 2018.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

8. IOT Track

Course Code	Course name	L	T	P	C
CSGG2110P	Introduction to IoT, Sensors and Microcontrollers	4	0	0	4
Total Units to be Covered: 05	Total Contact Hours: 60				
Prerequisite(s):	Computer Programming	Syllabus version: 1.0			

Course Objectives

1. Understand the basics of Internet of Things
2. Learn to program the micro controllers boards
3. Understand the working of sensors and programming them
4. Implement the knowledge on a real world project

Course Outcomes

On completion of this course, the students will be able to:

CO1. Students will be able to understand the fundamental concepts and principles of the Internet of Things (IoT) and its applications.

CO2. Students will be able to demonstrate proficiency in programming microcontrollers and interfacing sensors for IoT applications.

CO3. Students will be able to design and develop IoT applications using appropriate hardware and software components.

CO4. Students will be able to apply principles of sensor instrumentation and data acquisition for IoT systems.

CO5. Students will be able to analyze and solve challenges in IoT deployment, security, and scalability.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	2	-	-	-	1	-	-	-	-	1	2

CO 2	-	-	-	-	2	-	-	-	1	-	-	-	-	3	2
CO 3	-	-	-	-		-	-	-		-	-	-	-		1
CO4	-	-	-	-		-	-	-		-	-	-	-	2	1
CO5	-	-	-	-	2	-	-	-	2	-	-	-	-	2	2
Average	-	-	-	-	1.2	-	-	-	.8	-	-	-	-	1.6	1.6

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: IoT Introduction **12 Lecture Hours**

I. Concepts and Definitions of The Internet of Things (IoT).
II. History of IoT
III. IoT Market Study
IV. Requirements Functionalists and structure of IoT.
V. IoT enabling technologies.
VI. IoT Architecture.
VII. Major component of IoT (Hardware & Software).
VIII. Challenges in IoT

Unit II: IoT Programming Platforms **12 Lecture Hours**

I. Microcontrollers Basics
II. Introduction to Arduino and various variants
III. Overview of Breadboard, Jumper Wires, LEDs, LCD, Male header Pins, Potentiometers, Buttons
IV. Arduino Pin Description
V. Digital vs Analog vs PWN Pins
VI. Arduino Programming basics
VII. Blinking LED and playing Buzzer

VIII. Communication with SPI and I2C

IX. Displaying on LCD and OLED

Unit III: Sensors and Actuators with Arduino **10 Lecture Hours**

I. Temperature and Humidity Sensor, Distance Ranging Sensor

II. Acceleration Sensor, Tilt Switch Module, SD Card Module, JoyStick, RF Module, BT Module

III. IR and PIR Sensor, Force Sensors, Gas Sensor, Water Flow sensor, Sound sensor module

IV. Medical Sensors: ECG, Pulse and Heart Rate Sensors, Fingerprint, Touch,

V. Soil and pH sensor

VI. GPS Module and GSM Module

VII. Servo Motor, DC Motors programming with Arduino

Unit IV: Sensor and Instrumentation **14 Lecture Hours**

I. Diode, Resistors, Capacitors, Transistor

II. Data Acquisition System

III. Sensor Classifications

IV. Sensor Characteristics:- Transfer function, Calibration, Accuracy, Reliability

V. Physical Principle of Sensing: Capacitance, Magnetism, Resistance, Induction, Piezo electric effect, Hall effect, Light

VI. Dynamic Models of Sensor Elements

Unit V: Popular Microcontrollers for IoT **12 Lecture Hours**

I. ESP32 Introduction and variants

II. ESP32 Pin Desciption

III. ESP32 as AP, WebServer

IV. ESP32 connection to Internet with MQTT

V. Raspberry Pi Introduction and variants

VI. Arduino vs Raspberry Pi vs ESP32

VII. Cloud Platforms for IoT:- ThingSpeak, AWS IoT Core, Azure IoT Hub, GCP IoT Core

Total lecture Hours 60

Textbooks

1. Jermy Blum, “Exploring Arduino”, Wiley, 2013.
2. Sudip Misra, Anandarup Mukherjee, and Arijit Roy, “Introduction to IoT”, Cambridge University Press, 2022.
3. D. Patranabis, “Sensors and Transducers”, 2nd Edition, PHI Learning, 2003.

Reference Books

1. Jacob Fraden, "Handbook of Modern Sensors: Physics, Designs, and Applications”, Springer, 4th Edition, 2014.
2. Dr Kamlesh Lakhwani, Dr Hemant Kumar Gianey, Joseph Kofi Wireko, and Kamal Kant Hiran, “Interent of Things”, BPB Publisher, 2020.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSIS2112P	**Introduction to IoT, Sensors and Microcontrollers Lab**	0	0	2	1
Total Units to be Covered: 10		**Total Contact Hours: 30**			
Prerequisite(s):	Computing Programming	**Syllabus version: 1.0**			

Course Objectives

1. Understand the basics of Internet of Things
2. Learn to program the micro controller's boards
3. Understand the working of sensors and programming them
4. Implement the knowledge on a real world project

Course Outcomes

CO1. Students will be able to understand the fundamental concepts and principles of the Internet of Things (IoT) and its applications.
CO2. Students will be able to demonstrate proficiency in programming microcontrollers and interfacing sensors for IoT applications.
CO3. Students will be able to design and develop IoT applications using appropriate hardware and software components.
CO4. Students will be able to apply principles of sensor instrumentation and data acquisition for IoT systems.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	2	-	-	-	1	-	-	-	-	1	2
CO 2	-	-	-	-	2	-	-	-	1	-	-	-	-	3	2
CO 3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1
CO4	-	-	-	-	-	-	-	-	-	-	-	-	-	2	1
Average	-	-	-	-	2	-	-	-	.5	-	-	-	-	1.5	1.5

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

List of Experiments

Experiment 1: Introduction to Arduino:

- Arduino IDE Installation.
- Setting up an Arduino board, writing a simple program to blink an LED, and understanding the basics of the Arduino IDE.
- Blink the inbuilt IDE with frequency as per the last two digit of your rollNo.
- Write the Arduino code to read a number from the serial monitor and display the inbuilt LED equally number of times.

Experiment 2: Sensor Interfacing:

- Write the Arduino code to read potentiometer value from the analog pins and display it on the Serial Monitor.
- Interfacing a temperature and humidity sensor (DHT11) with the Arduino board to measure the temperature and humidity.
- Interface the ultrasonic sensor(US) with Arduino and read the distance.
- WAP in Arduino to play a buzzer if the distance read from the US sensor is less than 50cm

Experiment 3: Display Data on LCD, OLED:

- Using a Passive Infrared (PIR) sensor to detect motion and trigger an output, such as an LED or buzzer.

- Implementing an Ultrasonic sensor to measure distance and display the results on an LCD.
- Interface the OLED display with Arduino and display your name or rollNo.
- Interface the DHT11 sensor with Arduino and display the temperature and humidity on OLED display.

Experiment 4: IoT Basic Communication:

- Establishing communication between two Arduino boards using the SPI or I2C protocol.
- Data Logging with SD Card: Collect any sensor data and store it on an SD card for future analysis.

Experiment 5: Data Processing:

- Use the Joystick to move any alphanumeric character over the OLED display.
- Read 10 consecutive readings from the US Sensor and display basic statistics on it over the OLED display.
- Display moving average of 10 consecutive readings from the US sensor over the OLED display.

Experiment 6: GPS Module:

- Interface a GPS module with Arduino, read GPS data such as latitude and longitude, and display the data on an LCD screen or serial monitor.
- Utilize a GPS module and Arduino to measure the speed of a moving object or vehicle by calculating the distance traveled over time using GPS coordinates.

The speed data can be displayed on an LCD or sent wirelessly for further analysis.

Experiment 7: Bluetooth Communication:

- Interface a Bluetooth module with the Arduino board and transfer data to a PC.
- With the help of Bluetooth module control the brightness of the Arduino inbuilt LED.

Experiment 8: Medical Sensor:

- Write Arduino code to read PPG data and calculate the heart beat over a period of 3 minutes.
- Write Arduino code to read ECG data and calculate the heart beat over a period of 3 minutes.

Experiment 9: Motor Control:

- Control the movement of a servo motor with Arduino. Rotate the servo motor left by 45 degree and right by 90 degree and repeat the pattern.
- With the help of motor driver shield, interface and control the speed of the DC motor by entering various values through serial monitor.

Experiment 10: RFID Tag Module:

- Read and display the unique identification (UID) of RFID tags using an Arduino and an RFID module.
- For the use case of an inventory management system using Arduino and RFID technology. Use RFID tags to label and track few items. The Arduino can read

the RFID tags and update the inventory database accordingly, providing real-time information on item availability and location.

Total Lab hours 30

Textbooks

1. Jermy Blum, "Exploring Arduino", Wiley, 2013.
2. Sudip Misra, Anandarup Mukherjee, and Arijit Roy, "Introduction to IoT", Cambridge University Press, 2022.
3. D. Patranabis, "Sensors and Transducers", 2nd Edition, PHI Learning, 2003.

Reference Books

1. Jacob Fraden, "Handbook of Modern Sensors: Physics, Designs, and Applications", Springer, 4th Edition, 2014.
2. Dr Kamlesh Lakhwani, Dr Hemant Kumar Gianey, Joseph Kofi Wireko, and Kamal Kant Hiran, "Interent of Things", BPB Publisher, 2020.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSIS3019P	**IoT Network Architecture and Communication Protocols**	4	0	0	4
Total Units to be Covered: 05	**Total Contact Hours: 60**				
Prerequisite(s):	Introduction to IoT Computer Networks Fundamentals	**Syllabus version: 1.0**			

Course Objectives

1. Understand IoT Network Architectures.
2. Explore Communication Protocols.
3. Implement Network Security Measures:
4. Ensure Interoperability and Integration:

Course Outcomes

On completion of this course, the students will be able to:

CO1. Students will be able to understand the fundamental principles and concepts of IoT network architecture.
CO2. Students will gain knowledge of various communication protocols and wireless protocols in IoT and effectively select and utilize the appropriate protocols for different IoT applications.
CO3. Students will be able to demonstrate proficiency in configuring and managing IoT networks, including network monitoring, diagnostics.
CO4. Students will be able to evaluate and implement appropriate security measures and protocols to safeguard IoT networks from potential threats and vulnerabilities.
CO5. Students will be able to explore IoT integration and interoperability concepts enabling seamless data exchange and interoperability between diverse IoT devices and systems.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3

CO 1	-	-	-	-	2	-	-	-	1	-	-	-	-	1	2
CO 2	-	-	-	-	2	-	-	-	1	-	-	-	-	3	2
CO 3	-	-	-	-		-	-	-		-	-	-	-		1
CO4	-	-	-	-		-	-	-		-	-	-	-	2	1
CO5	-	-	-	-	2	-	-	-	2	-	-	-	-	2	2
Average	-	-	-	-	1.2	-	-	-	.8	-	-	-	-	1.6	1.6

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

Syllabus

Unit I: Introduction to IoT Network Architecture **12 Lecture Hours**

I. Overview of Computer Network
II. OSI Model Brief
III. IoT Reference Model by IoT World Forum
IV. Serial vs Parallel communication
V. SPI, I2C, USB, CAN, RS232 Interfaces
VI. Concepts and fundamentals of IoT network architecture
VII. IoT network topologies and architectures
VIII. Edge computing and fog computing in IoT
IX. IoT network scalability and manageability
X. Security considerations in IoT network architecture

Unit II: IoT Communication Protocols **12 Lecture Hours**

I. IoT Data Protocols and Formats
   a. JSON (JavaScript Object Notation) for IoT data exchange
   b. XML (eXtensible Markup Language) for IoT data representation
   c. CBOR (Concise Binary Object Representation) for efficient IoT data encoding

II. MQTT (Message Queuing Telemetry Transport) protocol
III. CoAP (Constrained Application Protocol) for IoT
IV. AMQP (Advanced Message Queuing Protocol)
V. HTTP and RESTful APIs in IoT
VI. WebSocket protocol for real-time communication
VII. Modbus protocol for industrial IoT communication

Unit III: Wireless Communication Protocols **12 Lecture Hours**

I. Physical Layer, Modulation and demodulation
II. Radio Frequency Spectrum
III. RF Spectrum for communication
IV. Transmitting data with radio waves
V. Signal distortion and noise
VI. Bluetooth and Bluetooth Low Energy (BLE) in IoT
VII. Zigbee protocol for low-power wireless networks
VIII. LoRaWAN (Long Range Wide Area Network) for IoT
IX. Medium Access Control layer
X. IEEE 802.15.4 Protocol

Unit IV: IoT Security and Privacy Protocols **12 Lecture Hours**

I. Authentication and access control in IoT
II. Encryption and secure communication protocols
III. DTLS (Datagram Transport Layer Security) for IoT
IV. IPSec (Internet Protocol Security) for secure IoT communication

Unit V: IoT Network Management and Monitoring **12 Lecture Hours**

I. SNMP (Simple Network Management Protocol) in IoT
II. IoT device provisioning and management
III. Over-the-Air (OTA) updates for IoT devices

Total lecture Hours 60

Textbooks

1. MIsha Dohler, "Internet of Things: Architectures, Protocols, and Standards".
2. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, Robert Barton, and Jerome Henry, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", Cisco Press, 2017.

