K.R. MANGALAM UNIVERSITY

SCHOOL OF ENGINEERING

AND TECHNOLOGY

B.Tech (Computer Science & Engineering) With Specialization in

Full Stack Development

Undergraduate Course

2023-27

Preamble

Welcome to the School of Engineering and Technology at K. R. Mangalam University!

At the forefront of innovation and academic excellence, the School of Engineering and Technology is a vibrant hub of learning that nurtures aspiring engineers and technologists. Our commitment to fostering a dynamic learning environment, coupled with a passion for pushing the boundaries of knowledge, empowers our students to embark on a transformative educational journey.

With a blend of cutting-edge curriculum, state-of-the-art facilities, and a distinguished faculty, we are dedicated to equipping our students with the skills, insights, and practical experience they need to thrive in a rapidly evolving technological landscape. Our programs are designed not only to impart technical proficiency but also to cultivate critical thinking, creativity, and ethical leadership.

As we embrace the spirit of innovation and discovery, we invite students to engage in hands-on projects, collaborative research endeavours, and experiential learning opportunities. Through industry partnerships, internships, and exposure to real-world challenges, our students gain a holistic understanding of their fields, preparing them to make meaningful contributions to society.

At the School of Engineering and Technology, we believe in fostering a sense of community and camaraderie among students, faculty, and industry professionals. This collaborative ethos encourages the exchange of ideas, the pursuit of excellence, and the development of lifelong connections.

Whether you aspire to be a trailblazing engineer, a tech-savvy entrepreneur, or a visionary researcher, the School of Engineering and Technology is here to nurture your ambitions and empower you to shape a brighter future.

Welcome to a place where innovation knows no bounds, and where your journey towards academic and professional success begins.

Dean,

School of Engineering and Technology,

K. R. Mangalam University.

Preface

The program Bachelor of Technology (B.Tech) in Computer Science & Engineering with a specialization in Full Stack Development is designed to equip students like you with the knowledge and skills necessary to excel in the rapidly evolving field of computer science and become proficient full stack developers. In today's digital age, where technology is at the forefront of innovation and progress, the demand for skilled professionals who can develop, design, and maintain complex software systems is ever-increasing. As a full stack developer, you will be responsible for handling both the front-end and back-end aspects of web development, ensuring seamless functionality and a delightful user experience. The B.Tech program in Computer Science & Engineering with a specialization in Full Stack Development is structured to provide you with a comprehensive understanding of computer science fundamentals, software development principles, and the latest industry practices. From programming languages and algorithms to databases, web development frameworks, and cloud technologies, this program covers a wide range of topics that will help you become a well-rounded and versatile professional. Throughout your journey in this program, you will have the opportunity to work on practical projects, collaborate with industry experts, and gain hands-on experience with cutting-edge tools and technologies. The curriculum is designed to foster critical thinking, problem-solving skills, and creativity, empowering you to tackle real-world challenges in the field of full stack development. Moreover, you will be guided by a team of experienced faculty members who are passionate about their subjects and committed to providing you with a stimulating learning environment. They will mentor and support you in acquiring both the technical expertise and the soft skills necessary for a successful career in the industry. As a student in this program, you will also have access to various co-curricular activities, workshops, and industry partnerships that will enrich your learning experience and expose you to the latest industry trends. You will be encouraged to participate in hackathons, coding competitions, and internships, allowing you to apply your skills in real-world scenarios and build a strong professional network. By the end of this program, you will have developed a deep understanding of full stack development, gained hands-on experience with industry-standard tools and frameworks, and built a portfolio of impressive projects that showcase your abilities. Armed with this knowledge and practical expertise, you will be ready to embark on a successful career as a full stack developer or pursue further research in the field of computer science.

We are excited to have you join our B.Tech program in Computer Science & Engineering with a specialization in Full Stack Development. We believe that this program will empower you to become a competent professional who can contribute to the technological advancements of the future. Get ready to embark on an exciting journey filled with opportunities for growth, innovation, and success.

Objectives of the program

1. Core Computer Science Knowledge: The program seeks to provide students with a solid foundation in fundamental computer science concepts such as algorithms, data structures, programming languages, computer networks, databases, and software engineering. This knowledge forms the basis for understanding the principles and practices in full-stack development.
2. Full Stack Development Expertise: The specialization in Full Stack Development focuses on teaching students how to build end-to-end web applications, covering both the front-end and back-end aspects. Students learn to work with various technologies such as HTML, CSS, JavaScript, front-end frameworks (e.g., React, Angular), and back-end technologies (e.g., Node.js, Python, Ruby on Rails).
3. Hands-on Project Experience: The program emphasizes practical, hands-on learning through real-world projects. Students work on individual and group projects that involve designing, developing, and deploying full-stack applications. This helps them gain practical experience and enhances their problem-solving skills.
4. Industry-Relevant Skills: The curriculum is designed to align with the latest industry trends and requirements. Students are exposed to emerging technologies and industry best practices in full-stack development. This ensures that graduates are well-equipped to meet the demands of the job market.
5. Collaboration and Teamwork: Full-stack development often involves collaboration with cross-functional teams. The program encourages teamwork and communication skills, enabling students to work effectively with designers, front-end developers, back-end developers, and other stakeholders in a software development project.
6. Communication and Presentation Skills: In addition to technical skills, the program emphasizes the development of strong communication and

presentation abilities. These skills are crucial for effectively conveying ideas, collaborating with others, and showcasing one's work in professional settings.

7. Entrepreneurial and Innovative Thinking: The program encourages entrepreneurial and innovative thinking, inspiring students to come up with creative solutions to real-world problems and empowering them to be potential tech leaders or entrepreneurs.
8. Placement and Career Opportunities: Ultimately, the objective of the program is to prepare students for successful careers in the IT industry. This includes facilitating placement opportunities, internships, and industry interactions to bridge the gap between academia and the professional world.

Career Avenues

1. Diverse career avenues available to graduates of the B.Tech Computer Science & Engineering with a specialization in Full Stack Development program are as follows:
2. Full Stack Developer: Graduates can pursue a career as a full stack developer, where they are responsible for designing, developing, and maintaining both the front-end and back-end of web applications. They work with a wide range of technologies, frameworks, and programming languages.
3. Front-end Developer: Front-end developers focus on the user interface (UI) and user experience (UX) aspects of web applications. They work with HTML, CSS, JavaScript, and various front-end frameworks to create visually appealing and interactive interfaces.
4. Back-end Developer: Back-end developers deal with server-side application logic and databases. They work with technologies like Node.js, Python, Ruby on Rails, PHP, and databases like MySQL or MongoDB to manage data and ensure the proper functioning of web applications.
5. Web Application Developer: Graduates can become specialized web application developers, working on the development of web-based software and applications for various industries and domains.
6. Software Engineer: With a strong foundation in computer science and software development, graduates can pursue careers as software engineers, contributing

to the design, development, and maintenance of various software systems beyond web applications.

7. Mobile App Developer: They can also enter the mobile app development industry and create applications for iOS, Android, or cross-platform environments using technologies like React Native or Flutter.
8. DevOps Engineer: DevOps engineers bridge the gap between development and operations teams, focusing on automating and streamlining the software development and deployment processes.
9. Product Manager: With a good understanding of both technical and business aspects, graduates can take on roles as product managers, responsible for guiding the development and improvement of software products.
10. Entrepreneurship: Armed with the skills to build full-stack applications, some graduates might choose to start their own tech ventures or work as freelance developers.
11. UX/UI Designer: A specialization in Full Stack Development also provides a foundation for graduates to move into the field of user experience (UX) and user interface (UI) design, creating intuitive and visually appealing designs for digital products.
12. Data Analyst: Graduates can explore opportunities in data analysis and visualization, utilizing their programming skills to process and analyze large datasets.
13. Cyber security Analyst: With the growing importance of cybersecurity, graduates can pursue roles in securing web applications and systems from potential threats.
14. E-commerce Specialist: E-commerce companies often require developers with full-stack expertise to manage their online platforms and optimize user experiences.
15. Cloud Solutions Architect: As cloud computing becomes increasingly popular, graduates can work as cloud solutions architects, designing and implementing scalable and efficient cloud-based applications.

Duration

4 Years (Full-Time)

Eligibility Criteria

The candidate should have passed 10+2 or its equivalent examination from a recognized Board with a minimum of 50% marks in aggregate. The reservation and relaxation for SC/ST/OBC/PWD and other categories shall be as per the rules of central/state government, whichever is applicable.

K.R. MANGALAM UNIVERSITY

Table of Contents

University Vision & Mission

Vision

KR Mangalam University aspires to become an internationally recognized institution of higher learning through excellence in inter-disciplinary education, research and innovation, preparing socially responsible life-long learners contributing to nation building.

Mission

- Foster employability and entrepreneurship through futuristic curriculum and progressive pedagogy with cutting-edge technology.
- Instill notion of lifelong learning through stimulating research, Outcomes-based education and innovative thinking;
- Integrate global needs and expectations through collaborative programs with premier universities, research centers, industries and professional bodies;
- Enhance leadership qualities among the youth having understanding of ethical values and environmental realities;

School Vision & Mission

Vision

To create, disseminate, and apply knowledge in science and technology to meet the higher education needs of India and the global society, To serve as an institutional model of excellence in scientific and technical education characterized by integration of teaching, research and innovation.

Mission

- To create an environment where teaching and learning are prioritised, with all support activities being held accountable for their success.
- To strengthen the institution's position as the school of choice for students across the State & Nation.
- To promote creative, immersive, and lifelong learning skills while addressing societal concerns.
- To promote co- and extra-curricular activities for over-all personality development of the students.
- To promote and undertake all-inclusive research and development activities.
- To instill in learners an entrepreneurial mindset and principles.
- Enhance industrial, institutional, national, and international partnerships for symbiotic relationships.
- To help students acquire and develop knowledge, skills and leadership qualities of the 21st Century and beyond.

About School

The School of Engineering and Technology at K. R. Mangalam University started in 2013 to create a niche of imparting quality education, innovation, entrepreneurship, skill development and creativity. It has excellent infrastructure, state of the art Labs, and a team of qualified and research-oriented faculty members.

School of Engineering & Technology (SOET) brings together outstanding academicians, industry professionals, and experienced researchers to impart hands-on and multi-disciplinary learning experience. The curriculum of the programs caters to the ever-changing needs and demands of industry. The school has state-of-the-art infrastructure and domain-specific labs.

The school is offering undergraduate programs (B.Tech, BCA, B.Sc), postgraduate programs (M.Tech, MCA) and Ph.D (all disciplines). We are offering B.Tech programs in recent areas of specializations like AI & ML, Data Science, Cyber Security, Automotive Designs & Electrical Vehicle, Sustainable Development & Smart Cities, Full stack development, UI/UX development etc.

Our curriculum, being one of our highlights, has been designed in line with the requirements of the new National Education Policy 2020, Pedagogy of Employment, Sustainable Development Goals, IR 4.0 etc. The curriculum focuses on problem-solving, design, development, and application of various emerging technologies with a focus on innovative teaching learning methodologies. It is our endeavor to constantly evolve curriculum support, so our students stay abreast with the latest updates in this technologically developed world.

SOET aims at transforming the students into competitive engineers with adequate analytical skills, making them more acceptable to potential employers in the country. There is a great focus on experiential & project-based learning with Industry collaborations. Our B.Tech programs are in collaborations with Industries like IBM, Siemens, Samatrix, Xebia, ImaginXP etc. Our students get an opportunity to learn directly by professionals from industry.

Program Outcome (PO)

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

PO 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.

PO 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.

PO 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.

PO 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.

PO 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 employability.

PO 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.

PO 8 Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO 9 Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO 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.

PO 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 through entrepreneurship skills and in multidisciplinary environments.

PO 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 through skill development.

Program Educational Objectives (PEO)

PEO1 - Develop expertise in Full Stack Development and related technologies to pursue a successful career in software development, web application development, or related fields with a focus on lifelong learning.

PEO2 - Apply critical thinking, analytical, and problem-solving skills to develop innovative solutions that address complex real-world problems in the field of Full Stack Development.

PEO3 - To provide our students with the necessary communication and interpersonal skills, multidisciplinary teamwork, and leadership abilities so they can achieve their professional obligations.

PEO4 - To make students aware of current concerns so they can develop morals and sense of responsibility for themselves, their coworkers, society, and the nation.

Program Specific Outcomes (PSO)

PSO1 - Demonstrate expertise in Full Stack Development and related technologies through the ability to design, develop, and deploy high-quality software applications and web applications that meet industry standards and best practices.

PSO2 - Apply critical thinking, analytical, and problem-solving skills to identify and solve complex problems related to Full Stack Development, such as issues related to database design, system integration, security, and performance.

PSO3 - Demonstrate effective communication and interpersonal skills, as well as the ability to work collaboratively in multidisciplinary teams, to accomplish tasks related to Full Stack Development projects.

PSO4 - Develop leadership abilities and take responsibility for professional obligations, such as project management, team coordination, and decision-making, in the context of Full Stack Development projects.

PSO5 - Demonstrate an awareness of ethical and societal concerns related to Full Stack Development and apply moral and ethical principles in the development and deployment of software applications and web applications.

.

Program Highlights

- Specialization in Full Stack Development.
- Strong foundation in core computer science subjects.
- Hands-on project-based learning.
- Industry-experienced faculty.
- Focus on industry-relevant skills and technologies.
- Internship opportunities for practical experience.
- State-of-the-art infrastructure and labs.
- Hackathons and coding competitions.
- Emphasis on soft skills development.
- Industry visits and guest lectures.
- Research and innovation opportunities.