Reference Books

1. Yan Zhang, Laurence T. Yang, and Huansheng Ning, "Wireless Communications and Networks for the Internet of Things", Auerbach Publications, 2019.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSIS3119P	IoT Network Architecture and Communication Protocols Lab		0	0	2	1
Total Units to be Covered: 10		Total Contact Hours: 30				
Prerequisite(s):	Computing Programming Introduction to IoT		Syllabus version: 1.0			

Course Objectives

1. Understand IoT Network Architectures.
2. Explore Communication Protocols.
3. Implement Network Security Measures:
4. Ensure Interoperability and Integration:

Course Outcomes

On completion of this course, the students will be able to:

CO1. Students will be able to understand the fundamental principles and concepts of IoT network architecture.

CO2. Students will gain knowledge of various communication protocols and wireless protocols in IoT and effectively select and utilize the appropriate protocols for different IoT applications.

CO3. Students will be able to demonstrate proficiency in configuring and managing IoT networks, including network monitoring, diagnostics.

CO4. Students will be able to evaluate and implement appropriate security measures and protocols to safeguard IoT networks from potential threats and vulnerabilities.

CO5. Students will be able to explore IoT integration and interoperability concepts enabling seamless data exchange and interoperability between diverse IoT devices and systems.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	2	-	-	-	-	-	-	-	2	1	2
CO 2	-	-	-	-	2	-	-	-	-	-	-	-	2	1	2
CO 3	-	-	-	-	2	-	-	-	-	-	-	-	2	1	2
CO4	-	-	-	-	2	-	-	-	-	-	-	-	2	1	2
CO5	-	-	-	-	2	-	-	-	-	-	-	-	2	2	2
Average	-	-	-	-	2	-	-	-	-	-	-	-	2	1.2	2

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Introduction to ESP8266 family of microcontrollers:

- ESP8266 Architecture and Pin Description
- ESP32, ESP8266 variants
- Using ESP32 scan all the Wi-Fi network available and print their Wi-Fi signal strength and other information.

Experiment 2: Sensor Interfacing:

- Write the ESP32 code to read potentiometer value from the analog pins and display it on the Serial Monitor.
- Interfacing a temperature and humidity sensor (DHT11) with the ESP32 to measure the temperature and humidity.
- Interface the ultrasonic sensor (US) with ESP32 and read the distance.

Experiment 3: ESP8266 Web Server:

- Create a webserver with ESP8266 with login functionality.
- Create ESP8266 as AP and connect 2-3 WiFi devices to it.
- WAP to display WiFi signal strength on a OLED display using ESP8266.

Experiment 4: IoT Data Protocol:

- Connect two ESP8266 boards and transfer sensor data from one to the other using JSON format.
- Connect two ESP8266 boards and transfer sensor data from one to the other using XML format.

Experiment 5: Wireless control:

- Write the Arduino code for ESP8266 to measure soil moisture and pH levels using a soil moisture sensor and a pH sensor.
- Write the Arduino code for ESP8266 to rotate the servo motor by 'n' degree through the web interface.

Experiment 6: UDP Protocol:

- Set up the ESP32 as a Wi-Fi access point and configure it to act as a UDP server. Create a UDP socket on the ESP32 and bind it to a specific port for incoming data. In the main loop, continuously listen for UDP packets from connected clients. Upon receiving a UDP packet, extract and process the data.

Experiment 7: ZigBee Protocol:

- Understand ZigBee Protocol stack and architecture and the boards and devices available.
- Set up a simple wireless communication system using Zigbee modules. Connect two Arduino boards with Zigbee modules and program them to establish a wireless communication link. Send and receive data between the two modules to demonstrate the basic functionality of the Zigbee protocol.

Experiment 8: ZigBee Networking:

- Create a sensor network using Zigbee modules and Arduino boards. Connect multiple sensor nodes to an Arduino board, each equipped with a Zigbee module. Program the nodes to collect data from sensors and transmit it to a central coordinator node. Receive and process the sensor data at the coordinator node to showcase the ability of Zigbee to form a robust and scalable network.
- Build a Zigbee mesh network using multiple Arduino boards and Zigbee modules. Configure the network with a coordinator node and several router nodes. Program the nodes to transmit data in a multi-hop manner, allowing the data to traverse through multiple nodes to reach the coordinator. Demonstrate the self-healing and self-organizing capabilities of Zigbee mesh networking by dynamically adding or removing nodes from the network.

Experiment 9: IoT connectivity with Cloud:

- IoT Connectivity: Connecting the ESP32 board to the internet using the ESP32 module and MQTT protocol.
- Remote Sensing with GSM: Sending any sensor data to a cloud platform and ThingSpeak.

Experiment 10: Node Red and Security:

- Real-Time Data Visualization: Creating a real-time data visualization dashboard using Node-RED to monitor sensor readings.
- IoT Security: Implementing security measures, such as encryption and authentication, to secure IoT communications and data.

Total Lab hours 30

Textbooks

1. MIsha Dohler, "Internet of Things: Architectures, Protocols, and Standards",
2. David Hanes, Gonzalo Salgueiro, Patrick Grossetete, and Robert Barton, "IoT Fundamentals: Networking Technologies, Protocols, and Use Cases for the Internet of Things", Cisco Press, 2017.

Reference Books

1. Yan Zhang, Laurence T. Yang, and Huansheng Ning, "Wireless Communications and Networks for the Internet of Things", Auerbach Publication, 2019.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSIS3020P	Industrial IoT and ARM based Embedded Programming	4	0	0	4
Total Units to be Covered: 05		Total Contact Hours: 60			
Prerequisite(s):	Introduction to IoT Programming in C	Syllabus version: 1.0			

Course Objectives

1. Understand ARM Architecture and Embedded Programming.
2. Explore Industrial IoT (IIoT) and IIoT Applications.
3. Learn about Industrial Automation and SCADA Systems:
4. Understand Medical IoT and Healthcare Applications

Course Outcomes

On completion of this course, the students will be able to:

CO1. Students will be able to develop Proficiency in ARM Architecture and Embedded Programming.

CO2. Students will be able to program ARM controllers for sensor interfacing and data processing in embedded C.

CO3. Students will be able to apply Industrial IoT Principles and Technologies.

CO4. Students will be able to explore IIoT Applications in Different Industries.

CO5. Students will be able to understand Medical IoT and Healthcare Applications:

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	2	-	-	-	1	-	-	-	-	1	2
CO 2	-	-	-	-	2	-	-	-	1	-	-	-	-	3	2
CO 3	-	-	-	-		-	-	-	-	-	-	-	-		1
CO4	-	-	-	-		-	-	-	-	-	-	-	-	2	1
CO5	-	-	-	-	2	-	-	-	2	-	-	-	-	2	2

Average	**-**	**-**	**-**	**-**	1.2	**-**	**-**	**-**	.8	**-**	**-**	**-**	**-**	1.6	1.6

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: ARM Architecture **12 Lecture Hours**

I. ARM Architecture Overview
II. 32 bit MC benefits
III. ARM vs Intel
IV. Bus and Register Structures
V. Pins and Ports
VI. ARM based processor family
VII. ARM Cortex M Series
VIII. Vector Tables

Unit II: Programming ARM boards **12 Lecture Hours**

I. STM32 Boards based on ARM
II. GPIO Programming
III. USART Protocol
IV. USART Receiver and Transmitter driver
V. Developing ADC Drivers
VI. System Tick (SysTick) Timer
VII. General Purpose Timers in ARM
VIII. Interrupt Programming for GPIO, UART and ADC
IX. DMA Driver Development
X. I2C overview and driver development
XI. SPI overview and driver development

Unit III: Introduction to Industrial IoT **10 Lecture Hours**

I. Industrial Revolutions
II. Industry 4.0 revolutions
III. Cyber Physical system and next generation sensors
IV. IIoT Computing
V. Digital Twin Technology
VI. Digital Twin Conceptual Architecture
VII. AWS Digital Twin Maker
VIII. Software defined networking in IIoT
IX. Security in IIoT

Unit IV: Industrial Automation and IIoT Applications **14 Lecture Hours**

I. Introduction to PLC
II. Evolution, Types, advantages, limitations and applications of PLC
III. Introduction to SCADA
IV. Need and Applications
V. SCADA Architecture
VI. Human Machine Interfaces (HMI)
VII. Smart Home and Cities
VIII. Industry Overview and IIoT applications in
    a. Oil, Chemical, Pharmaceutical, Manufacturing Industry
    b. Logistic Industry
    c. Smart Agriculture
    d. Other Industries

Unit V: Medical IoT **12 Lecture Hours**

I. Healthcare Industry and Challenges
II. Telemedicine
III. Health Vital signs and symptoms
IV. Wearable sensors
V. Body Area Network
VI. Sensors for Medical applications

Total lecture Hours 60

Textbooks

1. Alasdair Gilchrist, "Industry 4.0: The Industrial Internet of Things", Apress, 2017.
2. Sabina Jeschke, Christian Brecher, Houbing Song, and Danda B. Rawat, "Industrial Internet of Things: Cybermanufacturing Systems", Springer, 2018.

Reference Books

1. Muhammad A. Mazidi, Shujen Chen, Eshragh Ghaemi, "STM32 Arm Programming for Embedded Systems", Microdigitaled, 2018.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSIS3120P	Industrial IoT and ARM based Embedded Programming Lab	0	0	2	1
Total Units to be Covered: 10	Total Contact Hours: 30				
Prerequisite(s):	Introduction to IoT Programming in C	Syllabus version: 1.0			

Course Objectives

1. Understand ARM Architecture and Embedded Programming.
2. Explore Industrial IoT (IIoT) and IIoT Applications.
3. Learn about Industrial Automation and SCADA Systems:
4. Understand Medical IoT and Healthcare Applications

Course Outcomes

On completion of this course, the students will be able to:

CO1. Students will be able to develop Proficiency in ARM Architecture and Embedded Programming.

CO2. Students will be able to program ARM controllers for sensor interfacing and data processing in embedded C.

CO3. Students will be able to apply Industrial IoT Principles and Technologies.

CO4. Students will be able to explore IIoT Applications in Different Industries.