Program Scheme

Semester I

SN	Category	Course Code	Course Title	L	T	P	C
1	Major	ENMA101	Engineering Calculus	3	1	-	4
2	Minor	ENCS101	Fundamentals of Computer programming	4	-	-	4
3	Major	ENPH101/ENCH101	Engineering Physics / Engineering Chemistry	3	1	-	4
4	Major	ENSP103	Software Craftsmanship	4	-	-	4
5	SEC	SEC033	Engineering Drawing & Workshop Lab	-	-	4	2
6	Major	ENPH151/ENCH151	Engineering Physics Lab/ Engineering Chemistry lab	-	-	2	1
7	Major	ENCS151	Fundamentals of Computer Programming Lab	-	-	2	1
8	Minor	ENSP153	Software Craftsmanship Lab	-	-	2	1
9	VAC I		Environmental Studies & Disaster Management	2	-	-	2
TOTAL				16	2	10	23

Semester II

SN	Category	Course Code	Course Title	L	T	P	C
1	Major	ENMA102	Linear Algebra and Ordinary Differential Equations	3	1	-	4
2	Major	ENCH101/ENPH 101	Engineering Chemistry / Engineering Physics	4	-	-	4
3	Major	ENEE101	Basics of Electrical & Electronics Engineering				
4	Minor	ENSP108	Front End Development with ReactJS	4	-	-	4
5	Minor	ENSP158	Front End Development with ReactJS Lab	-	-	2	1
6	Major	ENCH151/ENPH 151	Engineering Chemistry Lab / Engineering Physics lab	-	-	2	1
7	Major	ENEE151	Basics of Electrical & Electronics Engineering Lab	-	-	2	1
8	Open Elective		Open Elective-I	3	-	-	3
9	VAC		VAC II (Along with Community Service)	2	-	-	2
TOTAL				19	2	6	24

Semester III

SN	Category	Course Code	Course Title	L	T	P	C
1	Major	ENCS203	Discrete Mathematics	3	1	-	4
2	Major	ENCS205	Data Structures	4	-	-	4
3	Minor	ENSP201	Backend Development using NodeJS	4	-	-	4
4	Major	ENCS253	DATA STRUCTURES LAB	-	-	2	1
5	Minor	ENSP253	Backend Development using NodeJS Lab	0	0	2	1
6	AEC	AEC011	LIFE SKILLS FOR PROFESSIONALS-i	3	0	0	3
7	INT	ENSI251	Summer Internship/Project-I	-	-	-	2
8	VAC		VAC III	2	0	0	2
9	Open Elective		Open Elective - II	3	0	0	3
TOTAL				18	2	4	24

Semester IV

SN	Category	Course Code	Course Title	L	T	P	C
1	Major	ENCS202	Analysis and Design of Algorithms	4	-	-	4
2	Minor	ENSP208	Test Automation	4	-	-	4
3	Major	ENCS204	Database Management Systems	4	-	-	4
4	Minor	ENSP258	Test Automation Lab	-	-	2	1
5	Major	ENCS254	Database Management Systems Lab	-	-	2	1
6	Major	ENCS256	Analysis and Design of Algorithms Lab	-	-	2	1
7	Open Elective		Open Elective	3	-	-	3
8	VAC		VAC IV	2	-	-	2
9	Project	ENSI252	MINOR PROJECT-I	-	-	-	2
10	AEC	AEC012	LIFE SKILLS FOR [PROFESSIONALS - ii	3	-	-	3
TOTAL				19	1	6	25

Semester V

SN	Category	Course Code	Course Title	L	T	P	C
1	Major	ENCS301	Theory of Computation	3	1	-	4
2	Major	ENCS303	Operating Systems	3	1	-	4
3	Minor	ENSP313	Mobile Application Development	4	-	-	4
4	Major	ENCS305	Software Engineering	4	-	-	4
5	Major	ENCS351	Operating System Lab	-	-	2	1
6	Minor	ENSP363	Mobile Application Development Lab	-	-	2	1
7	INT	ENSI351	Summer Internship/Project - II	-	-	-	2
8	AEC	AEC013	Life Skills for Professionals-III	3	-	-	3
9	SEC	SEC042	New Age Programming Languages lab	-	-	4	2
TOTAL				17	2	8	25

Semester VI

SN	Category	Course Code	Course Title	L	T	P	C
1	Major	ENCS302	Computer Organization & Architecture	3	1	-	4
2	Major	ENCS304	Computer Networks	4	-	-	4
3	Minor	ENSP314	CI/CD & Microservice	4	-	-	4
4	Minor	ENSP316	DevOps & Automation	4	-	-	4
5	Major	ENCS352	Computer Networks Lab	-	-	2	1
6	Minor	ENSP364	CI/CD & Microservice Lab	-	-	2	1
7	Minor	ENSP366	DevOps & Automation Lab	-	-	2	1
8	SEC	SEC036	Competitive Coding Lab	-	-	4	2
9	Project	ENSI352	Minor Project-II				2
TOTAL				15	1	10	23

Note: Practical training will be of eight weeks duration at the end of sixth semester during summer break and the evaluation will be done at the end of seventh semester.

Semester VII

SN	Category	Course Code	Course Title	L	T	P	C
1	Minor		Department Elective-I	4	-	-	4
2	Minor		Department Elective-II	4	-	-	4
3	Minor		Department Elective-I Lab	-	-	2	1
4	Minor		Department Elective-II lab	-	-	2	1
5	Proj	ENSI451	Minor Project-III	-	-	-	2
TOTAL				8	0	4	12
OR							
	INT	ENSI453	Professional Internship from Industry	-	-	-	12

* Students have the choice to do professional Internship of six months from Industry instead of regular courses of 12 credits

Department Elective I (Cyber Security)							
(i)	Minor	ENSP301	Secure Coding and Vulnerabilities	4	-	-	4
	Minor	ENSP351	Secure Coding and Vulnerabilities lab	-	-	2	1
(ii)	Minor	ENSP303	Cyber Crime Investigation & Digital Forensics	4	-	-	4
	Minor	ENSP353	Cyber Crime Investigation & Digital Forensics lab	-	-	2	1
(iii)	Minor	ENSP305	AI in Cyber Security	4	-	-	4
	Minor	ENSP355	AI in Cyber Security Lab	-	-	2	1
(iv)	Minor	ENSP307	Social Media Security	4	-	-	4
	Minor	ENSP357	Social Media Security Lab	-	-	2	1

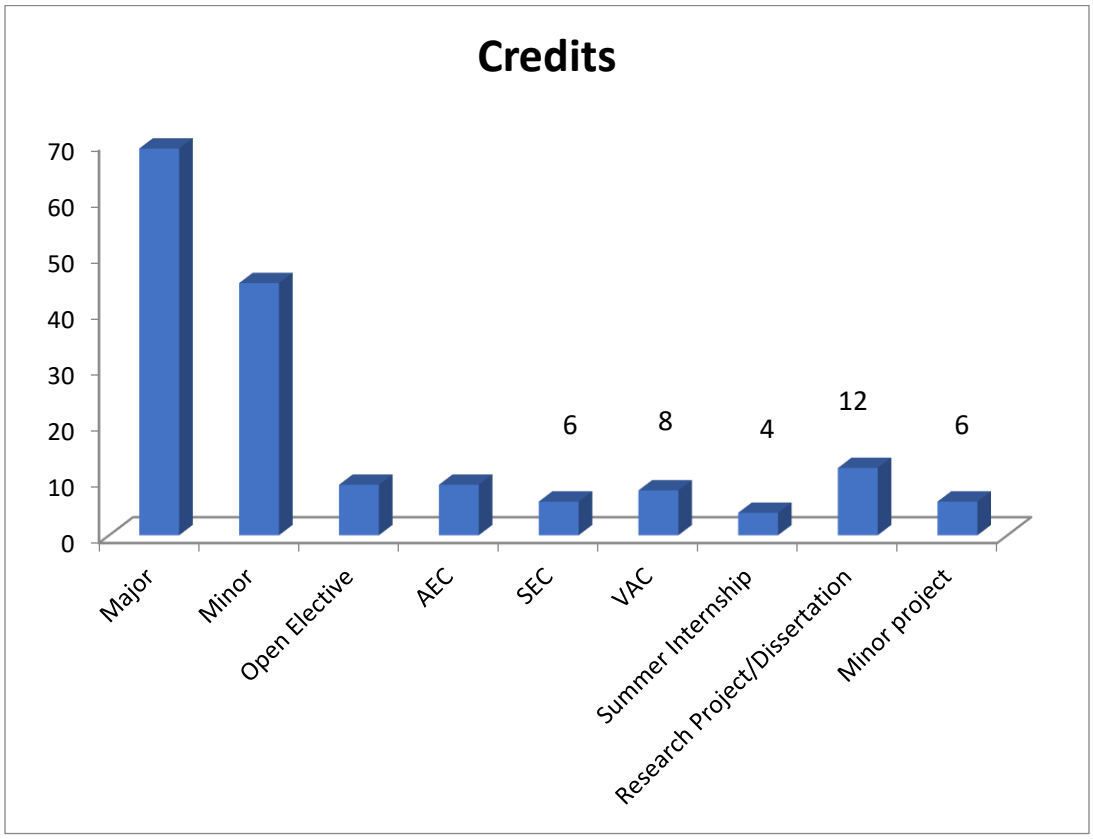
	Department Elective II (Cloud Computing)						
(i)	Minor	ENSP401	Computational Services in The Cloud	4	-	1	4
	Minor	ENSP451	Computational Services in The Cloud Lab	-	-	2	1
(ii)	Minor	ENSP403	Microsoft Azure Cloud Fundamentals	4	-	1	4
	Minor	ENSP453	Microsoft Azure Cloud Fundamentals Lab	-	-	2	1
(iii)	Minor	ENSP405	Storage and Databases on Cloud	4	-	1	4
	Minor	ENSP455	Storage and Databases on Cloud Lab	-	-	2	1
(iv)	Minor	ENSP407	Application Development and DevOps on Cloud	4	-	1	4
	Minor	ENSP457	Application Development and DevOps on Cloud Lab	-	-	2	1

Semester VIII

SN	Category	Course Code	Course Title	L	T	P	C
1	PROJ	ENSI452	Industrial Project/R&D Project/Start-up Project	-	-	-	12
TOTAL							12

Total Credits: 168

Categorization of Courses



Syllabus

Semester: 1

ENGINEERING CALCULUS

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Engineering Calculus	**ENMA101**	3-1-0	4
Type of Course:	Major		
Pre-requisite(s): Calculus knowledge at higher secondary level			
Brief Syllabus: The objective of this course is to familiarize the graduate engineers with techniques in calculus, multivariate calculus, vector calculus and their applications. It aims to equip the students with standard concepts and tools from intermediate to advanced level that will enable them to tackle more advanced level of mathematics and applications that they would find useful in their disciplines.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Differential Calculus- I**	**No. of hours: 10**	
Content Summary: Introduction to limits, continuity and differentiability, Rolle's Theorem, Lagrange's Mean value theorem with their Geometrical Interpretation and applications, Cauchy's Mean value Theorem, Taylor's Series. Applications of definite integrals to evaluate surface areas and volumes of revolutions of curves (Only in Cartesian coordinates), Successive Differentiation (nth order derivatives), Leibnitz theorem and its application, Curve tracing: Cartesian and Polar co-ordinates.			
Unit Number: 2	**Title: Multivariable Calculus (Partial Differentiation and applications)**	**No. of hours: 10**	

Content Summary: Partial derivatives, Total derivative, Euler's Theorem for homogeneous functions, Taylor and Maclaurin's theorems for a function of one and two variables, Maxima and Minima of functions of several variables, Lagrange Method of Multipliers, Jacobians.

Unit Number: 3	Title: Multivariable Calculus-II (Integration)	No. of hours: 10

Content Summary: Evaluation of Double Integrals (Cartesian and polar coordinates), change of order of integration (only Cartesian form), Evaluation of Triple Integrals: Change of variables (Cartesian to polar) for double and (Cartesian to Spherical and Cylindrical polar coordinates) for triple integrals. Applications: Areas (by double integrals) and volumes (by double integrals and triple integrals), Center of mass and center of gravity (Constant and variable densities).

Unit Number: 4	Title: Vector Calculus	No. of hours: 10

Content Summary: Vector differentiation: Gradient, Curl and Divergence and their Physical interpretation, Directional derivatives, Tangent and Normal planes. Vector Integration: Line integral, Surface integral, Volume integral, Gauss's Divergence theorem, Green's theorem, Stoke's theorem (without proof) and their applications.