CO5. Students will be able to understand Medical IoT and Healthcare Applications:

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	2	-	-	-	1	-	-	-	-	1	2
CO 2	-	-	-	-	2	-	-	-	1	-	-	-	-	3	2
CO 3	-	-	-	-	-	-	-	-	-	-	-	-	-		1
CO4	-	-	-	-	-	-	-	-	-	-	-	-	-	2	1
CO5	-	-	-	-	2	-	-	-	2	-	-	-	-	2	2

Average	-	-	-	-	1.2	-	-	-	.8	-	-	-	-	1.6	1.6

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

List of Experiments

Experiment 1: Introduction to ARM Programming IDE:

- Installing STM32Cube IDE and exploring the environment
- Exploring the documentation for ARM based micro controllers
- Overview and exploration of ARM cortex M series
- Exploring the ports and pins of the Nucleo boards.

Experiment 2: Introduction to ARM Programming:

- LED Blinking: Write a program to control an LED connected to an ARM-based board and make it blink at regular intervals.
- Interface a push-button with the ARM board and turn ON/OFF an LED.

Experiment 3: ARM GPIO Programming:

- Control GPIO outputs using the bit set/reset register (BSRR)
- Develop the GPIO input driver for reading value from the potentiometer
- Utilize timers to generate PWM signals on GPIO pins. Control the intensity of an LED or a servo motor using Pulse Width Modulation (PWM).

Experiment 4: UART Protocol:

- Develop the UART transmitter protocol to send any random number on the LCD display
- Develop the UART receiver protocol to read temperature sensor value and display it on the LCD.

Experiment 5: ADC Programming

- Develop the ADC single conversion cycle to read the data from an analog sensor.

Experiment 6: ARM SysTick timer

- Demonstrate and write the System tick timer driver using the Nucleo board.
- Use the SysTick Timer to blink the onboard LED on the Nucelo board.

Experiment 7: STM HAL Library

- Implement a GPIO (General Purpose Input/Output) experiment using STM HAL library on Nucleo board. Configure GPIO pins for input and output modes, toggle LED connected to an output pin, and read status from a push-button connected to an input pin.
- Set up a UART communication experiment using STM HAL library on Nucleo board. Configure UART module with desired baud rate, and establish bi-directional communication between Nucleo board and an external device, such as a PC, to exchange data.

Experiment 8: ARM Interrupt Timers

- Develop the GPIO interrupt timer for blinking an external LED on pushing external push button.

- Develop and demonstrate the UART transmitter timer.

Experiment 9: SPI Protocol Interfacing

Develop the SPI diver to Interface the ADXL accelerometer and read the sensor values.

Experiment 10: I2C Protocol Interfacing

Develop the I2C diver to Interface the ADXL accelerometer and read the sensor values.

Total Lab hours 30

Textbooks

1. Alasdair Gilchrist, "Industry 4.0: The Industrial Internet of Things", Apress, 2019.
2. Sabina Jeschke, Christian Brecher, Houbing Song, Danda B. Rawat, "Industrial Internet of Things: Cybermanufacturing Systems", Springer, 2017.

Reference Books

1. Muhammad A. Mazidi, Shujen Chen, and Eshragh Ghaemi, "STM32 Arm Programming for Embedded Systems", Microdigitaled, 2018.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSIS4011P	Single Board Computers and IoT Applications Development	4	0	0	4
Total Units to be Covered: 05		Total Contact Hours: 60			
Prerequisite(s):	Introduction to IoT Programming in C	Syllabus version: 1.0			

Course Objectives

1. Understand Single Board Computers and Embedded System.
2. Master Raspberry Pi and Linux-based Systems.
3. Develop Skills in Programming and Interfacing with Raspberry Pi.
4. Mastering cloud technologies for IoT applications.

Course Outcomes

On completion of this course, the students will be able to:

CO1. Students will be able to demonstrate Proficiency in Single Board Computers and Embedded Systems.

CO2. Students will apply Real-Time Operating Systems (RTOS) Principles.

CO3. Students will be able to utilize Raspberry Pi and Linux-Based Systems.

CO4. Students will be able to develop Skills in Programming and Interfacing with Raspberry Pi:

CO5. Students will be able to apply IoT Principles and Cloud Integration.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	2	-	-	-	1	-	-	-	-	1	2
CO 2	-	-	-	-	2	-	-	-	1	-	-	-	-	3	2
CO 3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1
CO4	-	-	-	-	-	-	-	-	-	-	-	-	-	2	1
CO5	-	-	-	-	2	-	-	-	2	-	-	-	-	2	2

Average	**-**	**-**	**-**	**-**	1.2	**-**	**-**	**-**	.8	**-**	**-**	**-**	**-**	1.6	1.6

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction **12 Lecture Hours**

I. System on Chip Overview
II. ARM Cortex series as SoC
III. Single Board Computers
IV. Components and configurations
V. SBC Types
VI. ASUS Tinkerboard
VII. BeagleBone Black
VIII. NVIDIA Jetson
IX. Latte Panda
X. Raspberry Pi

Unit II: RTOS Basics **12 Lecture Hours**

I. RTOS Need and challenges
II. RTOS Task Scheduling
III. Real time design issues
IV. Multi-threading models, threading issues, thread libraries, synchronization Mutex: creating, deleting, prioritizing mutex, mutex internals
V. Messages, Buffers, mailboxes, queues, semaphores
VI. Free RTOS Overview

Unit III: Raspberry Pi Introduction **12 Lecture Hours**

I. Linux- Introduction, Architecture, File System
II. Linux commands, sudo, apt

III. Raspbian O.S.- Introduction
IV. Rasperry Pi Pin description
V. RPi configurations
VI. On board components on RPi
VII. SoC used for RPi
VIII. Pi Pico Introduction
IX. Pico vs Arduino Nano
X. Programming in Pico

Unit IV: Programming and Interfacing **12 Lecture Hours**

I. Sensors Interfacing- Temperature and Humidity Sensor (DHT11), Motion Sensor (PIR), Obstacle detection using Ultrasonic sensor
II. Communicating using RPi- GSM interfacing, Accessing on-board Wi-Fi
III. Connecting Database with RPi
IV. Using Wiring Pi for GPIO Programming
V. Various sensors, modules interfacing with SPI, I2C and UART
VI. Interfacing Rpi using C
VII. SSH command and working
VIII. RPi as SSH
IX. RPi as Web server
X. GPIO Control over WebBrowser
XI. Node-RED, MQTT Protocol
XII. Camera interfacing

Unit V: Cloud Technologies for IoT **12 Lecture Hours**

I. Overview of GCP, AWS, Azure, ThingSpeak
II. Overview of Azure cloud computing platform and services
III. Key Azure services, such as virtual machines, storage, networking, and databases
IV. Introduction to Azure IoT Hub
V. Understanding IoT device connectivity and messaging with Azure IoT Hub
VI. Provisioning IoT devices with Azure IoT Hub and managing device identities
VII. Device authentication and authorization with Azure IoT Hub

VIII. Connecting Azure IoT Hub with Azure Stream Analytics for real-time data processing

IX. Connecting RPi with Azure

X. Raspberry Pi data logging and visualization with Azure

Total lecture Hours 60

Textbooks

1. Simon Monk, "Programming the Raspberry Pi", 2nd Edition, McGraw Hill TAB, 2015.
2. Ritesh Modi, "Azure for Architects" by Packt Publishing Limited, 2017.

Reference Books

1. Doug Abbott, "Linux for Embedded and Real-Time Applications", Newnes, 2003.
2. Richard Barry, "Mastering the FreeRTOS™ Real Time Kernel", Real Time Engineers Ltd. Available online at: https://freertos.org/Documentation/161204_Mastering_the_FreeRTOS_Real_Time_Kernel-A_Hands-On_Tutorial_Guide.pdf

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	**IA**	**MID SEM**	**End Sem**	**Total**
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	**Weightage in calculation of Internal Assessment (100 marks)**
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSIS4111P	Single Board Computers and IoT Applications Development Lab	0	0	2	1
Total Units to be Covered: 10	Total Contact Hours: 30				
Prerequisite(s):	Introduction to IoT Programming in C	Syllabus version: 1.0			

Course Objectives

1. Understand Single Board Computers and Embedded System.
2. Master Raspberry Pi and Linux-based Systems.
3. Develop Skills in Programming and Interfacing with Raspberry Pi.
4. Mastering cloud technologies for IoT applications.

Course Outcomes

On completion of this course, the students will be able to:

CO1. Students will be able to demonstrate Proficiency in Single Board Computers and Embedded Systems.

CO2. Students will apply Real-Time Operating Systems (RTOS) Principles.

CO3. Students will be able to utilize Raspberry Pi and Linux-Based Systems.

CO4. Students will be able to develop Skills in Programming and Interfacing with Raspberry Pi:

CO5. Students will be able to apply IoT Principles and Cloud Integration.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	2	-	-	-	-	-	-	-	2	1	2
CO 2	-	-	-	-	2	-	-	-	-	-	-	-	2	1	2
CO 3	-	-	-	-	2	-	-	-	-	-	-	-	2	1	2

CO4	-	-	-	-	2	-	-	-	-	-	-	-	2	1	2
CO5	-	-	-	-	2	-	-	-	-	-	-	-	2	2	2
Average	-	-	-	-	2	-	-	-	-	-	-	-	2	1	2

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Introduction to Raspberry Pi Pico:

- To familiarize with the Raspberry Pi Pico GPIO and its programming by blinking an LED.
- Interface an analog sensor (e.g., potentiometer) with the Raspberry Pi Pico, and read analog values using the built-in ADC.
- Connect a DHT11 or DHT22 sensor to the Raspberry Pi Pico and read temperature and humidity data using GPIO pins and the sensor library.

Experiment 2: Real-Time Operating Systems (RTOS) on Raspberry Pi Pico**:**

- Install and configure FreeRTOS on Raspberry Pi to understand its basic functionalities and features.
- Implement task scheduling and prioritize and run multiple tasks for real-time execution using FreeRTOS.

Experiment 3: Raspberry Pi Introduction and Linux-Based Systems**:**

- Explore the Linux operating system and its architecture, including the file system and shell commands.

- Install, configure Raspbian OS on Raspberry Pi and familiarize with its boot process and essential services.
- Configure a buzzer and play various frequency tones.

Experiment 4: Sensor Interfacing with Raspberry Pi

- Interface temperature and humidity sensor (DHT11) with Raspberry Pi to collect environmental data.
- Set up a motion sensor (PIR) to detect motion and play buzzer using Raspberry Pi..
- Develop obstacle detection using an ultrasonic sensor and process the data for obstacle avoidance and play buzzer with various patterns as per obstacle distance.

Experiment 5: Communication Methods on Raspberry Pi

- Utilize GSM interfacing on Raspberry Pi to send and receive SMS messages or data.
- Establish Wi-Fi connectivity on Raspberry Pi to communicate with other devices or the internet.
- Using SPI protocol to connect to an appropriate sensor and display the readings.
- Using I2C protocol to connect to accelerometer sensor and display the readings.

Experiment 6: GPIO Programming and Control

- Develop GPIO programming using Wiring Pi library to control LEDs, buttons, and other peripherals.

- Interface a servo motor with Raspberry Pi and control its movements through GPIO pins.
- Implement PWM control on Raspberry Pi to adjust the brightness of an LED or control motor speed.

Experiment 7: Node-RED and MQTT Protocol Integration

- Set up Node-RED on Raspberry Pi and create flows to process and visualize sensor data.
- Implement MQTT communication protocol to enable message exchange between Raspberry Pi and other IoT devices.
- Integrate Node-RED and MQTT to create a simple IoT application for real-time data streaming.

Experiment 8: Raspberry Pi Camera Module

- interface the Raspberry Pi Camera Module with the Raspberry Pi, install the necessary libraries, and capture images using Python code.
- To record videos using the Raspberry Pi Camera Module, configure video settings such as resolution and frame rate, and save the recorded video to the storage.

Experiment 9: Microsoft Azure IoT/ AWS IoT/ GCP Hub Integration

- Connect Raspberry Pi with Microsoft Azure IoT Hub for secure device connectivity.
- Provision IoT devices on Azure and manage their identities for secure communication.

- Implement device authentication and authorization to ensure data security on the Azure platform.

Experiment 10: Raspberry Pi-Based Data Logging and Visualization with Azure or AWS or GCP:

- Implement data logging on Raspberry Pi to collect and store sensor data over time.
- Use Azure Stream Analytics to process and analyze real-time data from Raspberry Pi.
- Visualize the analyzed data using Azure tools like Power BI for effective data representation.

Total Lab hours 30

Textbooks

1. Simon Monk, "Programming the Raspberry Pi", 2nd Edition, by McGraw Hill TAB, 2015.
2. Ritesh Modi, "Azure for Architects" by Packt Publishing Limited, 2017.

Reference Books

1. Doug Abbott, "Linux for Embedded and Real-Time Applications", Newnes, 2003.
2. Richard Barry, "Mastering the FreeRTOS™ Real Time Kernel", Real Time Engineers Ltd. Available online at: https://freertos.org/Documentation/161204_Mastering_the_FreeRTOS_Real_Time_Kernel-A_Hands-On_Tutorial_Guide.pdf

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSIS4012P	**Data Analytics for IoT**	3	0	0	3
Total Units to be Covered: 05	**Total Contact Hours: 45**				
Prerequisite(s):	Python Programming Introduction to IoT Programming in C	**Syllabus version: 1.0**			

Course Objectives

1. Understand the Fundamentals of IoT Data Analytics:
2. Master Time Series Data Analysis Techniques for IoT:
3. Develop Machine Learning Competence for IoT Data:
4. Apply TensorFlow for IoT Data Analytics:

Course Outcomes

On completion of this course, the students will be able to:

CO1. Students will be able to apply Data Analytics Techniques to IoT Data.

CO2. Students will be able to analyze Time Series Data for IoT Applications.

CO3. Students will be able to implement Machine Learning Algorithms for IoT Data.

CO4. Students will be able to utilize TensorFlow for IoT Data Analytics.

CO5. Students will be able to apply IoT Data Analytics in Real-World Use Cases.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	2	-	-	-	1	-	-	-	-	1	2
CO 2	-	-	-	-	2	-	-	-	1	-	-	-	-	3	2
CO 3	-	-	-	-		-	-	-	-	-	-	-	-		1
CO4	-	-	-	-		-	-	-	-	-	-	-	-	2	1
CO5	-	-	-	-	2	-	-	-	2	-	-	-	-	2	2
Average	-	-	-	-	2	-	-	-	1	-	-	-	-	2	1.6

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to IoT Data Analytics **8 Lecture Hours**

I. Understanding the role and significance of data analytics in IoT applications
II. Types of Data Analysis
   a. Descriptive
   b. Diagnostics
   c. Predictive
   d. Prescriptive
III. Structured vs Unstructured Data
IV. Use cases for IoT Data Analytics
   a. Inventory management
   b. Predictive Maintenance
   c. Healthcare

Unit II: Time Series Data **12 Lecture Hours**

I. Characteristics and properties of time series data
II. Types of time series data, such as univariate and multivariate
III. Applications and challenges
IV. Anomaly and Outlier detection in time series
V. Autocorrelation, Seasonality, Stationarity
VI. Forecasting methods ARIMA, SARIMA, Moving Average and exponential smoothing
VII. Seasonality and trend analysis in time series data
VIII. Decomposition methods for separating trend, seasonality
IX. Time series clustering and segmentation techniques
X. Spectral analysis and wavelet transforms
XI. EEG Data analysis example

Unit III: Machine Learning Basics **10 Lecture Hours**

- **I.** Supervised vs Unsupervised learning vs Reinforcement Learning
- II. Linear Regression
- III. Curve Fitting and Polynomial Regression
- IV. Logistic Regression
- V. Classification with SVM, Decision Tree, Neural Network
- VI. Ensemble Learning
- VII. Dimensionality Reduction
- VIII. Deep Learning Overview
- IX. Convolution Technique
- X. Activation functions
- XI. Convolution Neural Network
- XII. Other models

Unit IV: TensorFlow for IoT **10 Lecture Hours**

- I. Tensorflow
- II. Tensor in Tensorflow
- III. Tensorflow constants, variables
- IV. Tensorflow keras model
- V. Compile a tensorflow model
- VI. Training and evaluating
- **VII.** TensorFlow Lite
- **VIII.** Features of TensorFlow Lite
- **IX.** TensorFlow Lite on Arduino
- **X.** TensorFlow Lite on Raspberry Pi

Unit V: Use Case Discussion **5 Lecture Hours**

- I. Problem formulation
- II. Detailed solution with IoT
- III. Data collection and cleaning
- IV. Model curation and parametrization

V. Testing and Validation

Total lecture Hours 45

Textbooks

1. Suhasini Subba Rao, "A course in Time Series Analysis", 2022. Available online at: https://web.stat.tamu.edu/~suhasini/teaching673/time_series.pdf.

2. Aurélien Géron, "Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow", 3rd Edition, Shroff/O'Reilly, 2022

Reference Books

1. Karthikeyan NG, "Machine Learning Projects for Mobile Applications", Packt Publishing Limited, 2018.
2. Robert Shumway, and David Stoffer, "Time Series: A Data Analysis Approach Using R", Chapman and Hall/CRC, 2019.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSIS4012P	Data Analytics for IoT Lab	0	0	2	1
Total Units to be Covered: 09	Total Contact Hours: 30				
Prerequisite(s):		Syllabus version: 1.0			

Course Objectives

1. Understand the Fundamentals of IoT Data Analytics:
2. Master Time Series Data Analysis Techniques for IoT:
3. Develop Machine Learning Competence for IoT Data:
4. Apply TensorFlow for IoT Data Analytics:

Course Outcomes

On completion of this course, the students will be able to:

CO1. Students will be able to apply Data Analytics Techniques to IoT Data.
CO2. Students will be able to analyze Time Series Data for IoT Applications.

CO3. Students will be able to implement Machine Learning Algorithms for IoT Data.

CO4. Students will be able to utilize TensorFlow for IoT Data Analytics.

CO5. Students will be able to apply IoT Data Analytics in Real-World Use Cases.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	2	-	-	-	-	-	-	-	2	1	-
CO 2	-	-	-	-	2	-	-	-	-	-	-	-	2	1	-
CO 3	-	-	-	-	2	-	-	-	-	-	-	-	2	1	-
CO4	-	-	-	-	2	-	-	-	-	-	-	-	2	1	-
CO5	-	-	-	-	2	-	-	-	-	-	-	-	2	2	-
Average	-	-	-	-	2	-	-	-	-	-	-	-	2	1.2	-

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Lab. Exercise 1	Introduction to Data Analytics for IoT
Lab. Exercise 2	Time series programming lab 1 a) Exploring time series data and its attributes b) Working out forecasting methods.
Lab. Exercise 3	Time series programming lab 1 a) Experimenting trend analysis b) Clustering and segmentation techniques
Lab. Exercise 4	Programming exercise 1: Exploring machine learning basics.
Lab. Exercise 5	Programming exercise 2: Exploring machine learning basics.
Lab. Exercise 6	Implementation of CNN on an available dataset.
Lab. Exercise 7	Exploring tensor flow library for IoT data analytics.
Lab. Exercise 8	Using tensor flow for training and evaluating a model
Lab. Exercise 9	Implementation of Tensorflow Lite on Arduino

Total Lab hours 30

Textbooks

1. "A course in Time Series Analysis" by Suhasini Subba Rao
2. "Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow" by Aurélien Géron

Reference Books

1. " Machine Learning Projects for Mobile Applications" by Karthikeyan NG

2. “Time Series: A Data Analysis Approach Using R” by Robert Shumway and David Stoffer

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

9 . Graphics & Gaming Track

Course Code	Course name		L	T	P	C
CSGG2011P	Introduction to Graphics and Animation		4	0	0	4
Total Units to be Covered: 04		Total Contact Hours: 60				
Prerequisite(s):	Basic knowledge about Computers.		Syllabus version: 1.0			

Course Objectives

Students will be able to learn:

1. Students will be able to learn design thinking methodologies and principles, applying them to problem-solving and user-centered design.
2. Students will be able to learn proficiency in industry-standard design software tools, effectively utilizing them in graphics, 3D modeling, animation, UI/UX design, and interactive design.
3. Students will be able to learn the application of design principles, such as typography, color theory, and gestalt principles, to create visually appealing and effective designs.
4. Students will be able to learn skills in 3D modeling, animation techniques, and interactive design, enabling them to create captivating and engaging interactive experiences.

Course Outcomes

On completion of this course, the students will be able to:

CO1: Demonstrate a comprehensive understanding of design thinking methodologies and principles and apply them effectively to solve complex design problems and create user-centered interactive experiences.

CO2: Apply user experience (UX) design principles and conduct user research to design intuitive and user-friendly interfaces, and effectively communicate user flows and interactions through wireframes and prototypes.

CO3: Apply design principles, such as typography, color theory, and gestalt principles, to create visually appealing and cohesive designs that effectively communicate the intended message and engage users.

CO4: Create high-quality 3D models, animations, and interactive elements, demonstrating proficiency in using relevant software and techniques to bring designs to life.

CO5: Develop proficiency in industry-standard design software tools, utilizing them proficiently in graphics, 3D modeling, animation, UI/UX design, and interactive design projects.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2
CO 2	-	-	-	-	2	-	-	-	2	-	-	-	-	-	2
CO 3	-	-	-	-		-	-	-	-	-	-	-	-	-	2
CO4	-	-	-	-	2	-	-	-	2	-	-	-	-	-	2
CO5	-	-	-	-	2	-	-	-	2	-	-	-	-	-	3
Average	-	-	-	-	1.2	-	-	-	1.2	-	-	-	-	-	2.2

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **“_” means there is no correlation**

Syllabus

Unit I: **Design Fundamentals** **15 Lecture Hours**

Theory and Design Thinking, Introduction to Graphics and Visual Communication, Design Principles and Composition, Typography and Color Theory, Gestalt Principles and Advanced Design Techniques.

Unit II: UI/UX Design **15 Lecture Hours**

User Experience (UX) Design Principles and Methodologies, User Interface (UI) Design Best Practices

Designing Intuitive and User-Friendly Interfaces, Conducting User Research and Usability Testing, Creating Interactive Elements and User Interactions, Designing User Flows and Wireframes, Prototyping Interactive Designs using FIGMA, Photoshop.

Unit III: 3D Modelling and Animation. **15 Lecture Hours**

Blender/Maya Interface and Navigation, Mesh Modeling Techniques, UV Mapping and Texturing, Rigging, Lightning and Rendering. Principles of Animation by Disney, Techniques for 2d Animation and 3D Animation.

Unit IV: Capstone Projects **15 Lecture Hours**

Project Planning and Concept Development, Pre-production: research, concept art, storyboards, script

Production: asset creation, character rigging, animation, post-production: refining animation, adding sound effects and music, Presentation and Documentation: final presentation, project documentation, portfolio creation.

A capstone project depicting the above skillsets in the following project categories. Short Film Production, Character Animation Showcase, Motion Graphics Advertisement, Game Animation, Visual Effect Showcase.

Total lecture Hours 60

Textbooks

1. Jennifer Preece, Yvonne Rogers, and Helen Sharp “Interaction Design: Beyond Human-Computer Interaction", 5th Edition, Wiley, 2019.
2. Todd Palamar, "Mastering Autodesk Maya 2022: A Comprehensive Guide to 3D Animation and Modeling".