***Self-Learning Components:**

https://onlinecourses.nptel.ac.in/noc22_ma75/preview

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components

Text Books: -

1. G.B. Thomas and R.L. Finney, Calculus and Analytic geometry, 9th Edition, Pearson, Reprint, 2002.

Reference Books: -

1. B. V. Ramana, Higher Engineering Mathematics, Tata Mc Graw-Hill Publishing Company Ltd., 2008.

2. B. S. Grewal, Higher Engineering Mathematics, Khanna Publisher, 2005.

3. R K. Jain & S R K. Iyenger, Advance Engineering Mathematics, Narosa Publishing House 2002

4. E. Kreyszig, Advance Engineering Mathematics, John Wiley & Sons, 2005.

5. Ray Wylie C and Louis C Barret, Advanced Engineering Mathematics, Tata Mc-Graw-Hill; Sixth Edition.

Other useful resource(s):

1. Link to NPTEL course contents:
https://onlinecourses.nptel.ac.in/noc18_ma05/preview

2. Link to topics related to course:
https://www.whitman.edu/mathematics/calculus_online/chapter14.html

Define Course Outcomes (CO)

COs	
CO 1	To **apply** the knowledge of differential calculus in the field of engineering.
CO 2	To **deal** with functions of several variables those are essential in optimizing the results of real life problems.
CO 3	Multiple integral tools to deal with engineering problems involving centre of gravity, volume etc.
CO 4	To **deal** with vector calculus that is required in different branches of Engineering to graduate engineers.
CO 5	Geometrical approach to the mean value theorems and their **application** to the mathematical problems

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels (C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1	-	P1
CO2	C2	-	P2
CO3	C2	-	P2
CO4	C3, C4	-	P3
CO5	C5	-	P4, P5

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	-	2	-	2	-	-	-	-	-	-	2
CO2	3	3	2	2	-	-	-	-	-	-	-	3
CO3	3	-	-	-	-	-	2	-	-	-	-	2
CO4	-	-	3	-	-	-	-	-		-	2	3
CO5	3	2	-	3	-	-	-	-	-	-	-	2

Justification for mapping must be relevant

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	-	-	-	1
CO2	-	3	-	-	-
CO3	3	-	-	-	-
CO4	-	-	2	-	1
CO5	-	-	-	3	1

Relevance of the Syllabus to various indicators

Unit I	**Differential Calculus- I**
Local	Understanding differential calculus can aid in solving local engineering and scientific problems.
Regional	Understanding differential calculus can be applied to regional projects or research in engineering fields.
National	Differential calculus plays a significant role in various national engineering and scientific endeavors.
Global	Differential calculus is fundamental to global scientific research, engineering projects, and innovations.
Employability	Knowledge of differential calculus enhances employability in various technical and engineering professions.
Entrepreneurship	Knowledge of differential calculus can inspire entrepreneurial opportunities in engineering-based ventures.
Skill Development	Learning differential calculus develops analytical and problem-solving skills for engineering applications.
Professional Ethics	Applying differential calculus with integrity ensures ethical practices in engineering and scientific fields.
Gender	Differential calculus education is equally important for individuals of all genders pursuing engineering.

Human Values	Applying differential calculus with ethical considerations contributes to human-centered engineering solutions.
Environment & Sustainability	Differential calculus can be used to model and optimize environmentally sustainable engineering practices.
Unit II	**Multivariable Calculus (Partial Differentiation and applications)**
Local	Understanding multivariable calculus can aid in solving local engineering and scientific problems.
Regional	Understanding multivariable calculus can be applied to regional projects or research in engineering fields.
National	Multivariable calculus plays a significant role in various national engineering and scientific endeavors.
Global	Multivariable calculus is fundamental to global scientific research, engineering projects, and innovations.
Employability	Knowledge of multivariable calculus enhances employability in various technical and engineering professions.
Entrepreneurship	Knowledge of multivariable calculus can inspire entrepreneurial opportunities in engineering-based ventures.
Skill Development	Learning multivariable calculus develops analytical and problem-solving skills for engineering applications.
Professional Ethics	Applying multivariable calculus with integrity ensures ethical practices in engineering and scientific fields.
Gender	Multivariable calculus education is equally important for individuals of all genders pursuing engineering.
Human Values	Applying multivariable calculus with ethical considerations contributes to human-centered engineering solutions.
Environment & Sustainability	Multivariable calculus can be used to model and optimize environmentally sustainable engineering practices.
Unit III	**Multivariable Calculus-II (Integration)**
Local	Understanding integration can aid in solving local engineering

	and scientific problems.
Regional	Understanding integration can be applied to regional projects or research in engineering fields.
National	Integration plays a significant role in various national engineering and scientific endeavors.
Global	Integration is fundamental to global scientific research, engineering projects, and innovations.
Employability	Knowledge of integration enhances employability in various technical and engineering professions.
Entrepreneurship	Knowledge of integration can inspire entrepreneurial opportunities in engineering-based ventures.
Skill Development	Learning integration develops analytical and problem-solving skills for engineering applications.
Professional Ethics	Applying integration with integrity ensures ethical practices in engineering and scientific fields.
Gender	Integration education is equally important for individuals of all genders pursuing engineering.
Human Values	Applying integration with ethical considerations contributes to human-centered engineering solutions.
Environment & Sustainability	Integration can be used to model and optimize environmentally sustainable engineering practices.
Unit IV	**Vector Calculus**
Local	Understanding vector calculus can aid in solving local engineering and scientific problems.
Regional	Understanding vector calculus can be applied to regional projects or research in engineering fields.
National	Vector calculus plays a significant role in various national engineering and scientific endeavors.
Global	Vector calculus is fundamental to global scientific research, engineering projects, and innovations.

Employability	Knowledge of vector calculus enhances employability in various technical and engineering professions.
Entrepreneurship	Knowledge of vector calculus can inspire entrepreneurial opportunities in engineering-based ventures.
Skill Development	Learning vector calculus develops analytical and problem-solving skills for engineering applications.
Professional Ethics	Applying vector calculus with integrity ensures ethical practices in engineering and scientific fields.
Gender	Vector calculus education is equally important for individuals of all genders pursuing engineering.
Human Values	Applying vector calculus with ethical considerations contributes to human-centered engineering solutions.
Environment & Sustainability	Vector calculus can be used to model and optimize environmentally sustainable engineering practices.
SDG	(SDG) 4, Quality Education.
NEP 2020	Subject aligns with the objectives of the National Education Policy 2020.
POE/4th IR	Subject is relevant to the Fourth Industrial Revolution and emerging technologies.

SOFTWARE CRAFTSMANSHIP

Department:	**Department of Computer Science and Engineering**		
Course Name: **Software Craftmanship**	**Course Code**	**L-T-P**	**Credits**
	ENSP103	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any: None			
Brief Syllabus: This course gives learners a detailed overview of Software Craftsmanship and its associated concepts. It covers the essentials of craftsmanship and code design. The basics of code structure, code formatting and documentation are detailed out. The course also introduces testing and debugging. The learners will be able to explore how to refactor the code and improving code structure, code base, useful software frameworks and tools, and the programming processes.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction**	**No. of hours: 8**	
Content Summary: Definition, History of the emergence of software craftsmanship, Software craftsmanship, Process versus paradigm, Software development processes, Software development models, Software design paradigms, Software development paradigms, Major programming paradigms Procedural programming paradigm, Object-oriented programming paradigm, Functional programming paradigm, Dimensions of craftsmanship, Craftsmanship - Mastery of the paradigm Describing and defining well-crafted code, Becoming a craftsman, The programming process			
Unit Number: 2	**Title: Code Design**	**No. of hours: 6**	

Content Summary: Clean code and its fundamental concepts, Code Design, Software design considerations, Kent Beck's principle of simple design, Fundamental characteristics of good design, Design Patterns: Reusing best practices, SOLID design principles, Programming Principles		
Unit Number: 3	**Title: Code Structure**	**No. of hours: 8**
Content Summary: Classes, packages and methods: building blocks of code, organizing code: the size of methods and classes, what makes methods and classes "good", Software metaphors, Objects and data structures, data transfer objects, Using libraries, Overview of the best practices in structure: Law of Demeter and open close principle		
Unit Number: 4	**Title: Python Programming**	**No. of hours: 6**
Content Summary: Introduction, Object-oriented programming (classes, objects, inheritance), Libraries and packages (e.g., NumPy, Pandas, Matplotlib), String manipulation and regular expressions, List, tuple, dictionary, and set data structures, Working with files and directories (file handling, path manipulation), Debugging and error handling techniques, File handling and input/output operations , Exception handling, Python's standard library and built-in functions, Working with external APIs and web scraping, Database integration with Python (e.g., using SQLite or MySQL)		
Unit Number: 5	**Title: Code Formatting & Documentation**	**No. of hours: 6**
Content Summary: Introduction, Variants, Vertical Openness, Vertical Density, Distance and Ordering, Naming Best Practices, Intention-Revealing Names, Avoid Mental Mappings, Naming Classes, Methods and Functions, Comments, Writing Code Documentation		
Unit Number: 6	**Title: Testing Debugging & Refactoring**	**No. of hours: 8**

Content Summary:
Testing and Debugging, Basic Test-driven Development (TDD), Categories of TDD and Unit tests, Unit Testing Techniques, Automating Testing Using Junit, Refactoring: Improving Structure, Refactoring: Changing Code Structure without Changing Functionality, The need for Refactoring, The Refactoring Process and the Different Levels of Refactoring, Refactoring Strategies, Code Smells: Symptoms of Poorly Designed Code, Categories of Code Smells, Code Base, Using Frameworks & Tools

***Self-Learning Components:**

- Clean Code Principles and Practices
- Develop effective communication skills for technical and non-technical stakeholders**.**

Please Note:

1)Students are supposed to learn the components on self-basis

2)Mention open-source tools/ new concepts/technologies that students will be required to learn and present through presentations in class

3) At least 5-10 % syllabus will be asked in end term exams from self-learning components

Reference Books:

- Robert C Martin, "Clean Code: A Handbook of Agile Software Craftsmanship", O'Reilly
- Brian Allbee, "Hands on Software engineering with Python", O'Reilly
- Pride Sandro Mancuso, "The Software Craftsman: Professionalism, Pragmatism", O'Reilly
- Robert Martin, "Clean Architecture: A Craftsman's Guide to Software Structure and Design" Robert Martin Paperback
- Pete McBreen, "Software Craftsmanship: The New Imperative", Addison-Wesley Professional

SOFTWARE CRAFTSMANSHIP LAB

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Software Craftsmanship Lab	**ENSP153**	0-0-2	1
Type of Course:	Minor		
Pre-requisite(s), if any: Integration/Differentiation			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Understand the significance of clean code principles and how they contribute to maintainable software.
CO 2	Apply SOLID principles to refactor existing code and improve its quality.
CO 3	Assess the impact of automated testing and continuous integration on the software development lifecycle.
CO 4	Design comprehensive software architecture for a complex application, considering scalability and maintainability.

Ex. No	**Experiment Title**	**Mapped CO/COs**
1	Apply SOLID principles and design patterns to code.	CO1
2	Practice TDD by writing unit tests before implementing code.	CO2

3	Use testing frameworks (e.g., JUnit, NUnit) for automated testing.	CO1
4	Identify code smells and apply refactoring techniques.	CO3
5	Set up a CI/CD pipeline for automated builds and deployments.	CO2
6	Use tools like Jenkins or Travis CI for integration and testing.	CO2
7	Pair program with a partner to improve collaboration and code quality.	CO1
8	Conduct effective code reviews to provide feedback and catch issues.	CO3
9	Understand Scrum roles, artifacts, and events.	CO3
10	Improve communication with team members and stakeholders.	CO3
11	Analyze code metrics (e.g., code coverage, cyclomatic complexity).	CO4
12	Apply architectural patterns (e.g., MVC, Microservices) to software projects.	CO4
13	Design and implement modular and maintainable software components.	CO4
14	Use debugging tools and techniques to identify and fix issues.	CO4
15	Learn effective error handling and exception management.	CO3

ENGINEERING PHYSICS

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Engineering Physics	**ENPH101**	3-1-0	4
Type of Course:	Major		
Pre-requisite(s), if any: Integration/Differentiation			
Brief Syllabus: This course provides an introduction to the principles and applications of optoelectronics, as well as an overview of new engineering materials. The course is divided into four units, each focusing on different aspects of the subject matter. By the end of the course, students will have a solid foundation in optoelectronics principles, including lasers and fiber optics, as well as an understanding of new engineering materials and their applications			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Mechanics**		**No. of hours: 10**
Content Summary: Centre of mass, centre of mass of two particle system and a rigid body, Rotational motion, Moment of Inertia and its physical significance, Radius of gyration, Acceleration due to gravity, simple harmonic motion, differential equation of S.H.M., Examples of S.H.M. (simple and compound pendulum)			
Unit Number: 2	**Title: Optics**		**No. of hours: 10**
Content Summary: **Light**: Introduction of light, properties of light, Dual Nature of light, refraction, Refraction by prism, Interference of light, interference by divison of wavefront (Young's double slit experiment), Interference by division of wave amplitude (Newton's ring), difference between diffraction and interference, types of diffraction,			

Fraunhoffer diffraction (single and double slit), theory of plane diffraction grating, determination of wavelength of a spectral line using transmission grating
Laser: Introduction, principle of Laser, stimulated and spontaneous emission, Ruby laser, He-Ne Laser, Application of Lasers.

Unit Number: 3	**Title: Polarization**	**No. of hours: 10**

Content Summary:

Polarization: Polarization by reflection and refraction, Brewster's law, double refraction, nicol prism, quarter and half-wave plates, Production and analysis of circularly and elliptically polarized light

Unit Number: 4	**Title: New Engineering Materials**	**No. of hours: 10**

Content Summary:

Dielectric materials: Definition – Dielectric Breakdown – Dielectric loss – Internal field – Claussius Mossotti relation.

Superconducting materials: Introduction – Properties- Meissner effect – Type I & Type II superconductors – BCS theory-Applications.

Nanomaterials: Introduction – Synthesis of nano materials – Top down and Bottom-up approach- Ball milling- PVD method- Applications. Smart materials: Shape memory alloys-Biomaterials (properties and applications)

***Self-Learning Components:**

Crystal Structure - https://youtu.be/UXqWixel_f8

[1] Classification of solids
[2] Types of crystal systems
[3] Bonding in solids

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning

components

Reference Books:

1. N. Subrahmanyam, B. Lal, M.N.Avadhanulu,Optics (S.Chand).
2. A Text Book of Engineering Physics, M.N.Avadhanulu, P.G. Kshirsagar (S.Chand)
3. Ajoy Ghatak, Optics (Tata McGraw Hill).
4. http://www.gpcet.ac.in/wp-content/uploads/2018/09/UNIT-1-EP-PDF.pdf
5. https://fractory.com/fibre-lasers-explained/
6. https://www.brainkart.com/article/Modern-Engineering-Materials_6830/

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** the principles and applications of lasers, fiber optics, and electromagnetic waves.
CO2	**Apply** the concepts of polarization to analyze and manipulate light in various optical systems.
CO3	**Evaluate** the properties and applications of dielectric materials, superconducting materials, and nanomaterials in engineering contexts.
CO4	**Design** and propose innovative applications of lasers, fiber optics, and smart materials for specific engineering challenges.
CO5	**Analyze and solve** problems related to the behavior of electromagnetic waves, polarization, and optical communication systems**.**

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1	-	P5
CO2	C3	-	P2
CO3	C5	-	P4
CO4	C6	-	P3
CO5	C4	-	P3

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	-	2	-	2	-	-	-	-	-	-	2
CO2	3	3	2	2	-	-	-	-	-	-	-	3
CO3	3	-	--	-	-	-	2	-	-	-	-	3
CO4	-	-	3	-	-	-	-	-		-	2	2
CO5	3	2	-	-	-	-	-	-	-	-	-	3

Justification for mapping must be relevant.

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4
CO1	2	-	-	-
CO2	-	3	-	-
CO3	3	-	-	-
CO4	-	-	2	-
CO5	-	-	-	3

Relevance of the Syllabus to various indicators

Unit I	**Mechanics**
Local	Mechanics is essential for understanding local physical structures, infrastructures, and natural phenomena.
Regional	Relevant to regional infrastructure, architectural planning, and transportation systems.
National	Contributes to national standards in construction, engineering, and transportation.
Global	Integral to global physics research, technological advancements, and architectural standards.
Employability	Fundamental for careers in engineering, physics, automotive design, and other technical fields.
Entrepreneurship	Essential knowledge for any startup in the tech, automotive, or engineering sector.
Skill Development	Helps in enhancing problem-solving, analytical skills, and understanding of physical systems.
Professional Ethics	Understanding and applying mechanics ethically is crucial, especially in construction and tech.
Gender	Mechanics as a subject doesn't directly address gender issues. However, promoting gender equality in mechanical fields is vital.
Human Values	Mechanics fosters an appreciation for the natural laws and the intricacies of the physical world.
Environment & Sustainability	Understanding mechanics can lead to the development of more efficient and sustainable machinery and technologies.
Unit II	**Optics**
Local	The understanding of light and its properties directly affects numerous local tools and technologies, such as eyeglasses, microscopes, and various optical tools. Laser technologies can also be employed in local surgeries, treatments, and commercial applications.

Regional	Optics plays a significant role in regional optical communication systems, architectural designs considering light, and in regional healthcare for surgeries and treatments involving lasers.
National	At the national level, optics becomes integral in broader research initiatives, national defense optics, and broader communication infrastructures. Laser technologies might be used in defense, medical surgeries, and other state-of-the-art applications.
Global	On a global scale, advances in optics, especially laser technologies, play a pivotal role in surgeries, communications, entertainment, research, and defense applications that have international implications.
Employability	Proficiency in optics is essential for careers in optometry, photonics, engineering, research, and more. Understanding laser technology is beneficial for professions in medical technology, defense, entertainment tech, etc.
Entrepreneurship	Innovations in optical technologies present numerous startup opportunities, especially with the rise of augmented reality (AR), virtual reality (VR), and other advanced optical applications. Lasers also offer entrepreneurial opportunities in medical, tech, and cosmetic sectors.
Skill Development	Studying optics boosts the understanding of light behavior, vital in many technical professions. Lasers add an advanced layer to optical understanding.
Professional Ethics	Using optics, especially lasers, requires ethical considerations, especially when used in surveillance, medical, and defense applications to ensure privacy, safety, and correct use.
Gender	While the subject itself doesn't differentiate, there's a broader need to ensure gender equality in optical and laser research, application, and professions.
Human Values	Optics, being a manifestation of natural phenomena, helps instill a sense of wonder and appreciation for the natural world and the intricacies of light. Lasers bring to fore the debate about responsible use vs. misuse.

Environment & Sustainability	Efficient and environmentally conscious use of optical technologies can aid in energy conservation. The sustainable use and disposal of laser equipment also have environmental implications.
Unit III	**Polarization**
Local	Local institutions and organizations involved in optical research or communications can benefit from the understanding and application of polarization, especially in fields like photography and telecommunication.
Regional	Polarization has implications for regional optical communication systems, educational institutions teaching advanced optics, and in medical applications where certain optical tools are used.
National	Polarization plays a role in the national telecommunication infrastructure, defense optics, and broader optical research initiatives.
Global	In the global arena, polarization is integral for cutting-edge optical research, international communications systems, and advanced technologies involving light manipulation.
Employability	Understanding polarization is crucial for careers in photonics, telecommunications, optical research, and other technical fields that involve advanced optics.
Entrepreneurship	Startups or businesses dealing with advanced optical tools, communication systems, or optical research might benefit from innovations in polarization techniques.
Skill Development	Studying polarization aids in honing analytical skills, enhances understanding of light behavior, and offers hands-on experience with optical tools.
Professional Ethics	Correct and ethical application of polarization, especially in surveillance, medical, or defense contexts, ensures the safety and privacy of individuals and data.
Gender	The topic of polarization doesn't inherently address gender issues, but there's an importance to promote gender inclusivity in optical research and professions.

Human Values	Studying the intricacies of polarization can instill a sense of wonder and appreciation for the complexities of the natural world. It underscores the importance of objective observation and the pursuit of knowledge.
Environment & Sustainability	Sustainable use and manufacturing of materials/tools that utilize polarization can contribute to environmental conservation.
Unit IV	**New Engineering Materials**
Local	Local manufacturing and industries can benefit from the understanding and application of new materials for enhanced product quality and efficiency.
Regional	Regional tech hubs, educational institutions, and manufacturing zones can incorporate advanced materials for better product outcomes, innovative research, and technology development.
National	At a national level, understanding and producing these new materials can lead to advanced infrastructure, defense capabilities, health technologies, and improved national R&D standings.
Global	New engineering materials contribute to global advancements in tech, medicine, defense, and more. Countries or entities leading in material research can have a global influence.
Employability	Knowledge in this field opens doors in R&D departments, high-tech industries, medical tech, and more.
Entrepreneurship	Innovations in material science present numerous startup opportunities, from tech gadgets to medical tools, to sustainable solutions.
Skill Development	It aids in the development of research skills, analytical thinking, and hands-on laboratory skills.
Professional Ethics	Correct and ethical applications of new materials, especially in medical and defense, are crucial. Ensuring materials are safe and don't infringe on privacy or security rights is vital.
Gender	While the material itself doesn't differentiate by gender, ensuring gender equality in material research, application, and

	professions is essential.
Human Values	Developing new materials fosters a sense of curiosity, wonder, and appreciation for technological advancements and their positive implications for human life.
Environment & Sustainability	Sustainable production and utilization of these materials, especially nanomaterials, are vital. The development of eco-friendly materials can revolutionize sustainable tech and industries.
SDG	SDG 4
NEP 2020	-
POE/4th IR	-

ENGINEERING PHYSICS LAB

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Engineering Physics Lab	**ENPH151**	0-0-2	1
Type of Course:	Major		
Pre-requisite(s), if any: Integration/Differentiation			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	**Understand** the principles and concepts related to the experiments involving bar pendulum, flywheel, Kater's pendulum, Newton's ring apparatus, plane diffraction grating, spectrometer, and half shade polarimeter.
CO 2	**Apply** the principles and concepts learned to conduct experiments and analyze experimental data, plot graphs, and interpret the results to determine various physical quantities.
CO 3	**Evaluate** the accuracy and reliability of experimental measurements and results obtained from the conducted experiments.
CO 4	**Apply** critical thinking and problem-solving skills to troubleshoot experimental setups, identify sources of errors, and propose solutions to improve the accuracy and precision of measurements

Ex. No	Experiment Title	Mapped CO/COs
1	To plot a graph between the distance of the knife edge from the center of gravity and the time period of the bar pendulum. From the graph, find the acceleration due to gravity, the radius of gyration and the moment of inertia of the bar about an axis.	CO2, CO3
2	To determine the moment of inertia of a flywheel about its own axis of motion.	CO1, CO2, CO3, CO4
3	To determine the value of acceleration due to gravity using Kater`s pendulum.	CO1, CO2, CO3, CO4
4	To determine the wavelength of sodium light using Newton`s ring apparatus.	CO1, CO2, CO3
5	To determine the wavelength of prominent lines of mercury by plane diffraction grating.	CO1, CO2, CO3
6	To determine the refractive index of the material of the prism for the given colours (wavelengths) of mercury light with the help of spectrometer.	CO1, CO2, CO3
7	To determine the specific rotation of cane sugar solution with the help of half shade polarimeter.	CO1, CO2, CO3, CO4
8	To determine the wavelength of He-Ne LASER using transmission diffraction grating.	CO1, CO2, CO3

ENGINEERING DRAWING & WORKSHOP LAB

Department:	**Department of Mechanical Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Engineering Drawing and Workshop Lab	**SEC033**	0-0-4	2
Type of Course:	SEC		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	Statements
CO1	**Understand** the polygons, circles and lines with different geometric conditions
CO2	**Draw** the projection of points, lines and planes under different conditions and orthographic views from isometric views of simple objects
CO3	**Determine** manufacturing methods in different fields of engineering and Practical exposure to different fabrication techniques
CO4	**Creation** of simple components using different materials
CO5	**Exposure** to some of the advanced and latest manufacturing techniques being employed in the industry.

Ex. No	Experiment Title	Mapped CO/COs
	Engineering Drawing	
1	To study Drawing Instruments and their uses, BIS conventions, Lettering, Dimensioning line conventions and free hand practicing, AUTO CAD Commands.	CO1
2	Orthographic Projection: Introduction, Definitions - Planes of projection, reference line and conventions employed, Projections of points in all the four quadrants.	CO1
3	Study of Projections of straight lines True and apparent lengths, True and apparent inclinations to reference planes.	CO1
4	To study orthographic projection of plane surfaces.	CO2
5	To study orthographic projection of solids with examples.	CO2
6	To study sections and development of surfaces of solids with examples.	CO1
7	To study conversion of pictorial views into orthographic projections with examples.	CO1, CO2
8	To study isometric projections of solids with examples.	CO1, CO2
	Workshop:	
9	To make Different types of joints in carpentry shop.	CO, CO4
10	To make Double V-Butt and Lap joint in welding shop.	CO3, CO4
11	To prepare a Job on Lathe machines with step turning and chamfering operation.	CO3, CO4
12	To prepare a Job on Shaper/milling/grinding for finishing of a job.	CO3
13	To prepare a practice job in fitting shop.	CO3
14	To Study about the various machine tools	CO5
15	To make saw - cut filling V-cut taper at the corners, circular cut in fitting shop.	CO4

FUNDAMENTALS OF COMPUTER PROGRAMMING

<table>
<tr><td>Department:</td><td colspan="3">Department of Computer Science and Engineering</td></tr>
<tr><td rowspan="2">Course Name:

Fundamentals of Computer Programming</td><td>Course Code</td><td>L-T-P</td><td>Credits</td></tr>
<tr><td>ENCS101</td><td>4-0-0</td><td>4</td></tr>
<tr><td>Type of Course:</td><td colspan="3">Major Course</td></tr>
<tr><td colspan="4">Pre-requisite(s), if any: None</td></tr>
<tr><td colspan="4">Brief Syllabus:

Fundamentals of Computer Science, Python Introduction, Data Types, Operators, Python data Structures, Conditional Statements, Loops in Python, Functions, OPPS Concepts, basics of Data Pre-processing, Classification and Visualization in Python</td></tr>
<tr><td colspan="4">UNIT WISE DETAILS</td></tr>
<tr><td>Unit Number: 1</td><td colspan="2">Title: Introduction to Computer Fundamentals</td><td>No. of hours: 8</td></tr>
<tr><td colspan="4">Content Summary:

Introduction to Computers and Computing, Von Neumann architecture, Central Processing Unit (CPU): ALU, control unit, registers, Memory: primary and secondary storage, cache memory, virtual memory, Input/output devices: keyboard, mouse, monitor, printer, etc., Introduction to operating systems
Types of operating systems: batch processing, time-sharing, distributed, real-time, etc., Binary, decimal, and hexadecimal number systems, Bits, bytes, and data representation, Character encoding: ASCII, Unicode, Storage devices: hard disks, solid-state drives, optical storage, Introduction to algorithms and complexity analysis, Introduction to software development life cycle (SDLC)
Overview of computer security, Types of threats: viruses, malware, phishing
Ethical considerations in computer use</td></tr>
</table>

Unit Number: 2	Title: Introduction to Python programming & its Environment	No. of hours: 8

Content Summary:

Python Features, Local Environment Setup, Installing Python, Setting up PATH, Python Syntax, Keywords, Understanding Variables, Data Types: Scalar Types, Sequence Type, Mapping Type, Set Types, Mutable Types: List, Dictionary, Set, Immutable Types: Numbers, String, tuple, Operators: Arithmetic, Assignment, Comparison, Logical, Identity, Membership, Bitwise, Basic of String, Manipulating strings, Modify Strings, String Concatenation, Format – Strings, Escape Characters, Inbuilt method of Strings, Basic of Regular Expressions

Unit Number: 3	Title: Condition and Control Structures, Functions and Recursion	No. of hours: 8

Content Summary:

Condition: If, If. Else and nested if, Loops: For, while loops, Nested loops, Enumerate, Break, Continue Statement. Sequence and Iterable Objects, Randomization

Function calls, type conversion and coercion, math functions, adding new function, parameters and argument, recursion and its use, Recursive Fibonacci, Tower of Hanoi, Lambda Functions and Anonymous Functions in Python

Unit Number: 4	Title: Objects and Classes	No. of hours: 8

Content Summary:

Basics of Object-Oriented Programming, Creating Class and Object, Constructors in Python – Parameterized and Non-parameterized, Inheritance in Python, In built class methods and attributes, Multi-Level and Multiple Inheritance, Method Overriding and Data Abstraction, Encapsulation and Polymorphism.