Reference Books

1. Ben Simonds, “Blender Master Class: A Hands-On Guide to Modeling, Sculpting, Materials, and Rendering", No Starch Press, 2013.
2. Don Norman, “The Design of Everyday things”, Hachette Audio, 2018.
3. Richard E Williams, “The Animator’s Survivor Kit”, Faber & Faber, 2009.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSGG2110P	Introduction to Interactive Design and 3D Animation Lab		0	0	2	1
Total Units to be Covered: 10	Total Contact Hours: 30					
Prerequisite(s):	Elementary knowledge about computers.	Syllabus version: 1.0				

Course Objectives

Students will be able to learn:

1. Students will be able to learn design thinking methodologies and principles, applying them to problem-solving and user-centered design.
2. Students will be able to learn proficiency in industry-standard design software tools, effectively utilizing them in graphics, 3D modeling, animation, UI/UX design, and interactive design.
3. Students will be able to learn the application of design principles, such as typography, color theory, and gestalt principles, to create visually appealing and effective designs.
4. Students will be able to learn skills in 3D modeling, animation techniques, and interactive design, enabling them to create captivating and engaging interactive experiences.

Course Outcomes

On completion of this course, the students will be able to:

CO1: Demonstrate a comprehensive understanding of design thinking methodologies and principles and apply them effectively to solve complex design problems and create user-centered interactive experiences.

CO2: Apply user experience (UX) design principles and conduct user research to design intuitive and user-friendly interfaces, and effectively communicate user flows and interactions through wireframes and prototypes.

CO3: Apply design principles, such as typography, color theory, and gestalt principles, to create visually appealing and cohesive designs that effectively communicate the intended message and engage users.

CO4: Create high-quality 3D models, animations, and interactive elements, demonstrating proficiency in using relevant software and techniques to bring designs to life.

CO5: Develop proficiency in industry-standard design software tools, utilizing them proficiently in graphics, 3D modeling, animation, UI/UX design, and interactive design projects.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	2
CO 2	-	-	-	-	2	-	-	-	2	-	-	-	-	-	2
CO 3	-	-	-	-		-	-	-	-	-	-	-	-	-	2
CO4	-	-	-	-	2	-	-	-	2	-	-	-	-	-	2
CO5	-	-	-	-	2	-	-	-	2	-	-	-	-	-	3
Average	-	-	-	-	1.2	-	-	-	1.2	-	-	-	-	-	2.2

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Introduction to Graphic Design Principles and Composition

1. Create a mood board using visual elements to evoke specific moods or styles.
2. Design a simple composition applying graphic design principles.

Experiment 2: Typography and Color Theory Exploration

1. Design a typographic poster using appropriate typefaces, font sizes, and hierarchy.
2. Apply color theory principles to create a visually appealing color scheme.

Experiment 3: Gestalt Principles and Advanced Design Techniques

1. Create a series of design compositions demonstrating the application of Gestalt principles.
2. Experiment with advanced design techniques such as visual illusions, negative space, or unconventional layouts.

Experiment 4: User Experience (UX) Design Principles, Usability Studies

1. Conduct a small-scale user research study, including interviews or surveys.
2. Analyze the collected data and create user personas.
3. Present findings and recommendations for improving the user experience of a selected application.

Experiment 5: Designing User Flows and Wireframes

1. Design user flows illustrating the step-by-step process and navigation of a selected application or website.
2. Create wireframes for key screens or pages considering information hierarchy and user interactions.
3. Present wireframes, explaining the design decisions and justifications.

Experiment 6: Prototyping Interactive Designs using Figma

1. Create interactive prototypes using Figma, incorporating user interactions, transitions, and feedback elements.
2. Conduct usability testing on the prototypes and gather feedback for refinement.

Experiment 7: Introduction to 3D Modelling

1. Use 3D modeling software to create simple 3D models of objects.
2. Apply learned techniques to add realism to the models through texture mapping.

Experiment 8: Advance 3D Modelling

1. Apply UV mapping techniques to unwrap 3D models.
2. Add textures to the models, considering lighting and shading principles.
3. Render the models to create visually appealing 3D scenes.

Experiment 9: Animation

1. Create a short 2D animation applying animation principles and techniques.
2. Explore basic 3D animation techniques, including keyframing, rigging, and character animation.

Experiment 10: Capstone Project

1. Work on a comprehensive design project integrating graphic design, UX design, and 3D modeling or animation elements.
2. Create a final deliverable, such as a presentation or prototype, showcasing the project's design process and outcomes.

Total Lab hours 30

Textbooks

1. Jennifer Preece, Yvonne Rogers, and Helen Sharp “Interaction Design: Beyond Human-Computer Interaction", 5th Edition, Wiley, 2019.
2. Todd Palamar, "Mastering Autodesk Maya 2022: A Comprehensive Guide to 3D Animation and Modeling".

Reference Books

1. Ben Simonds, “Blender Master Class: A Hands-On Guide to Modeling, Sculpting, Materials, and Rendering", No Starch Press, 2013.
2. Don Norman, “The Design of Everyday things”, 2nd Edition, Basic Books, 2013.
3. Richard E Williams, “The Animator’s Survivor Kit”, Faber & Faber, 2009.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name		L	T	P	C
CSGG3019P	Game Programming		4	0	0	4
Total Units to be Covered: 05	Total Contact Hours: 60					
Prerequisite(s):	Elementary knowledge of Computer, OOPs, DSA, 3d Mathematics	Syllabus version: 1.0				

Course Objectives

Students will be able to:

1. Understand the game design process, including game genres, storytelling, and prototyping.
2. Develop programming skills for game development, covering variables, control structures, OOP, scripting, and debugging.
3. Gain proficiency in Unity game engine for scene management, asset manipulation, physics, AI, UI, and optimization.
4. Explore advanced game development topics, such as procedural content generation, advanced AI, audio, and visual effects.
5. Apply knowledge through a capstone project, encompassing project planning, development, iteration, and presentation.

Course Outcomes

Students will be able to:

CO1 Analyze and evaluate game design principles and concepts, demonstrating an understanding of the game development process, game genres, and storytelling techniques.

CO2 Apply programming fundamentals for game development, utilizing variables, control structures, object-oriented programming principles, scripting languages, and debugging techniques.

CO3 Demonstrate proficiency in using the Unity game engine, including project setup, scene management, asset manipulation, physics simulation, user input and controls, AI implementation, user interface design, sound design, and visual effects.

CO4 Explore advanced game development topics, such as procedural content generation, advanced physics simulation, advanced AI techniques, advanced user interface design, advanced audio design, and advanced visual effects and shaders.

CO5 Plan, manage, and develop a game project using the knowledge and skills acquired throughout the course, iterating on the project based on feedback, and presenting the final game project to showcase competence in game development.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	2	-	-	-	-	-	-	-	-	-	2	-	-
CO 2	1	-	2	-	-	-	-	-	-	-	-	-	2	-	-
CO 3	1	-	2	-	2	-	-	-	2	-	-	-	-	-	3
CO4	1	-	2	-	2	-	-	-	2	-	-	-	-	-	3
CO5	-	-	-	-	2	-	-	-	-	-	-	-	2	2	3
Average	1	-	1.6	-	1.2	-	-	-	0.8	-	-	-	1.2	0.4	1.8

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to Game Design and Development **12 Lecture Hours**

Overview of game design and development process, Game Development Life Cycle, Introduction to game engines and their importance, Understanding the roles in game development teams, Fundamentals of game design principles and concepts, Introduction to game genres and platforms, Game ideation and concept development, Game storytelling and narrative design, Game prototyping and playtesting, Game Design Document

Unit II: Programming Fundamentals for Game Development 12 Lecture Hours

Introduction to programming languages used in game development, Variables, data types, and operators, Control structures (conditionals and loops), Functions and procedures, Object-Oriented Programming (OOP) principles for game development, Event-driven programming in game development, Memory management in game programming, Debugging and error handling in game development, Scripting languages for game development, Scripting for game mechanics and interactions, Programming patterns in game development(Singleton Pattern, Factory Pattern etc.)

Unit III: Game Development with Unity 12 Lecture Hours

Introduction to Unity game engine and its features, Unity interface and project setup, Working with scenes and game objects, Introduction to components and scripts in Unity, Scripting in Unity using C#, Importing and manipulating game assets (e.g., models, textures, audio), Physics simulation and collisions in Unity, User input and controls in Unity, AI and pathfinding in game development with Unity, User interface design and implementation in Unity, Sound design and implementation in Unity, Visual effects and shaders in Unity, Game optimization and performance in Unity, Networking and multiplayer implementation in Unity, Ray-casting, Tilemaps, Animator.

Unit IV: Advanced Game Development Topics 12 Lecture Hours

Procedural content generation in game development, Advanced physics simulation in game development, Advanced AI and behavior trees in game development, Advanced user interface design and implementation, Advanced audio design and implementation, Advanced visual effects and shaders, Game analytics and player behavior analysis, Cross-platform game development, Emerging technologies in game development

Unit V: Capstone Project: Game Development 12 Lecture Hours

Game project planning and management, Game project development and iteration, Game project polishing and finalization, Game project presentations and feedback.

Total lecture Hours 60

Textbooks

1. Harrison Ferrone, “Unity in Action: Multiplatform Game Development in C#”, 2nd Edition, Manning Publications, 2018.
2. Robert Nystrom, “Game Programming Patterns”, The Pragmatic Bookshelf, 2014.
3. Jeremy Gibson Bond, “Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#”, 3rd Edition, Addison-Wesley Professional, 2022.
4. Joe Hocking, “Unity in Action: Multiplatform Game Development in C#”, 2nd Edition, Manning Publications, 2018.

Reference Books

1. Alan Thorn, “Mastering Unity 2D Game Development”, Packt Publishing, 2014.
2. Harrison Ferrone, “Learning C# by Developing Games with Unity”, 7th Edition, Packt Publishing, 2022.
3. Simon Jackson, “Mastering Unity 2D Game Development”, Packt Publishing, 2014.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSGG3113P	Game Programming Lab		0	0	2	1
Total Units to be Covered: 08		Total Contact Hours: 30				
Prerequisite(s):	Programming Basics		Syllabus version: 1.0			

Course Objectives

1. Understand the game design process, including game genres, storytelling, and prototyping.
2. Develop programming skills for game development, covering variables, control structures, OOP, scripting, and debugging.
3. Gain proficiency in Unity game engine for scene management, asset manipulation, physics, AI, UI, and optimization.
4. Explore advanced game development topics, such as procedural content generation, advanced AI, audio, and visual effects.
5. Apply knowledge through a capstone project, encompassing project planning, development, iteration, and presentation.

Course Outcomes

Students will be able to:

CO1 Analyze and evaluate game design principles and concepts, demonstrating an understanding of the game development process, game genres, and storytelling techniques.

CO2 Apply programming fundamentals for game development, utilizing variables, control structures, object-oriented programming principles, scripting languages, and debugging techniques.

CO3 Demonstrate proficiency in using the Unity game engine, including project setup, scene management, asset manipulation, physics simulation, user input and controls, AI implementation, user interface design, sound design, and visual effects.

CO4 Explore advanced game development topics, such as procedural content generation, advanced physics simulation, advanced AI techniques, advanced user interface design, advanced audio design, and advanced visual effects and shaders.

CO5 Plan, manage, and develop a game project using the knowledge and skills acquired throughout the course, iterating on the project based on feedback, and presenting the final game project to showcase competence in game development.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	2	-	-	-	-	-	-	-	-	-	2	-	-
CO 2	1	-	2	-	-	-	-	-	-	-	-	-	2	-	-
CO 3	1	-	2		2	-	-	-	2	-	-	-	-	-	3
CO 4	1	-	2		2	-	-	-	2	-	-	-	-	-	3
CO 5	-	-	-	-	2	-	-	-	-	-	-	-	-	2	3
Average	.8	-	1.6		1.2	-	-	-	.8	-	-	-	.8	.4	1.8

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment1 **Generating a Graphical Design Document (GDD):**

- Define the game concept and objectives.
- Identify the target audience and their preferences.
- Create a visual representation of game characters, environments, and user interfaces.
- Outline the gameplay mechanics and rules.