Unit Number: 5	Title: Data Pre-processing, Classification and Visualization in Python	No. of hours: 8

Content Summary:

Data Pre-processing: Data cleaning - Missing Values Noisy Data, Data Cleaning as a Process, Data Integration, Data Reduction, Data transformation and Data Discretization.

Data visualization: Introduction to Matplotlib & Seaborn Libraries, basic plotting, various charts

Mini Project: Students to work on good size dataset, apply data preprocessing and cleaning, write data retrieval queries, apply various data exploration & visualization techniques.

***Self-Learning Components:**

Students are expected to gain working knowledge of various IDE like Anaconda, Pytorch, Spider etc, various components of Anaconda Navigator. Use Github, Kaggle, Google Colab etc

**students will demonstrate the self-learning components through classroom presentations*

Please Note:

1)Students are supposed to learn the components on self-basis

2)Mention open-source tools/ new concepts/technologies that students will be required to learn and present through presentations in class

3) At least 5-10 % syllabus will be asked in end term exams from self-learning components

Reference Books:

TEXTBOOKS:

1. John V Guttag. "Introduction to Computation and Programming Using Python", Prentice Hall of India

Reference Books

1. R. Nageswara Rao, "Core Python Programming", Dreamtech

2. Wesley J. Chun. "Core Python Programming, Second Edition", Prentice Hall

3. Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser, "Data

Structures and Algorithms in Python", Wiley

Web References

https://www.tutorjoes.in/python_programming_tutorial/

https://www.udemy.com/course/100-days-of-code/

https://favtutor.com/blog-details/7-Python-Projects-For-Beginners

https://github.com/NaviRocker/100-days-of-python

https://hackr.io/blog/python-projects

Define Course Outcomes (CO)

COs	Statements
CO1	Acquire an **understanding** of Computer fundamental concepts and Python programming language features, working environment
CO2	**Demonstrate** proficiency in Python syntax and effectively utilize keywords in programming.
CO3	**Analyze** and **solve** programming problems that require the use of conditional statements and looping structures
CO4	**Apply** the concepts of functions, recursion & OOPS in problem solving
CO5	Be able to **implement & demonstrate** the concepts of data Pre-processing and data visualization techniques on real datasets.
CO6	**Involve** in self-Learning & be **able to develop** mini-projects in python

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels (C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1		P1
CO2	C1		P1, P2
C03	C2		P3
CO4	C3		P3
CO5	C6		P3
CO6	C6		P3

CO-PO Mapping

PO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO1 0	PO1 1	PO1 2
CO1	1	1	1	-	-	-	-	-	-	-	-	3
CO2	-	-	-	-	1	-	-	1	1	-	-	3
CO3	-	1	1	1	2	-	-	1	1	-	-	3
CO4	-	2	2	1	2	-	-	1	1	-	-	3
CO5	-	2	2	2	3	2	-	1	1	-	-	3
CO6	-	3	3	3	3	3	3	2	3	2	3	3

Justification for mapping must be relevant

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	1	-	1	2
CO2	3	1	-	1	1
CO3	3	2	-	1	-
CO4	3	2	1	2	-
CO5	3	3	3	3	-

Relevance of the Syllabus to various indicators

Unit I	Mapping
Local	All topics in the syllabus are relevant at the local level as they provide a foundational understanding of computer sc fundamentals & programming.
Regional	All topics in the syllabus can be applicable at the regional level, where individuals and organizations in a specific region utilize Python for various purposes.
National	All topics in the syllabus can be applicable at the national level, as Python is widely used across different industries and sectors.
Global	All topics in the syllabus can be applicable at the global level, as Python is a popular programming language used worldwide.
Employability	All topics in the syllabus, especially Python features, syntax, data types, and string manipulation, contribute to enhancing employability as Python is widely used in the job market.

Entrepreneurship	Understanding Python features, syntax, and data types can be beneficial for entrepreneurs who want to develop software solutions or build web applications.
Skill Development	All topics in the syllabus contribute to skill development in Python programming, including understanding variables, data types, operators, and string manipulation.
Professional Ethics	Professional ethics are not directly addressed in the syllabus topics but are important for ethical behaviour in any professional field.
Gender	NA
Human Values	Human values are not directly addressed in the syllabus topics but can be integrated into the teaching and learning process to emphasize the importance of ethical and responsible programming practices.
Environment & Sustainability	NA
Unit II	
Local	All topics are relevant at local level as programming problems of local nature can be solved using conditional & looping constructs in problem solving
Regional	All topics are relevant at regional level as programming problems of Regional level may utilize conditional & looping constructs in problem solving
National	All topics are relevant at National level as programming problems of national level may utilize conditional & looping constructs in problem solving
Global	All topics are relevant at global level as programming problems of global level may utilize conditional & looping constructs in problem solving
Employability	Conditional & Looping structures are very useful programming tools in problem solving questions during placement drives

Entrepreneurship	Conditional & Looping structures are very useful programming tools in problem solving, any startup involving programming concepts will be useful.
Skill Development	Conditional & Looping structures are very useful programming tools in problem solving and thus skill development.
Professional Ethics	Writing programming codes/solutions will always use conditional & looping structures. There are well defined professional ethics for writing codes.
Gender	NA
Human Values	Any computational solution created with applications for humankind will always involve these programming constructs.
Environment & Sustainability	All computational solutions mapped with environment & sustainability will somehow involve these constructs at the backend.
Unit III	
Local	All topics are relevant at local level as programming solutions at local nature can always use concepts of functions & recursions in problem solving
Regional	All topics are relevant at regional level as programming solutions at regional nature can always use concepts of functions & recursions in problem solving
National	All topics are relevant at national level as programming solutions at national level can always use concepts of functions & recursions in problem solving
Global	All topics are relevant at global level as programming solutions at global nature can always use concepts of functions & recursions in problem solving
Employability	Topics are quite relevant for placements
Entrepreneurship	Working Knowledge of topics can be a advantage for entrepreneurs

Skill Development	Topics can be quite useful for skill development on Programming
Professional Ethics	Writing codes involving functions can be related to coding ethics
Gender	NA
Human Values	NA
Environment & Sustainability	Related applications may be mapped with environment and sustainability
Unit IV	
Local	Solutions involving Descriptive analytics using data visualization techniques can have a local reference.
Regional	Regional reference is possible
National	National reference is possible
Global	Global aspect is also possible
Employability	Skills attained for developing applications using pandas, data visualizations have direct mapping with employability
Entrepreneurship	Mapping of this aspect is also possible. Startup problems can be using these learning aspects
Skill Development	Topics can be quite useful for skill development on descriptive analytics
Professional Ethics	descriptive analytics is related to presenting thoughts & values in ethical manner
Gender	NA
Human Values	NA
Environment & Sustainability	Applications/problem handling based on environment & sustainability issues
SDG	All SDG goals can be touched upon through development /problem solving of related issues

NEP 2020	All indicators of NEP 2020 can be touched upon through development /problem solving of related issues
POE/4th IR	All indicators of POE & 4th IR can be touched upon through development /problem solving of related issues
UNIT V	
Local	Data Cleaning (Missing Values, Noisy Data): These techniques are applicable at the local level to improve the quality of data for local applications and analysis. Data Integration: This process can be used to combine and integrate locally collected data from multiple sources for analysis and decision-making.
Regional	Data Reduction: By reducing the dimensionality or size of the data, regional organizations can efficiently process and analyze data within their specific context.
National	Data Transformation and Data Discretization: These techniques can be applied to national datasets to transform and discretize the data for further analysis and decision-making. Mini Project: The mini project, which involves data preprocessing, exploration, and visualization, can be conducted using national datasets to address relevant national-level issues or challenges.
Global	Data Cleaning as a Process: Data cleaning is a crucial step in ensuring the quality and reliability of global datasets that are used for international research, collaborations, and decision-making
Employability	Data preprocessing, cleaning, and visualization skills are highly valuable in various industries and sectors, enhancing employability prospects.
Entrepreneurship	Data pre-processing and visualization skills are essential for entrepreneurs to gain insights from data, make data-driven decisions, and develop innovative solutions.
Skill Development	The syllabus provides an opportunity for students to

	develop skills in data preprocessing, data cleaning, data integration, and data visualization, which are in high demand in the current job market.
Professional Ethics	The syllabus includes aspects of data cleaning and integration, which contribute to maintaining data integrity and upholding professional ethics in data analysis and decision-making processes.
Gender	The syllabus does not specifically address gender-related aspects.
Human Values	The syllabus indirectly promotes human values by emphasizing the importance of data quality, integrity, and accuracy in decision-making processes.
Environment & Sustainability	The syllabus does not directly address environmental or sustainability aspects.
SDG	The syllabus can indirectly contribute to various SDGs, such as SDG 4 (Quality Education) by providing students with essential data processing and analysis skills.
NEP 2020	The syllabus aligns with the objectives of NEP 2020 by focusing on the development of practical skills, project-based learning, and application of data analysis techniques.
POE/4th IR	The syllabus aligns with the demands of the fourth industrial revolution by equipping students with data pre-processing, cleaning, and visualization skills that are essential for data-driven decision-making and automation.

FUNDAMENTALS OF COMPUTER PROGRAMMING LAB

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Fundamentals of Computer Programming lab	**ENCS151**	0-0-2	1
Type of Course:	Major Course		

Defined Course Outcomes

CO1	Demonstrate the problem-solving skills using variables, basic data types, python data structures, operators, conditional statements, looping structures, functions etc.
CO 2	Solve real world problems involving the concepts based on OOPS
CO 3	Solve Real world problems using pandas, data visualization libraries in python
CO 4	Work on real world data sets and develop mini projects in python

Proposed Lab Experiments

S. N	Experiment	Mapped COs
1	Working with variables, operators in Python to manage inputs	CO1
2	understanding datatypes & formatting commands	CO1
3	Control Flow & Loops in Python	CO1

4	Python Data Types- List & Tuples	CO1
5	Python Data Types- Dictionary, Sets	CO1
6	Python Functions and Recursion	CO1
7	Python Object Oriented Programming: Class and Constructor	CO2
8	Python Object Oriented Programming: Inheritance, Methods Method Overriding and Data Abstraction, Encapsulation and Polymorphism	CO2
9	Handling data in Python	CO3
10	Data Visualization with various python packages	CO3
11	Mini Projects (Suggested): Project 1: Interactive Basic Calculator: Create a calculator that accepts two numbers and an operator (+,-,/,*,&,<,>,// etc) using keyboard. Depending on operator, calculator must calculate the appropriate answer Project 2: Create a formatted student marksheet. Refer to your marksheet format Project 3: Guess the Number game Project 4: Dice Rolling Simulator Project 5: **Rock, Paper, Scissors Game** Project 6: Contact Management System: Project 7: Sudoku Solver Project 8: Bank Account Management System: Project 9: Student Management System Project 10: Library Management System: Project 11: Employee Payroll System: Projects 12: Contact Management System with Inheritance: Project 13: Banking System with Multiple Inheritance: Project 14: School Management System with Multi-Level Inheritance:	CO4

Session 1: Working with variables, operators in Python to manage inputs
▪ Rules for creating variables in Python

▪ Comments
▪ Handling user inputs from keyboard using 'input' command and printing on console
▪ Multiple assignment
▪ Basics of how memory is allocated to variables in python.
▪ Mathematical Operators in Python: Arithmetic, Logical, shift, identity, membership, bitwise
Exercises:
1. Your first program: Say Hello
2. Calculate the multiplication and sum of two numbers
3. Print the sum of the current number and the previous number
4. Exercise on python operators & input command
5. Create a BMI calculator with Python
Project 1: Interactive Basic Calculator: Create a calculator that accepts two numbers and an operator (+,-,/,*,&,<,>,// etc) using keyboard. Depending on operator, calculator must calculate the appropriate answer
Session 2: understanding datatypes & formatting commands
▪ Python Primitive Data Types
▪ Type Error, Type Checking and Type Conversion
▪ Number Manipulation and Strings in Python
▪ Formatting numbers & strings
▪ Random number generation in python
Exercises:
1. Exercise on primitive data types
2. Exercise on number & string manipulation
3. Exercise on Type Error, Type Checking and Type Conversion
4. Exercise on random number generation in python
5. Convert Decimal number to octal using print() output formatting
6. Display float number with 2 decimal places using print()
7. Accept any three string from one input() call
8. Format variables using a string.format() method.