- Specify the audio and visual elements.
- Document the technical requirements and constraints.

Experiment2 Exploring the Fundamentals of C#: Variables, Control Statements, and Loops:

- Understand the concept of variables and their types.
- Learn how to declare and initialize variables in C#.
- Explore control statements such as if-else, switch, and loops like for, while, and do-while.
- Practice using conditional statements and loops in simple coding exercises.
- Understand the importance of variable scope and its impact on program execution.

Experiment3 Mastering Object-Oriented Programming using C#:

- Understand the principles of object-oriented programming (OOP).
- Learn about classes, objects, and their relationships.
- Explore encapsulation, inheritance, and polymorphism concepts.
- Practice creating classes, defining properties and methods.
- Implement inheritance and understand its benefits in code reuse and organization.

Experiment4 Setting up Unity: Installation, Utilizing Unity Hub, and Acquiring Licenses:

- Download and install the Unity game engine.
- Set up Unity Hub to manage projects and versions.
- Create a Unity ID and acquire necessary licenses (if applicable).
- Familiarize yourself with the Unity Editor's interface and features.
- Configure Unity preferences and project settings.

Experiment5 Immersing in Unity3D: Creating a Two-Player Coin Collection Game:

- Create a new Unity project and set up a 3D environment.
- Design player characters and coin model.
- Implement player movement and collision detection.
- Program the coin collection mechanics and keep track of player scores.
- Set win/lose conditions and implement game over functionality.

Experiment6 Enhancing User Experience: Working with Unity3D UI Elements and Implementing Game Start and Closure Functions:

- Add UI elements like buttons, text boxes, and menus to the game interface.
- Implement a start menu to allow players to begin the game.
- Create a game over screen with options for replaying or exiting the game.
- Add sound effects and visual cues to enhance the user experience.
- Implement functionality to pause or exit the game during gameplay.

Experiment7 Advanced Capabilities of the Unity Game Engine: Creating a Mouse-Controlled Star Destroyer Game:

- Design a 2D or 3D environment for the game.
- Implement a star destroyer character that follows the mouse cursor.
- Define the game objective, such as collecting power-ups or avoiding obstacles.
- Implement enemy entities or obstacles that interact with the star destroyer.
- Program game mechanics and win/lose conditions.

Experiment8 Multi-Platform Development: Designing a Candy Crush-style Game for Various Operating Systems:

- Plan the game mechanics and rules similar to Candy Crush.
- Design the game board with colorful candies and match patterns.
- Implement swipe or click-based controls for selecting and swapping candies.
- Program matching algorithms to detect and remove matched candies.
- Test and optimize the game for different operating systems, such as Windows, macOS, iOS, and Android.

Total Lab hours 30

Textbooks

1. Harrison Ferrone, “Unity in Action: Multiplatform Game Development in C#”, 2nd Edition, Manning Publications, 2018.
2. Robert Nystrom, “Game Programming Patterns,” Lightning Source Inc, 2014.

3. Jeremy Gibson Bond, “Introduction to Game Design, Prototyping, and Development: From Concept to Playable Game with Unity and C#”, 3rd Edition, Addison-Wesley Professional, 2022.
4. Joe Hocking, “Unity in Action: Multiplatform Game Development in C#”, 2nd Edition, Manning Publications, 2018.

Reference Books

1. Alan Thorn, “Mastering Unity 2D Game Development”, Packt Publishing.
2. Harrison Ferrone, “Learning C# by Developing Games with Unity”, 6th Edition, Packt Publishing, 2021.
3. Simon Jackson, “Mastering Unity 2D Game Development”, Packt Publishing, 2014.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSGG3020P	Computer Graphics	4	0	0	4
Total Units to be Covered: 05	Total Contact Hours: 60				
Prerequisite(s):	Basic knowledge about Matrices and 3D geometry. Elementary knowledge of C/C++	Syllabus version: 1.0			

Course Objectives

Students will be able to:

1. Students will be able to create 2D and 3D objects.
2. Students will be able to apply various transformations on the 2D and 3D objects.
3. Students will apply hidden surface removal techniques along with various shading algorithms.
4. Students will create 3D graphics with realistic effects.

Course Outcomes

On completion of this course, the students will be able to:

CO1. Classify various graphics hardware and software devices.

CO2. Use primitive operations to create 2D and 3D objects and perform various operations thereon.

CO3. Perform complex 2D and 3D transformations on objects.

CO4. Implement various hidden surface removal techniques.

CO5. Create 3D realistic imagery by applying shading and coloring techniques on objects.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	1	-	-	2	-	-	-	-	-	-	-	1	-	-
CO 2	2	1	1	-	-	-	-	-	-	-	-	-	3	-	-
CO 3	2	1	1	-	-	-	-	-	-	-	-	-	2	-	1
CO4	2	1	1	-	-	-	-	-	-	-	-	-	2	-	1

CO5	2	2	2	-	2	-	-	-	2	-	-	-	2	-	2
Average	1.8	1.2	1.25	-	2	-	-	-	2	-	-	-	2	-	1.33

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: **12 Lecture Hours**

Introduction to Computer Graphics and OpenGL: Overview of Computer Graphics, Computer Graphics Application and Software, Description of some graphics devices, Introduction to pixel. Why OpenGL, features in OpenGL, OpenGL operations, Abstractions in OpenGL – GL, GLU & GLUT, 3D viewing pipeline, viewing matrix specifications, a few examples and demos of OpenGL programs.

Unit II: **12 Lecture Hours**

Scan conversion – lines, circles and Ellipses; Filling polygons and clipping algorithms Scan Converting Lines, Mid-point criteria, Aliasing and Antialiasing, Problems of Aliasing, end-point ordering and clipping lines, Scan Converting Circles, Scan Converting Ellipses, Filling Polygons Clipping Lines algorithms– Cyrus Beck, Cohen-Sutherland and Liang-Barsky, Clipping Polygons, problem with multiple components.

Unit III: **12 Lecture Hours**

2-D and 3-D Transformations: Transformations and Matrices, Transformation Conventions, 2D Transformations, Homogeneous Coordinates and Matrix Representation of 2D Transformations, Translations and Homogeneous Coordinates, Rotation, Reflection, Scaling, Combined Transformation, Transformation of Points, Transformation of The UNIT Square, Rotation About an Arbitrary Point, Reflection through an Arbitrary Line, A Geometric Interpretation of Homogeneous Coordinates, The Window-to-Viewport Transformations. Three-Dimensional Scaling, Three-Dimensional Shearing, Three-Dimensional Rotation, Three Dimensional Reflection, Three-Dimensional Translation, Multiple Transformation, Rotation about an Arbitrary Axis in Space, Reflection through an Arbitrary Plane, Matrix Representation of 3D Transformations, Composition of 3D Transformations.

Unit IV: **12 Lecture Hours**

Rendering : Visible-Surface Determination, Techniques for efficient Visible-Surface Algorithms, Categories of algorithms, Back face removal, the Z-Buffer Algorithm, Scan-line method, Painter's algorithms (depth sorting), Octrees , BSP trees, Visible-Surface Ray Tracing, comparison of the methods.

Illumination and Shading Models for Polygons, Reflectance properties of surfaces, Ambient, Specular and Diffuse reflections, Phong's model, Gouraud shading, some examples.

Unit V: **12 Lecture Hours**

Plane Curves and Surfaces

Curve Representation, Nonparametric Curves, Cubic Splines, Bezier Curves, Bspline Curves, B-spline Curve Fit, Knot Vectors, NURBS, Parametric Cubic Curves, Quadric Surfaces. Bezier Surfaces, Fractals.

Total lecture Hours 60

Textbooks

1. Donald D. Hearn and M. Pauline Baker, "Computer Graphics using OpenGL", Pearson Education India, 2014.

Reference Books

1. David F. Rogers and J. Alan Adams, "Mathematical Elements for Computer Graphics", 2nd Edition, McGraw- Hill, Inc, 2017.
2. James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes "Computer Graphics: Principles and Practice", Pearson, 2002.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name		L	T	P	C
CSGG3120P	Computer Graphics Lab		0	0	2	1
Total Units to be Covered: 10		Total Contact Hours: 30				
Prerequisite(s):	Basic Knowledge of Mathematics like Matrix and Geometry,Good knowledge of C and C++ for OpenGL.		Syllabus version: 1.0			

Course Objectives

1. Acquaint with OpenGL library and understand the graphics code structure with it
2. Apply primitive operations to create 2D and 3D objects and perform various operations on them.
3. Carry out complex 2D and 3D transformations on objects and create curves of 2nd and 3rd
 degree
4. Explore and implement various hidden surface removal techniques.

Course Outcomes

CO1. Demonstrate knowledge of raster and vector graphics, transformations, rendering techniques, and image representation in computer graphics.

CO2. Implement graphics algorithms, handle user input, and create interactive applications using OpenGL.

CO3. Design and animate objects, simulate lighting, shadows, and reflections, and develop visually appealing graphics and special effects.

CO4. Analyze performance bottlenecks, optimize rendering processes, and implement efficient algorithms and data structures in computer graphics.

CO5. Apply computer graphics concepts and techniques to solve real-world problems in fields such as architecture, visualization, simulation, or graphics development.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	-	-	-	-	-	-	-	-	2	1	2
CO 2	-	-	-	-	2	-	-	-	2	-	-	-	2	1	2
CO 3	-	-	-	-	-	-	-	-		-	-	-	2	1	2
CO 4	-	-	-	-	2	-	-	-	2	-	-	-	2	1	2
CO 5	-	-	-	-	2	-	-	-	2	-	-	-	2	1	3
Average	-	-	-	-	1.2	-	-	-	1.2	-	-	-	2	1	2.2

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Introduction to OpenGL: [Lab Environment Setup]

- What is OpenGL?
- What is GLU/GLUT?
- What is OpenGL Architecture?
- Setting up the environment.
- First OpenGL Program: This initializes a window of Green color.
- Draw a Hut.

 # Discuss all the steps & functions in the program.

Experiment 2: Event Handling

- Implement mouse input functionality.
- Implement keypress functionality.
- Implement another call back functions.

  #Implement above with the help of animation.

Imp: For all the remaining experiments, take the input with the help of mouse clicks, don't use console for any type of input.

Experiment 3: Drawing a line [Usage of Open GL on Linux Environment for Virtual Environment]

- Draw a line using equation of line Y=m*X+C.
- Draw a line using DDA algorithm for slope m<1 and m>1.
- Draw a line using Bresenham algorithm for slope m<1 and m>1.

 # Take the input from user for all the three scenarios i.e. value of (x1, y1) and (x2, y2).

Experiment 4: Drawing a Circle and an Ellipse [Done on OpenGL Environment]

- Draw the circle with the help of polar equations
- Draw the circle with the help of mid-point method.
- Draw the Ellipse with the mid-point method.

 # Take the value of radius, major axis and minor axis as input from the user.

Experiment 5 Seed Fill Algorithms [Small Project will be given for demonstration]

- WAP to fill the polygon using scan lines.

- WAP to fill a region using boundary fill algorithm using 4 or 8 connected approaches.
- WAP to fill a region using flood fill algorithm using 4 or 8 connected approaches.

 # Take the value of seed point, intensity of new color as input from user.

Experiment 6: Viewing and Clipping [Geographical Animation for demonstration]

- Write an interactive program for line clipping using Cohen Sutherland line clipping algorithm.
- Write an interactive program for line clipping using Liang-Barsky line clipping algorithm.
- Write an interactive program for polygon clipping using Sutherland – Hodgeman polygon clipping algorithm.