Project 2: Create a formatted student marksheet. Refer to your marksheet format
Session 3 & 4 : Control Flow & Loops in Python
▪ Control Flow with if / else and Conditional Operators
▪ Nested if statements and elif statements
▪ Multiple If Statements in Succession
▪ Loops in python,
▪ Concept of break, continue, pass statement
Exercises:
1. Program which asks the user for a number. If number is even print 'Even', else print 'Odd'.
2. Program to print the largest of the three numbers
3. Program that accepts marks of a student & calculates his percentage & grade. Make suitable assumptions
4. program to print counting from 1 to 10
5. program which prints all the divisors of a number.
6. program to check if input number is a prime number
7. Write a program to print all the numbers between 1000 and 2000 which are divisible by 7 but are not a multiple of 5.
8. program to find greatest common divisor (GCD) or highest common factor (HCF) of given two numbers.
9. Write a program to calculate factorial of a number.
10. program to print multiplication table of a given number
11. Count the total number of digits in a number
12. Display numbers from -10 to -1 using for loop
13. Use else block to display a message "Done" after successful execution of for loop
14. Program to display all prime numbers within a range
15. Display all Armstrong numbers in a given range
16. Display Fibonacci series up to 10 terms
17. Calculate the cube of all numbers from 1 to a given number
18. Find the sum of the series upto n terms
19. Print the following pattern
1

1 2
1 2 3
1 2 3 4
1 2 3 4 5
20. program to use for loop to print the following reverse number pattern
5 4 3 2 1
4 3 2 1
3 2 1
2 1
1
21. Write a program to print the following start pattern using the for loop
*
* *
* * *
* * * *
* * * * *
* * * *
* * *
* *
*
22. [Interactive Coding Exercise] Leap Year
23. [Interactive Coding Exercise] Pizza Order Practice
Project 3: Guess the Number game

create a program in which the system will choose a random number between any ranges defined, and then the user is given a hint to guess the number. Every time the user guesses the number wrongly, he is given another clue leading him toward the answer. The clue can be of any type like smaller, greater, multiples, dividers, etc.
Project 4: Dice Rolling Simulator
As the name of the program suggests, this project will be imitating a rolling dice. This python project will generate a random number each time the dice is rolled and also the user can repeat this program as long as he wants. The program is projected in such a way that when the user rolls a die, the program will generate a random number between 1 and 6.
The program will use the in-build function to generate the random number for rolling dice. It will also ask the user if they wish to roll the dice again.
Session 5: Python Data Types- List & Tuples
▪ list operations and manipulations
▪ list & tuple functions
▪ list slicing
▪ list comprehension
▪ Tuples
▪ List VS Tuples
▪ List & Tuple methods
▪ Uses of List & Tuples
Exercises:
1. Reverse a list in Python
2. Concatenate two lists index-wise
3. Turn every item of a list into its square
4. Concatenate two lists in the following order
5. Iterate both lists simultaneously
6. Remove empty strings from the list of strings
7. Add new item to list after a specified item

8. Extend nested list by adding the sublist
9. Replace list's item with new value if found
10. Remove all occurrences of a specific item from a list.
11. Python program to get the smallest number from a list
12. Python program to count the number of strings where the string length is 2 or more and the first and last character are same from a given list of strings
13. Implement the concept of stacks & queues with lists
14. Write a Python program to remove duplicates from a list
15. Write a Python program to clone or copy a list
16. program to find the list of words that are longer than n from a given list of words
17. Program that get two lists as input and check if they have at least one common member
18. program to print a specified list after removing the 0th, 4th and 5th elements. (enumerate)
19. program to print the numbers of a specified list after removing even numbers from it
20. program to shuffle and print a specified list (shuffle)
21. program to generate all permutations of a list in Python. (itertools)
22. program to convert a list of characters into a string
23. program to find the index of an item in a specified list
24. Python program to add a list to the second list
25. program to check whether two lists are circularly identical
26. program to find the second smallest number in a list
27. program to get unique values from a list
28. Program to get the frequency of the elements in a list.
29. Create a list by concatenating a given list which range goes from 1 to n
30. Program to find common items from two lists
31. Program to Extract elements with Frequency greater than K
32. program to check if the list contains three consecutive common numbers in Python
33. program to Replace all Characters of a List Except the given character
34. Program to Prefix frequency in string List
35. program to reverse All Strings in String List
36. Program to Swap elements in String list
37. Program to Retain records with N occurrences of K
38. Program to Remove Consecutive K element records
39. Copy specific elements from one tuple to a new tuple
40. Check if all items in the tuple are the same

41. Create acronyms using Python
42. Python program to find tuples which have all elements divisible by K from a list of tuples
43. Python program to find Tuples with positive elements in List of tuples
44. Python – Count tuples occurrence in list of tuples
45. Python – Removing duplicates from tuple
46. Python – Remove duplicate lists in tuples (Preserving Order)
47. Python – Cross Pairing in Tuple List
48. Python – Consecutive Kth column Difference in Tuple List
49. Python – Kth Column Product in Tuple List
50. Python – Flatten tuple of List to tuple
Project 5: Rock, Paper, Scissors Game
We have always played rock, and paper scissors game with our playmates. But what if your playmate is not available? Then you can play rock, paper, scissors along with your computer that is designed by you. It is one of the most fun python project ideas for beginners to build their skills.
In this program, we will use a random function for generating the random output by the computer side. The user will make the first move and then the program makes one. Then a function will check the validity of the move. At last, we will display the result and ask the user to play again or not.
Session 6: Python Data Types- Dictionary, Sets
▪ Create a dictionary
▪ Add Elements to a Python Dictionary
▪ Accessing Elements from Dictionary
▪ Removing elements from Dictionary
▪ Python Dictionary Methods
▪ Create a Set in Python
▪ Duplicate Items in a Set
▪ Add and Update Set Items in Python
▪ Add, update items
▪ Remove an Element from a Set
▪ Built-in Functions with Set

Exercises
▪ program to check whether a given key exists in a dictionary or not.
▪ program to iterate over dictionary items using for loop.
▪ program to print only keys & values of a dictionary.
▪ program in python to map 2 lists into a dictionary.
▪ program to remove a set of keys.
▪ program to sort dictionary by values (Ascending/ Descending).
▪ program to concatenate two dictionaries to create one.
▪ program to sum all the values of a dictionary.
▪ program to get the maximum and minimum value of dictionary.
▪ program to check if a dictionary is empty or not.
▪ program to sort dictionary values in python.
▪ program to check whether a key exists in the dictionary or not.
▪ program in python to map keys to dictionary.
▪ program in Python to remove repetitive items from a list.
Session 7: Python Functions and Recursion
▪ Learn Syntax for function calls in Python.
▪ Call built-in functions and user-defined functions.
▪ Understand the return value of a function call.
▪ Explicit type conversion using functions like int(), float(), str().
▪ Implicit type coercion in Python operations.
▪ Exploring the math module in Python.
▪ Commonly used math functions like sqrt(), sin(), cos(), etc.
▪ Defining and implementing user-defined functions in Python.
▪ Syntax for function definition, including the function name, parameters, and return statement.
▪ Understanding the concept of recursion.
▪ Recursive function calls and their relationship with the call stack.
▪ Identifying base cases and recursive cases in recursive functions.
▪ Implementing a recursive function to calculate the Fibonacci sequence.
▪ Understanding the mathematical definition of the Fibonacci sequence.
▪ Analyzing the efficiency and limitations of the recursive solution.
▪ Implementing a recursive function to solve the Tower of Hanoi puzzle.
▪ Understanding lambda functions as anonymous functions in Python.
▪ Using lambda functions in conjunction with built-in functions like map(), filter(), and reduce().

Exercises
Exercise 1: Function Calls and Type Conversion:
▪ Write a program that takes user input of two numbers, performs addition, and displays the result.
▪ Implement error handling to handle cases where non-numeric input is provided.
Exercise 2: Math Functions:
▪ Create a program that calculates the area of a circle using the math module's functions.
▪ Prompt the user for the radius and display the calculated area.
Exercise 3: Adding New Function:
▪ Write a program that converts temperature from Celsius to Fahrenheit and vice versa.
▪ Implement two separate functions for the conversion formulas and test them.
Exercise 4: Parameters and Arguments:
▪ Create a program that calculates the total cost of a meal, including tax and tip.
▪ Define a function that takes the cost of the meal, tax percentage, and tip percentage as arguments.
Exercise 5: Recursion and Fibonacci:
▪ Write a program that calculates the Fibonacci sequence up to a given term using recursion.
▪ Prompt the user for the desired term and display the Fibonacci sequence.
Exercise 6: Tower of Hanoi:
▪ Implement a program that solves the Tower of Hanoi puzzle using recursion.
▪ Prompt the user for the number of disks and display the steps required to solve the puzzle.
Exercise 7: Lambda Functions and Anonymous Functions:
▪ Write a program that uses lambda functions to perform basic arithmetic operations.
▪ Prompt the user for two numbers and allow them to choose the operation to perform.

Project 6: Contact Management System:
• Create a contact management system that allows users to add, search, update, and delete contacts.
• Implement the functionalities using functions for each operation.
• Provide options for sorting and displaying contacts using different criteria, such as name or date added.
Project 7: Sudoku Solver:
• Develop a program that can solve Sudoku puzzles using recursion.
• Write a recursive function to fill in the empty cells of the puzzle one by one, checking for valid entries at each step.
• Allow users to input the unsolved puzzle and display the solved result.
Session 8: Python Object Oriented Programming: Class and Constructor
▪ Understanding the fundamental concepts of OOP, such as objects, classes, encapsulation, inheritance, and polymorphism.
▪ Differentiating between procedural programming and OOP.
▪ Defining classes in Python to create blueprints for objects.
▪ Introducing constructors as special methods in classes.
▪ Implementing a non-parameterized constructor using the __init__ method.
▪ Implementing a parameterized constructor to initialize attributes with user-provided values.
▪ Defining class attributes that are shared among all instances of a class.
▪ Understand and implement Class Methods and Static Methods
▪ Implementing inheritance by defining a subclass that inherits attributes and methods from a superclass.
▪ Defining method overriding to provide a different implementation of a method in a subclass.
▪ Understanding polymorphism as the ability of objects to take on multiple forms.
Exercises
Exercise 1: Class and Object Creation
▪ Create a class called **Student** that represents a student's information.
▪ Define attributes such as name, age, and grade for the **Student** class.
▪ Instantiate multiple objects of the **Student** class and set their attributes.
▪ Display the information of each student object.
Exercise 2: Non-parameterized Constructor

▪ Create a class called **Rectangle** that represents a rectangle.
▪ Define attributes for the length and width of the rectangle.
▪ Implement a non-parameterized constructor in the **Rectangle** class to initialize the attributes with default values.
▪ Instantiate a **Rectangle** object and display its dimensions.
Exercise 3: Parameterized Constructor
▪ Extend the **Rectangle** class with a parameterized constructor.
▪ Modify the parameterized constructor to accept length and width values as parameters and initialize the attributes accordingly.
▪ Instantiate a **Rectangle** object by providing length and width values through the constructor.
▪ Display the dimensions of the rectangle object.
Exercise 4: Bank Account Class
▪ Create a class called **BankAccount** that represents a bank account.
▪ Define attributes such as account number, account holder name, and balance.
▪ Implement a parameterized constructor to initialize the attributes.
▪ Include methods to deposit and withdraw funds from the bank account.
▪ Instantiate a **BankAccount** object and perform deposit and withdrawal operations.
▪ Display the updated balance after each transaction.
Exercise 5: Employee Class
▪ Create a class called **Employee** to represent an employee's information.
▪ Define attributes such as employee ID, name, and salary.
▪ Implement a parameterized constructor to initialize the attributes.
▪ Include a method to display the employee's details.
▪ Instantiate multiple **Employee** objects and display their information.
Exercise 6: Vehicle Class Hierarchy
▪ Create a base class called **Vehicle** with attributes such as make, model, and year.
▪ Define a parameterized constructor in the **Vehicle** class to initialize the attributes.
▪ Create derived classes like **Car**, **Motorcycle**, and **Truck** that inherit from the **Vehicle** class.
▪ Implement constructors in the derived classes to initialize additional attributes specific to each type of vehicle.
▪ Instantiate objects of each derived class and display their details.