 # Take the window coordinates as input from the user, also take polygon coordinates as input.

Experiment 7 : Basic 2D & 3D Transformations

- Write an interactive program for following basic transformation.
- Translation
- Rotation
- Scaling
- Reflection about axis.
- Reflection about a line Y=mX+c and aX+bY+c=0.
- Shear about an edge and about a vertex.

 # Perform all the experiment for 3-D transformation.
 # Take the following values as input from user: Theta (angle of rotation), translation factor, scaling factor and other values. Make necessary assumptions.

Experiment 8: Drawing Bezier curves. [Virtual GLUT based demonstration]

- Write a program to draw a cubic spline.
- WAP to draw a Bezier curve**.**

 # Take necessary values as input from the user like degree of the Bezier curve.

Experiment 9: VSD and Shading Algorithms.

- Implement VSD Algorithms like, Backface, Z Buffer.
- Implement Raycasting algorithm.

Experiment 10: Illumination and shading Algorithms.

- Implement Phong illumination model for computing light intensity
- Implement Shading algorithms: Flat, Phong and Gourard Shading.

Total Lab hours 30

Textbooks

1. Donald D. Hearn and M. Pauline Baker, “Computer Graphics using OpenGL”, Pearson Education India, 2014.

Reference Books

1. Dave Shreiner, John Kessenich, Bill Licea-Kane, The Khronos OpenGL ARB Working Group, “OpenGL: Programming Guide, the Official Guide to Learning OpenGL”, 8th Edition, Addison Wesley, 2013.

2. Mason Woo, and Dave Shreiner, “OpenGL Programming Guide”, Paperback, 2008.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name		L	T	P	C
CSGG4012P	Augmented and Virtual Reality Development		4	0	0	4
Total Units to be Covered: 05		Total Contact Hours: 60				
Prerequisite(s):	Unity, 3D Modelling and Animation, Data Structures		Syllabus version: 1.0			

Course Objectives

1. Understand the fundamental concepts and principles of Augmented Reality (AR) and Virtual Reality (VR), including their differences, applications, and potential impact on various industries.
2. Gain proficiency in using AR/VR hardware and software tools, including AR/VR devices, headsets, controllers, and development platforms.
3. Develop practical skills in creating AR/VR experiences by learning the necessary techniques and technologies, such as marker-based and marker less AR tracking, VR environment design, and user interaction implementation.
4. Acquire hands-on experience in AR/VR development using the Unity game engine, including asset importing, scene creation, scripting, and performance optimization.
5. Apply user experience (UX) and interaction design principles to create immersive and user-friendly AR/VR interfaces, and understand the challenges and considerations involved in designing and testing AR/VR applications.

Course Outcomes

CO1: Students will demonstrate knowledge and understanding of AR/VR concepts, technologies, and applications.
CO2: Students will apply their knowledge to create AR/VR experiences using industry-standard tools.
CO3: Students will analyze the effectiveness of different AR/VR design choices and techniques. They will assess the impact of these decisions on user experience, performance, and overall quality of AR/VR applications.
CO4: Students will critically evaluate AR/VR applications in terms of usability, user experience, and effectiveness.
CO5: Students will synthesize their knowledge and skills to design and develop original AR/VR applications.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	2	-	1	-	-	-	-	-	-	-	2	-	1
CO 2	1	-	1	-	3	-	-	-	-	-	-	1	-	-	1
CO 3	1	-	2	-	2	-	-	-	2	-	-	-	-	-	3
CO4	1	-	2	-	2	-	-	-	2	-	-	-	-	-	3
CO5	1	-	2	-	3	-	-	-	2	-	-	1	1	-	3
Average	1	-	1.8	-	2.2	-	-	-	2	-	-	1	1.5	-	2.2

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Introduction to Augmented and Virtual Reality **10 Lecture Hours**

Introduction to AR/VR, history of AR/VR, definition of AR/VR, difference between AR/VR, applications of AR/VR, introduction to XR (Extended Reality), future trends of AR/VR, Applications: Metaverse.

Unit II: Tools and Technologies for AR VR Development **10 Lecture Hours**

VR/MR Head mounted Display (Wired/Wireless): Oculus Quest 2, HTC Vive Pro, Oculus Quest Pro, Apple Mixed Reality. Software's: SDKs for AR development, Game Engines,3D Modelling Software, 3D Tracking and Sensors: SLAM (Simultaneous Localization and Mapping), Audio and Spatial Sound Design.

Unit III: AR/VR Development with Unity **24 Lecture Hours**

Introduction to Unity game engine and its AR/VR capabilities, importing assets and creating scenes for AR/VR, Scripting interactions and behaviors in Unity, Marker based AR Development using Vuforia, Marker less AR development using ARfoundation: ARCore/Arkit. Deploying Application on Playstore.

Introduction to Unity XR, VR Interaction Mechanics, VR Locomotion and Navigation, VR User Interface (UI) and User Experience (UX), VR Audio and Spatial Sound. XR Interaction Toolkit: XR Input and Controllers, Performance Optimization for VR, Testing and Deployment of VR Applications.

Unit IV: Designing for Immersive Applications 6 Lecture Hours

Interaction Design for Immersive Experiences, Spatial UI Design, Visual Design for Immersive Applications, Navigation and Wayfinding in Immersive Environments, Comfort and Accessibility in Immersive Design, Usability Testing, and Iterative Design for AR/VR Development.

Unit V: Capstone Project 10 Lecture Hours

Project proposal and planning, research and analysis, design and development, testing and iteration, documentation and presentation.

Total lecture Hours 60

Textbooks

1. Dieter Schmalstieg and Tobias Hollerer, "Augmented Reality: Principles and Practice", Pearson, 2016.
2. Steve Aukstakalnis, "Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR", Addison-Wesley, 2016.

Reference Books

1. Tony Parisi, “Learning Virtual Reality: Developing Immersive Experiences and Applications for Desktop, Web, and Mobile", O'Reilly Media, 2015.
2. Ralf Doerner, Wolfgang Broll, Paul Grimm, Bernhard Jung, “Virtual and Augmented Reality (VR/AR): Foundations and Methods of Extended Realities (XR)”, Springer, 2022.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSGG4112P	Augmented and Virtual Reality Development Lab	0	0	2	1
Total Units to be Covered: 10	Total Contact Hours: 30				
Prerequisite(s):	Unity3D, 3D Modelling	Syllabus version: 1.0			

Course Objectives

1. Understand the fundamental concepts and principles of Augmented Reality (AR) and Virtual Reality (VR).
2. Gain hands-on experience with marker-based AR using Vuforia.
3. Develop skills in marker less AR using ARCore and AR Foundation.
4. Learn VR development using the XR Plugin by Unity.
5. Apply the acquired knowledge and skills to create interactive AR/VR experiences.

Course Outcomes

CO1: Design and develop marker-based AR applications using Vuforia.
CO2: Implement Markerless AR using ARCore and AR Foundation.
CO3: Create immersive VR experiences using the XR Plugin by Unity.
CO4: Evaluate and analyze AR/VR applications for usability and user experience.
CO5: Design and develop an deploy AR and VR Applications for various Platforms.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	1	-	1	-	3	-	-	-	-	-	-	1	-	-	1
CO 2	1	-	2	-	3	-	-	-	2	-	-	1	-	-	3
CO 3	1	-	2	-	3	-	-	-	2	-	-	1	-	-	3
CO4	-	-	-	-		-	-	-	1	-	-	-	-	-	1

CO5	-	-	-	-	2	-	-	-	-	-	-	-	1	1	3
Average	0.6	-	1	-	2.2	-	-	-	1	-	-	.6	.2	.2	2.2

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

Experiment 1: Introduction to Augmented Reality/Virtual Reality/Mixed Reality

1. Install Unity, the development environment for AR/VR/MR.
2. Familiarize yourself with the Unity user interface, components, and editor.
3. Learn the basics of writing scripts in Unity for creating interactive AR/VR/MR experiences.

Experiment 2: Marker-based AR using Vuforia.

1. Install and set up the Vuforia SDK, a plugin for AR development.
2. Create a tracker using Vuforia for marker-based AR.
3. Learn how to detect the tracker and understand the working of AR.
4. Incorporate 3D game objects into Unity scenes for AR applications.

Experiment 3: Advanced Marker-based AR: Multiple Markers

1. Use the Vuforia AR Camera for image-related operations.
2. Explore Multi-Image Targets in Vuforia to detect multiple images simultaneously.
3. Convert any object into an Image Target for AR tracking.
4. Play holographic videos within an augmented environment.
5. Understand Simultaneous Localization and Mapping (SLAM) in Vuforia.
6. Create virtual buttons in Vuforia and explore their applications.

Experiment 4: Create an AR application using AR Foundation

1. Understand the AR Foundation, a cross-platform framework for AR applications.
2. Create a basic AR application using AR Foundation, including adding 3D models, interactivity, and user interface elements.
3. Imple

Experiment 5: Advance AR Foundation

1. Implement light estimation in AR application.
2. Implement distance estimation in AR application.
3. Implement portal functionality in AR application.
4. Implement direct and indirect manipulation.

Experiment 6

1. Pick a use case and implement it either using Marker based or Marker less concept.

Experiment 7: Introduction to VR

1. Learn basics of VR development using Unity
2. Setting up XR Plugin for VR development.
3. Creating a simple VR environment.
4. UI Design for VR Application

Experiment 8: VR Interaction Techniques

1. Implementing VR interaction using controllers
2. Introducing locomotion techniques in VR
3. Implement UI interactions.
4. Incorporating realistic physics and collisions in VR
5. Implementing immersive audio and haptic feedback in VR

Experiment 9&10: Create a VR game incorporating all the functionalities, Build the game for all platforms, Build game for all VR devices.

1. Design and develop a VR game, including gameplay mechanics, objectives, and immersive elements.
2. Implement player movement, interaction mechanics, and game rules to create a compelling gameplay experience.
3. Optimize the game's performance and compatibility for various VR platforms and devices.
4. Test, debug, and iterate on the game to ensure smooth gameplay and fix any issues or bugs encountered.

Build and deploy the VR game for multiple platforms and VR devices, maximizing accessibility and user reach.

Total Lab hours 30

Textbooks

1. Dieter Schmalstieg and Tobias Hollerer, "Augmented Reality: Principles and Practice", Pearson, 2016.
2. Steve Aukstakalnis, "Practical Augmented Reality: A Guide to the Technologies, Applications, and Human Factors for AR and VR", Addison-Wesley, 2016.

Reference Books

1. Jonathan Linowes and Krystian Babilinski, “Augmented Reality for Developers: Build Exciting AR Applications Using Unity, ARKit, ARCore, and Vuforia", Packt Publishing, 2017.
2. Jonathan Linowes, "Unity Virtual Reality Projects", 2nd Edition, Packt Publishing, 2018.

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %

Course Code	Course name	L	T	P	C
CSGG4013P	**Web Programming for Interactive 3D Graphics**	3	0	0	3
Total Units to be Covered: 05	**Total Contact Hours: 45**				
Prerequisite(s):	HTML, Computer Graphics,3D Mathematics	**Syllabus version: 1.0**			

Course Objectives

1. Understand the core principles of web programming.
2. Dive into the low-level details of WebGL, including the graphics pipeline, shader programming.
3. Develop proficiency in using the Three.js library to create complex 3D scenes, manage geometries, materials, and shaders.
4. Gain expertise in optimizing WebGL applications for performance in live Web applications.