*Students to do any of the following mini project
Project 8: Bank Account Management System:
• Create a bank account class with attributes such as account number, account holder name, and balance.
• Implement methods to deposit, withdraw, and display the account details.
• Use constructors to initialize the account attributes and handle different types of accounts (e.g., savings, checking).
Project 9: Student Management System:
• Develop a student class with attributes like student ID, name, and grades.
• Implement methods to calculate the average grade, display student information, and add new grades.
• Use constructors to initialize student attributes and handle student objects with different sets of grades.
Project 10: Library Management System:
• Create classes for books, authors, and library members.
• Implement methods to add books, search for books by author or title, and manage library member details.
• Use constructors to initialize book and member attributes and handle multiple book copies.
Project 11: Employee Payroll System:
• Develop an employee class with attributes like employee ID, name, salary, and designation.
• Implement methods to calculate monthly salary, display employee details, and handle promotions.
• Use constructors to initialize employee attributes and handle different employee designations.
Session 9: Python Object Oriented Programming: Inheritance, Methods Method Overriding and Data Abstraction, Encapsulation and Polymorphism.
▪ Understanding the fundamental concepts of OOP, such as objects, classes, encapsulation, inheritance, and polymorphism.
▪ Differentiating between procedural programming and OOP.
▪ Defining classes in Python to create blueprints for objects.
▪ Introducing constructors as special methods in classes.
▪ Implementing a non-parameterized constructor using the __init__ method.
▪ Implementing a parameterized constructor to initialize attributes with user-

provided values.
▪ Defining class attributes that are shared among all instances of a class.
▪ Understand and implement Class Methods and Static Methods
▪ Implementing inheritance by defining a subclass that inherits attributes and methods from a superclass.
▪ Defining method overriding to provide a different implementation of a method in a subclass.
▪ Understanding polymorphism as the ability of objects to take on multiple forms.
Exercises
Exercise 5: Class Attributes and Methods:
▪ Create a class that represents a car. Include class attributes such as the number of wheels and class methods to calculate the average speed of all cars created.
▪ Write a program to instantiate multiple car objects and access their class attributes and methods.
Exercise 5: Parameterized Constructors:
▪ Build a class called "Rectangle" that has attributes for length and width.
▪ Implement a parameterized constructor that takes the length and width as arguments and initializes the attributes.
▪ Write a program to create multiple rectangle objects with different dimensions and display their properties.
Exercise 5: Multi-Level Inheritance:
▪ Create a base class called "Animal" with attributes and methods common to all animals.
▪ Derive a subclass called "Mammal" from the "Animal" class with additional attributes and methods specific to mammals.
▪ Further derive a subclass called "Dog" from the "Mammal" class with specific attributes and methods for dogs.
▪ Write a program to instantiate a dog object, access its inherited attributes and methods, and invoke the dog-specific ones.
Exercise 5: Multiple Inheritance:
▪ Define a base class called "Shape" with attributes and methods related to shapes.
▪ Create two subclasses, "Circle" and "Rectangle," that inherit from the

"Shape" class.
▪ Implement another class called "Shape3D" that represents 3D shapes and inherits from both the "Circle" and "Rectangle" classes.
▪ Write a program to instantiate a 3D shape object, access attributes and methods from the multiple inherited classes, and display their properties.
Exercise 5: Method Overriding:
▪ Create a base class called "Vehicle" with a method called "start" that prints "Vehicle started."
▪ Derive a subclass called "Car" from the "Vehicle" class and override the "start" method to print "Car started."
▪ Derive another subclass called "Motorcycle" from the "Vehicle" class and override the "start" method to print "Motorcycle started."
▪ Write a program to instantiate a car and a motorcycle object and invoke their respective "start" methods.
Exercise 5: Polymorphism:
▪ Define a base class called "Shape" with a method called "area" that returns the area of a shape.
▪ Create subclasses, such as "Rectangle," "Circle," and "Triangle," that inherit from the "Shape" class and implement their own "area" methods.
▪ Write a program to instantiate objects of different shapes, invoke their "area" methods, and display the calculated areas.
Projects 12: Contact Management System with Inheritance:
• Build a contact management system that allows users to add, search, update, and delete contacts.
• Create a base class called "Contact" with common attributes and methods.
• Implement subclasses like "PersonContact" and "OrganizationContact" that inherit from the base class and add specific attributes and methods.
• Employee Management System with Class Methods:
•
• Develop an employee management system that stores and manages employee information.
• Use class methods to implement functionalities such as calculating average salary, finding the highest-paid employee, and displaying employee statistics.
Project 13: Banking System with Multiple Inheritance:
• Create a banking system that models different types of bank accounts, including savings accounts, checking accounts, and credit card accounts.
• Implement multiple inheritance by creating a base class called

"BankAccount" and subclasses for each specific account type.
• Use the appropriate attributes and methods from both the base class and the subclasses to handle account transactions and balance management.
Project 14: School Management System with Multi-Level Inheritance:
• Develop a school management system that handles student, teacher, and administrative tasks.
• Implement multi-level inheritance with a base class called "Person" and subclasses like "Student," "Teacher," and "Administrator" inheriting from it.
• Define specific attributes and methods for each subclass, such as calculating student grades, managing teacher schedules, and handling administrative tasks.
Session 10: Handling data in Python
• Text file handling
• What is Python Pandas?
• Pandas Series
• Basic Operations on Series
• Pandas DataFrame
• Basic Operations on DataFrames
• Extracting data from RDBMS using SQLite
Exercises:
• Read the data in different formats like csv,excel,text file etc
• Apply data cleaning activities
• Apply queries on data to fetch relevant data using various sample data sources
• Apply join operations on multiple data sources to extract data
• Extracting data from data in RDBMS using SQLite library
Project: https://github.com/jvns/pandas-cookbook
Session 11: Data Visualization with various python packages
Learn to create various types of visualization charts with following libraries
• Matplotlib
• Seaborn

- Bokeh
- Plotly

Exercises:

1.draw a line using given axis values with suitable label in the x axis , y axis and a title.

2.draw line charts of the financial data of Alphabet Inc. between October 3, 2016 to October 7, 2016.
Sample Financial data (fdata.csv):

```
Date,Open,High,Low,Close
10-03-16,774.25,776.065002,769.5,772.559998
10-04-16,776.030029,778.710022,772.890015,776.429993
10-05-16,779.309998,782.070007,775.650024,776.469971
10-06-16,779,780.47998,775.539978,776.859985
10-07-16,779.659973,779.659973,770.75,775.080017
```

3. display the grid and draw line charts of the closing value of Alphabet Inc. between October 3, 2016 to October 7, 2016. Customized the grid lines with linestyle -, width .5. and color blue.

```
Date,Close
03-10-16,772.559998
04-10-16,776.429993
05-10-16,776.469971
06-10-16,776.859985
07-10-16,775.080017
```

4. display the grid and draw line charts of the closing value of Alphabet Inc. between October 3, 2016 to October 7, 2016. Customized the grid lines with rendering with a larger grid (major grid) and a smaller grid (minor grid).Turn on the grid but turn off ticks

5. Write a Python program to create multiple plots.

Consider following data in csv format:

Exercise 1: Read Total profit of all months and show it using a line plot
Exercise 2: Get total profit of all months and show line plot with the following Style properties
Exercise 3: Read all product sales data and show it using a multiline plot
Exercise 4: Read toothpaste sales data of each month and show it using a scatter plot
Exercise 5: Read face cream and facewash product sales data and show it using the bar chart
Exercise 6: Read sales data of bathing soap of all months and show it using a bar chart. Save this plot to your hard disk
Exercise 7: Read the total profit of each month and show it using the histogram to see the most common profit ranges
Exercise 8: Calculate total sale data for last year for each product and show it using a Pie chart
Exercise 9: Read Bathing soap facewash of all months and display it using the Subplot
Exercise 10: Read all product sales data and show it using the stack plot
Project: https://github.com/CICIFLY/Data-Analytics-Projects

Self-Learning Components

Students are required to demonstrate their self-learning components by developing a mini project

List of Suggested projects for students

Project: Random password generator
Create a program that takes a number and generate a random password length of that number.
Topics: random module, joining strings, taking input
Hint: *Create a string with all characters, then take random characters from it and concatenate each char to make a big string.*

Project: Tic-Tac-Toe
We all have interesting memories of playing tic-tac-toe with our friends, don't we? This is the most fun and interesting game to play anywhere, all you need is a pen and paper.
create a 3x3 grid just like the traditional method. Then we will ask the user to put 'X' at any of the grid and respond accordingly by placing 'O' in the remaining places. Also, we will try to put 'O' in the program to create the vertical, horizontal, or diagonal lines as because whoever does that first will win the game and the message will be displayed.
Project: Number Guessing
The number-guessing game is a simple game where one player thinks of a number and the other player tries to guess the number. The player who is thinking of the number will provide hints to the other player, such as whether the guess is too high or too low, to help them narrow down the possibilities and eventually guess the correct number.
Project: Password Strength Checker
A password strength generator is a program that helps users create strong and secure passwords. **This type of program typically uses algorithms to evaluate the strength of a given password and provide feedback to the user on how to improve it.** The program may also be able to generate a strong password for the user automatically.
Strong passwords are important for protecting online accounts from hackers and other malicious actors. A good password should be at least 8 characters long and include a mix of uppercase and lowercase letters, numbers, and special characters. Using a password strength generator can help ensure that your password is strong and secure.
Project: Hangman Game
It is always fun to build a game using python programming. This project involves the same. Hangman Game or more like a "guess the word" game is the best program for good interactive learning.
Here, the user will have to guess an alphabet to complete a word, and also each user will have a limited number of chances to guess a letter. The programmer can create the preorganized list of words to be finished and then include a specific

function to check whether the user has guessed the correct letter if yes then include that letter to word for finishing the default word and if the guesswork is not true then reducing the count of chances to guess further.

While developing this python project you will learn the core concepts like variables, characters, strings, lists, conditional statements, loops, and functions. Also, the concept of the counter variable is used to limit the number of guesses.

Project: Password Strength Checker

A password strength generator is a program that helps users create strong and secure passwords. **This type of program typically uses algorithms to evaluate the strength of a given password and provide feedback to the user on how to improve it.** The program may also be able to generate a strong password for the user automatically.

Strong passwords are important for protecting online accounts from hackers and other malicious actors. A good password should be at least 8 characters long and include a mix of uppercase and lowercase letters, numbers, and special characters. Using a password strength generator can help ensure that your password is strong and secure.

Project: Mad Libs Generator

Mad Libs is a word game where players supply words to fill in blanks in a story or sentence, typically with funny or amusing results. A Mad Libs generator is a tool that can be used to automatically create Mad Libs stories. To use a Mad Libs generator, the user typically inputs various types of words, such as nouns, verbs, and adjectives, and the generator will use these words to fill in the blanks in a pre-written story or sentence. The resulting Mad Libs story will often be humorous and nonsensical.

Project: Snake Game

The snake game is a classic arcade game where the player controls a snake that moves around the screen, trying to eat food while avoiding obstacles and its own tail. The snake grows in length each time it eats food, making it more difficult to navigate and maneuver. The objective of the game is to score as many points as possible by eating as much food as possible without crashing into an obstacle or

the snake's own tail.
Project: Calculator Application:
• Build a calculator application that supports basic mathematical operations using functions.
• Implement error handling for invalid input and division by zero.
• Enhance the calculator by adding more advanced operations such as exponentiation and square root using math functions.
Project: Recursive File Search:
• Write a program that recursively searches a directory and its subdirectories for files with a specific extension.
• Implement the recursive file search using a function that calls itself.
• Allow the user to input the directory path and extension to search for.
Project: Recursive Drawing:
• Create a program that uses recursion to draw geometric patterns.
• Write functions to draw shapes such as squares, circles, and triangles, with varying sizes and positions.
• Implement recursion to generate more complex patterns by recursively calling the drawing functions.
Project: Text Adventure Game:
• Design and build a text-based adventure game using functions for different game actions and interactions.
• Implement recursion to allow for nested scenarios or choices within the game.
• Include multiple storylines, character interactions, and branching paths based on user decisions.
Project: Recursive Image Processing:
• Develop a program that applies recursive algorithms for image processing tasks, such as blurring or edge detection.
• Implement recursive functions to traverse the image pixels and modify their values based on neighboring pixels.
• Allow users to input an image file and display the processed result.
Project: Maze Solver:
• Build a program that solves a maze using recursion.
• Create a function that recursively explores the possible paths in the maze until the exit is found.
• Visualize the maze solving process by displaying the steps taken to navigate through the maze.

References

https://www.tutorjoes.in/python_programming_tutorial/

https://www.udemy.com/course/100-days-of-code/

https://favtutor.com/blog-details/7-Python-Projects-For-Beginners

https://github.com/NaviRocker/100-days-of-python

https://hackr.io/blog/python-projects

ENGINEERING DRAWING & WORKSHOP LAB

Department:	**Department of Mechanical Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Engineering Drawing and Workshop Lab	**SEC033**	0-0-4	2
Type of Course:	Programme Core		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	Statements
CO1	**Understand** the polygons, circles and lines with different geometric conditions
CO2	**Draw** the projection of points, lines and planes under different conditions and orthographic views from isometric views of simple objects
CO3	**Determine** manufacturing methods in different fields of engineering and Practical exposure to different fabrication techniques
CO4	**Creation** of simple components using different materials
CO5	**Exposure** to some of the advanced and latest manufacturing techniques being employed in the industry.