Course Outcomes

On completion of this course, the students will be able to:

CO1. Students will be able to apply HTML5, CSS3, and JavaScript to create interactive web applications.
CO2. Students will be able to design and implement complex 3D scenes using the WebGL API and the Three.js library.
CO3. Students will be able to utilize shader programming to implement advanced shading algorithms, such as Phong shading, texture mapping, normal mapping, and physically based rendering (PBR)
CO4. Students will be able to incorporate user interaction in 3D environments by handling input events and implementing object picking.
CO5. Students will be able to integrate web-based 3D graphics with other technologies, such as WebXR, to create immersive experiences and explore the possibilities of virtual reality (VR) and augmented reality (AR) in web applications.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	2	-	-	-	1	-	-	-	-	1	2
CO 2	-	-	-	-	2	-	-	-	1	-	-	-	-	3	2
CO 3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1
CO4	-	-	-	-	-	-	-	-	-	-	-	-	-	2	1
CO5	-	-	-	-	2	-	-	-	2	-	-	-	-	2	2
Average	-	-	-	-	1.2	-	-	-	0.8	-	-	-	-	1.6	1.6

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

Syllabus

Unit I: Front end development for Web **10 Lecture Hours**

Introduction to the course and the History of Web Development, HTML Tags, attributes and forms

Page styling with CSS, JavaScript Basics, if-else, loop, function, DOM manipulation with JavaScript, Responsive Web UI, Bootstrap Introduction with classes, grids, markers, buttons, Bootstrap Components

Unit II: Server-Side Programming **8 Lecture Hours**

Client Side vs Server side programming, Node JS Introduction, features and Installation, Node.js Modules, NPM and file system module, Asynchronous Programming with Node.js: Understanding the event loop and non-blocking I/O, Callbacks and handling asynchronous operations, Promises and async/await for asynchronous flow control, Express.js, Routing, Serving HTML, Images, CSS and static assets, Dynamic pages and templating, Working with Databases (MongoDB), Inserting, updating and deleting documents, REST API with Mongoose, Real-Time Communication with Socket.IO :Introduction to WebSocket communication, Building real-time applications with Socket.IO, Handling events and broadcasting messages

Unit III: WebGL **6 Lecture Hours**

Overview of WebGL and GPU based rendering, WebGL Rendering Pipeline, WebGL Data flow concepts and major steps in programming, Applying basic and composite Transformations in WebGL, Animation loop, Adding color to a Geometry, Adding Texture to a Geometry

Unit IV: Advance WebGL **6 Lecture Hours**

3D Programming Basics, Modeling the real world, Flow of concepts in WebGL, LookAt() method and its working, Transformation and Animation, Adding Lights, Fogs and Shadows.

Unit V: Frameworks on WebGL **15 Lecture Hours**

Type Script Installation and Setting up development environment, THREE JS basics: Scene, Camera, Renderer and Animation Loop, Object3D and Object3D Hierarchy, Basic Geometries in THREE JS, Materials:- MeshBasic, MeshPhong, MeshNormal, MeshLambert, MeshPhysical

Texture mapping in THREE JS, Lights:- Ambient, Directional, Point, Spot Loading External Objects with GLTF, GLTF Animations loadings, Phyiscs JS, Tween JS, Raycasting and Collision Detection, WebXR overview with THREE JS, Acessign gamepad and handling controller events

WebXR API: XRSystem, XRInput, Conclusion of the course with WebGPU

Total lecture Hours 45

Textbooks

1. Bhupendra Singh, “Learning 3D Programming on the Web with WebGL” by
2. Jos Dirksen, "Learning Three.js: Virtual and Augmented Reality with Three.js and WebXR" by

3. Ethan Brown, "Web Development with Node and Express: Leveraging the JavaScript Stack" by

Reference Books

1. Kouichi Matsuda and Rodger Lea, "WebGL Programming Guide: Interactive 3D Graphics Programming with WebGL", Addison-Wesley, 2013.
2. Jos Dirksen, "Learning Three.js: The JavaScript 3D Library for WebGL", Packt Pub Ltd, 2015.
3. Tony Parisi, "WebXR: Building Immersive AR and VR Applications on the Web".

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme

Components	IA	MID SEM	End Sem	Total
Weightage (%)	50	20	30	100

Detailed breakup of Internal Assessment

Internal Assessment Component	Weightage in calculation of Internal Assessment (100 marks)
Quiz 1	15%
Quiz 2	15%
Class Test 1	15%
Class Test 2	15%
Assignment 1/Project	20%
Assignment 2/Project	20%

Course Code	Course name	L	T	P	C
CSGG4113P	**Web Programming for Interactive 3D Graphics Lab**	0	0	2	1
Total Units to be Covered: 10	**Total Contact Hours: 30**				
Prerequisite(s):	HTML5, Computer Graphics,3D Mathematics	**Syllabus version: 1.0**			

Course Objectives

1. Understand the core principles of web programming.
2. Dive into the low-level details of WebGL, including the graphics pipeline, shader programming.
3. Develop proficiency in using the 3.js library to create complex 3D scenes, manage geometries, materials, and shaders.
4. Gain expertise in optimizing WebGL applications for performance in live Web applications.

Course Outcomes

On completion of this course, the students will be able to:

CO1. Students will be able to apply HTML5, CSS3, and JavaScript to create interactive web applications.

CO2. Students will be able to design and implement complex 3D scenes using the WebGL API and the 3.js library.

CO3. Students will be able to utilize shader programming to implement advanced shading algorithms, such as Phong shading, texture mapping, normal mapping, and physically based rendering (PBR)

CO4. Students will be able to incorporate user interaction in 3D environments by handling input events and implementing object picking.

CO5. Students will be able to integrate web-based 3D graphics with other technologies, such as WebXR, to create immersive experiences and explore the possibilities of virtual reality (VR) and augmented reality (AR) in web applications.

CO-PO Mapping

Program Outcomes / Course Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	-	-	-	-	2	-	-	-	1	-	-	-	-	1	2
CO 2	-	-	-	-	2	-	-	-	1	-	-	-	-	3	2
CO 3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1
CO4	-	-	-	-	-	-	-	-	-	-	-	-	-	2	1
CO5	-	-	-	-	2	-	-	-	2	-	-	-	-	2	2
Average	-	-	-	-	1.2	-	-	-	.8	-	-	-	-	1.6	1.6

1 – Weakly Mapped (Low) **2 – Moderately Mapped (Medium)**

3 – Strongly Mapped (High) **"_" means there is no correlation**

List of Experiments

EXPERIMENT 1

TITLE: Introduction to HTML5

Objective: Introduction to HTML5 and CSS

1. Create a static webpage with information about your personnel details.
2. Modify the web page created in 1, adding HTML5 features like nav, footer, header tags.
3. Modify the web page created in 2, separating all the styling related information in a separate CSS file.
4. Modify the above web page created in 3, adding video and audio files.

EXPERIMENT 2

TITLE: HTML5 Forms

Objective: Creating a working website

1. Create a sign-up registration page, asking the necessary details from the user. Consider the case of a college website and the students as the users.
2. For the page created in 1, add a login page and a JavaScript validating the required user's login.
3. For the page created in 2, the user should go to personnel info. Page created in 1.4. Also add new features like session timeout using HTML5 web storage feature

EXPERIMENT 3

TITLE: Introduction to Node JS

<u>Objective</u> :- Understanding the basic and file operations in Node.

1. Create a simple web server in Node JS and write "Hello World" on the console.
2. Create a Node.js program that reads the contents of a text file ("sample.txt") and displays its contents to the console.
3. Build a Node.js script that accepts user input and writes it to a new text file
4. Build a Node.js script that copies the contents of one text file ("source.txt") to another file ("destination.txt"). Ensure that the destination file is created if it doesn't exist, and overwrite its contents if it does.

EXPERIMENT 4

TITLE: RESP API

<u>Objective</u> :- Understanding the basic and file operations in Node.

1. Build a Node.js REST API that allows users to create and retrieve resources. Resources could be simple objects with properties like "id," "name," and "description." Implement endpoints for creating new resources (POST) and retrieving a list of all resources (GET).

2. Extend the previous exercise to include the ability to update (PUT) and delete (DELETE) resources. Implement endpoints to update an existing resource by ID and delete a resource by ID.
3. Enhance the API to support pagination and filtering of resources. Implement query parameters that allow users to specify the number of results per page and filter resources based on certain criteria (e.g., filtering by name or description).
4. Secure your API by adding authentication and authorization. Users should be required to authenticate before creating, updating, or deleting resources. Implement a basic user authentication system using tokens (e.g., JWT) and restrict access to certain routes based on user roles.

EXPERIMENT 5

TITLE: Socket programming

Objective :- Understanding the basic of socket programming in Node.

1. Create a basic chat application using Node.js and the socket.io library. Students should build a real-time chat server and client where multiple users can connect, send messages, and receive messages from others in real-time.

2. Build a Node.js server that sends real-time notifications to connected clients. For example, simulate a notification system where users receive notifications whenever a certain event occurs on the server.

3. Create a collaborative drawing application where multiple users can draw together on a shared canvas in real-time. Each user's drawing actions should be synchronized with others in the room.

4. Design a simple online multiplayer game (e.g., a multiplayer tic-tac-toe or a quiz game). Students should implement game logic, real-time player interactions, and score tracking using Node.js and sockets.

EXPERIMENT 6

TITLE: Introduction to WebGL

Objective :- Understanding the basics of WebGL

1. Write a simple WebGL code displaying circle, triangle and rectangle
2. Modify program 1 and add user inputs for creating different size of circle, triangle and rectangle
3. Write a simple WebGL code allowing user to create a freehand polygon
4. Modify code of 3 and add the options of changing pencil color, size, background and choosing pre-defined shapes like circle, rectangle and ellipse

EXPERIMENT 7

TITLE: Transformation and Viewing in WebGL

Objective :- Able to draw and animate 3-D objects and add realistic effects

1. Write a WebGL program to display a rotating cuboid coloring every surface with different colors

2. For program of 1, add the control of rotation, scaling and translation to the user

3. For program of 2, add the texture to any surface of the cuboid with user defined images stored in

local repository

4. Write a WebGL program displaying ‘n’ number of bouncing balls with ‘n’ given by the user. The
program should include 3 D projection and hidden surface removal effects.

EXPERIMENT 8

TITLE: THREE JS Introduction

Objective :- Able to display shapes, rotate them and apply texture.

1. Write a THREE JS program to display any five primitive shapes available in the library.
2. Write a THREE JS program to display a rotating cuboid with rotation speed depending upon the input by the user through the web interface.

3. Load an external 3D model (e.g., a .gltf or .obj file) which is selected through web based UI into a Three.js scene and display it.
4. With the help of texture apply any skybox images over a cube.

EXPERIMENT 9

TITLE: THREE JS Materials and lights

Objective :- Able to use advance materials and lights

1. Write a THREE JS program to display 3d Models of various types materials.
2. Add various types of lights to your scene.
3. Move and rotate any 3D object in the scene using keypads and mouse.

EXPERIMENT 10

TITLE: Advance topics in THREE JS

Objective :- Able to apply physics and other advanced topics.

1. Write a THREE JS program to demonstrate a bouncing ball using PHYSICS.js
2. Use ray casting to select objects from the scene using mouse buttons.

Total Lab hours 30

Textbooks

1. Bhupendra Singh, “Learning 3D Programming on the Web with WebGL”.
2. Jos Dirksen, "Learning Three.js: Virtual and Augmented Reality with Three.js and WebXR",
3. Ethan Brown, "Web Development with Node and Express: Leveraging the JavaScript Stack", 2nd Edition, Shroff/O'Reilly, 2019.

Reference Books

1. Kouichi Matsuda, and Rodger Lea, "WebGL Programming Guide: Interactive 3D Graphics Programming with WebGL", Pearson, 2013.
2. Jos Dirksen, "Learning Three.js: The JavaScript 3D Library for WebGL", Packt Pub Ltd, 2015.
3. Tony Parisi, "WebXR: Building Immersive AR and VR Applications on the Web".

Modes of Evaluation: Quiz/Assignment/ presentation/ extempore/ Written Examination

Examination Scheme: Continuous Assessment

Components	Quiz & Viva	Performance & Lab Report
Weightage (%)	50 %	50 %