Ex. No	Experiment Title	Mapped CO/COs
	Engineering Drawing	
1	To study Drawing Instruments and their uses, BIS conventions, Lettering, Dimensioning line conventions and free hand practicing, AUTO CAD Commands.	CO1
2	Orthographic Projection: Introduction, Definitions - Planes of projection, reference line and conventions employed, Projections of points in all the four quadrants.	CO1
3	Study of Projections of straight lines True and apparent lengths, True and apparent inclinations to reference planes.	CO1
4	To study orthographic projection of plane surfaces.	CO2
5	To study orthographic projection of solids with examples.	CO2
6	To study sections and development of surfaces of solids with examples.	CO1
7	To study conversion of pictorial views into orthographic projections with examples.	CO1, CO2
8	To study isometric projections of solids with examples.	CO1, CO2
	Workshop:	
9	To make Different types of joints in carpentry shop.	CO, CO4
10	To make Double V-Butt and Lap joint in welding shop.	CO3, CO4
11	To prepare a Job on Lathe machines with step turning and chamfering operation.	CO3, CO4
12	To prepare a Job on Shaper/milling/grinding for finishing of a job.	CO3
13	To prepare a practice job in fitting shop.	CO3
14	To Study about the various machine tools	CO5
15	To make saw - cut filling V-cut taper at the corners, circular cut in fitting shop.	CO4

Semester: 2

LINEAR ALGEBRA AND ORDINARY DIFFERENTIAL EQUATIONS

<table>
<tr><td>Department:</td><td colspan="3">Department of Computer Science and Engineering</td></tr>
<tr><td rowspan="2">Course Name:

Linear Algebra and Ordinary Differential Equations</td><td>Course Code</td><td>L-T-P</td><td>Credits</td></tr>
<tr><td>ENMA102</td><td>3-1-0</td><td>4</td></tr>
<tr><td>Type of Course:</td><td colspan="3">Major</td></tr>
<tr><td colspan="4">Pre-requisite(s): Single variable calculus, Matrices, Differentiation and Integration</td></tr>
<tr><td colspan="4">Brief Syllabus:

The purpose of the course Linear Algebra and Ordinary Differential Equations is to provide students with a foundational understanding of key concepts and techniques in linear algebra and differential equations. The course helps students develop a strong mathematical foundation by studying fundamental topics in linear algebra and differential equations. These areas of mathematics are essential in various scientific, engineering, and mathematical disciplines. This course is to equip students with the mathematical tools, problem-solving skills, and foundational knowledge required to understand and apply linear algebra and differential equations in various academic and professional settings.</td></tr>
<tr><td colspan="4">UNIT WISE DETAILS</td></tr>
<tr><td>Unit Number: 1</td><td colspan="2">Title: Matrices and Systems of Linear Equations</td><td>No. of hours: 10</td></tr>
<tr><td colspan="4">Content Summary: Matrix with operation, Types of Matrix (Symmetric and skew symmetric matrix, Hermitian and skew Hermitian matrix, unitary and orthogonal matrix), Determinant of Matrix, Inverse and transpose of matrices, Elementary row operations, Systems of Linear Equations, Homogeneous and non-</td></tr>
</table>

homogeneous systems, Solutions of linear systems Gaussian, elimination and row echelon form, Rank of matrix.

Unit Number: 2	Title: Eigenvalues and Eigenvectors	No. of hours: 10

Content Summary: Definition and properties of eigenvalues and eigenvectors, Diagonalization of matrices, Eigenvalues and eigenvectors of symmetric, skew symmetrix, hermition, skew hermition, unitary and orthogonal matrices, Calyey Hamilton Theorem, Rank and nullity of a matrix, Diagonalization of matrices, Minimal polynomial, characteristic polynomial, and generalized eigenvectors. The Jordan Normal Form Theorem for linear operators on a finite dimensional complex vector space, Quadratic forms, Applications of eigenvalues and eigenvectors.

Unit Number: 3	Title: Vector Spaces	No. of hours: 10

Content Summary: Introduction to vector spaces**,** Subspaces and spanning sets**,** Linear independence and basis**,** Dimension of vector spaces**,** Row space, column space, and null space**,** Linear transformations**,** Matrix representation of linear transformations, Inner Product Spaces,Inner products and orthogonality, Orthonormal bases and Gram-Schmidt process, Orthogonal projections and least squares approximations, Applications of Linear Algebra, Markov chains and transition matrices.

Unit Number: 4	Title: Ordinary Differential Equations	No. of hours: 10

Content Summary: Introduction to ordinary differential equations, Definition and classification of differential equations, First-order linear differential equations, Separable differential equations, Exact differential equations, Integrating factors, Applications of first-order differential equations, Second-order linear differential equations**,** Homogeneous differential equations**,** Method of undetermined coefficients**,** Variation of parameters**,** Applications of second-order differential equations

***Self-Learning Components:**

https://archive.nptel.ac.in/courses/111/108/111108081/

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components
Reference Books: 1. Christian Constanda, *Differential Equations*. Second Edition, Springer 2017, ISBN-13: 978-3-319-50223-6. 2. We will use the book by Sheldon Axler, *Linear Algebra Done Right,* third edition, Springer Nature, ISBN 978-3-319-30765-7 3. Elementary Linear Algebra: Applications Version, Howard Anton and Chris Rorres, 11/E Wiley 4. Elementary Linear Algebra w/Applications (Classic Verson), Kolman & Hill, 9/E, Pearson 5. Linear Algebra and Its Applications, Lay, Lay and McDonald, 5/E, Pearson.

Define Course Outcomes (CO)

COs	Statements
CO 1	**Identify** and **analyze** the properties of various types of matrices, such as symmetric, skew-symmetric, Hermitian, skew Hermitian, unitary, and orthogonal matrices.
CO 2	**Analyze** quadratic forms and apply eigenvalues and eigenvectors in practical situations.
CO 3	**Define** and **comprehend** vector spaces, subspaces, linear independence, and basis.
CO 4	**Determine** the dimension of vector spaces and compute row space, column space, and null space of matrices.
CO 5	**Solve** first-order linear, separable, exact, and homogeneous differential equations.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels(C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1	-	P1
CO2	C2	-	P2
CO3	C2	-	P2
CO4	C3, C4	-	P3
CO5	C5	-	P4, P5

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	-	2	-	2	-	-	-	-	-	-	2
CO2	3	3	2	2	-	-	-	-	-	-	-	3
CO3	3	-	--	-	-	-	2	-	-	-	-	2
CO4	-	-	3	-	-	-	-	-		-	2	3
CO5	3	2	-	3	-	-	-	-	-	-	-	2

Justification for mapping must be relevant.

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	-	-	-	-
CO2	-	3	-	-	-
CO3	3	-	-	-	1
CO4	-	-	2	-	1
CO5	-	2	-	1	-

Relevance of the Syllabus to various indicators

Unit I	**Matrices and Systems of Linear Equations**
Local	Understanding matrices and systems of linear equations can aid in solving local engineering and scientific problems.
Regional	Understanding matrices and systems of linear equations can be applied to regional projects or research in engineering fields.
National	Matrices and systems of linear equations play a significant role in various national engineering and scientific endeavors.
Global	Matrices and systems of linear equations are fundamental to global scientific research, engineering projects, and innovations.
Employability	Knowledge of matrices and systems of linear equations enhances employability in various technical and engineering professions.
Entrepreneurship	Knowledge of matrices and systems of linear equations can inspire entrepreneurial opportunities in engineering-based ventures.
Skill Development	Learning matrices and systems of linear equations develops analytical and problem-solving skills for engineering

	applications.
Professional Ethics	Applying matrices and systems of linear equations with integrity ensures ethical practices in engineering and scientific fields.
Gender	Matrices and systems of linear equations education is equally important for individuals of all genders pursuing engineering.
Human Values	Applying matrices and systems of linear equations with ethical considerations contributes to human-centered engineering solutions.
Environment & Sustainability	-
Unit II	**Eigenvalues and Eigenvectors**
Local	Understanding eigenvalues and eigenvectors can aid in solving local engineering and scientific problems.
Regional	Understanding eigenvalues and eigenvectors can be applied to regional projects or research in engineering fields.
National	Eigenvalues and eigenvectors play a significant role in various national engineering and scientific endeavors.
Global	Eigenvalues and eigenvectors are fundamental to global scientific research, engineering projects, and innovations.
Employability	Knowledge of eigenvalues and eigenvectors enhances employability in various technical and engineering professions.
Entrepreneurship	Knowledge of eigenvalues and eigenvectors can inspire entrepreneurial opportunities in engineering-based ventures.
Skill Development	Learning eigenvalues and eigenvectors develops analytical and problem-solving skills for engineering applications.
Professional Ethics	Applying eigenvalues and eigenvectors with integrity ensures ethical practices in engineering and scientific fields.
Gender	Eigenvalues and eigenvectors education is equally important for individuals of all genders pursuing engineering.

Human Values	Applying eigenvalues and eigenvectors with ethical considerations contributes to human-centered engineering solutions.
Environment & Sustainability	-
Unit III	**Vector Spaces**
Local	Understanding vector spaces can aid in solving local engineering and scientific problems.
Regional	Understanding vector spaces can be applied to regional projects or research in engineering fields.
National	Vector spaces play a significant role in various national engineering and scientific endeavors.
Global	Vector spaces are fundamental to global scientific research, engineering projects, and innovations.
Employability	Knowledge of vector spaces enhances employability in various technical and engineering professions.
Entrepreneurship	Knowledge of vector spaces can inspire entrepreneurial opportunities in engineering-based ventures.
Skill Development	Learning vector spaces develops analytical and problem-solving skills for engineering applications.
Professional Ethics	Applying vector spaces with integrity ensures ethical practices in engineering and scientific fields.
Gender	Vector spaces education is equally important for individuals of all genders pursuing engineering.
Human Values	Applying vector spaces with ethical considerations contributes to human-centered engineering solutions.
Environment & Sustainability	-
Unit IV	**Ordinary Differential Equations**
Local	Understanding ordinary differential equations can aid in

	solving local engineering and scientific problems.
Regional	Understanding ordinary differential equations can be applied to regional projects or research in engineering fields.
National	Ordinary differential equations play a significant role in various national engineering and scientific endeavors.
Global	Ordinary differential equations are fundamental to global scientific research, engineering projects, and innovations.
Employability	Knowledge of ordinary differential equations enhances employability in various technical and engineering professions.
Entrepreneurship	Knowledge of ordinary differential equations can inspire entrepreneurial opportunities in engineering-based ventures.
Skill Development	Learning ordinary differential equations develops analytical and problem-solving skills for engineering applications.
Professional Ethics	Applying ordinary differential equations with integrity ensures ethical practices in engineering and scientific fields.
Gender	Ordinary differential equations education is equally important for individuals of all genders pursuing engineering.
Human Values	Applying ordinary differential equations with ethical considerations contributes to human-centered engineering solutions.
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	-

ENGINEERING CHEMISTRY

Department:	**Department of Computer Science and Engineering**		
Course Name: **ENGINEERING CHEMISTRY**	**Course Code**	**L-T-P**	**Credits**
	ENCH101	3-1-0	4
Type of Course:	Major		
Pre-requisite(s), if any: Nil			
Brief Syllabus: Engineering Chemistry is a course that aims to provide engineering students with a foundational understanding of various chemical principles and their practical applications in engineering.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Water technology**	**No. of hours: 10**	
Content Summary: Introduction, water analysis: Hardness-determination by EDTA method-, Alkalinity-determination by double indicator method, Treatment of boiler feed water: Internal treatment (Phosphate,Colloidal and Calgon conditioning). External treatments: Ion exchange and lime-soda process, Zeolite processes. Determination of dissolved oxygen by Winkler's method and Determination of chemical oxygen demand, Boiler scales formation and ill effects, methods of prevention of scales. Numerical problems.			
Unit Number: 2	**Title: Chemical Fuels**	**No. of hours: 10**	
Content Summary: **Fuels**: Introduction, classification, calorific value (HCV & LCV), Determination of calorific value of fuel using Bomb calorimeter. **Solid fuel**: Coal- its analysis by proximate and ultimate analysis, Numerical			

problems.

Liquid fuels: Refining of petroleum, Petroleum cracking, Reformation of petrol-explanation with reactions, Knocking in IC engine, its ill effects and prevention of knocking. Anti-knocking agent: Leaded and unleaded petrol. Power alcohol and its advantages. Synthetic petrol - Bergius process.

Gaseous fuels: LPG, CNG and their applications.

Unit Number: 3	Title: Battery Technology	No. of hours: 10

Content Summary:

Introduction - Galvanic cell, electrode potential, EMF of the cell and cell representation. Batteries and their importance, Classification of batteries- primary, secondary and reserve batteries with examples. Battery characteristics - voltage, capacity, energy density, power density, energy efficiency, cycle life and shelf life.

Basic requirements for commercial batteries. Construction, working and applications of: Ni-Cd, and Lithium-ion battery.

Fuel Cells- Differences between battery and a fuel cell, Classification of fuel cells - based on type of fuel, electrolyte and temperature.

Unit Number: 4	Title: Polymer	No. of hours: 10

Content Summary:

Basic concepts of polymer, Types of polymers, Thermoplastic & thermosetting plastics, Preparation and application of some industrially important polymers (Natural rubber, Buna S, Buna-N, Neoprene, Isoprene, Nylon-6, nylon-6,6, Decron and Terylene). Conducting and biodegradable polymers.

***Self-Learning Components:**

Basics of electrochemistry:

https://mrcet.com/downloads/digital_notes/HS/4%20ENGINEERING%20CHEMISTRY.pdf

Basics of polymer:

https://gnindia.dronacharya.info/APS/Downloads/SubjectInformation/Chemistry/Unit2/Lecture_1_13022019.pdf

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components

Reference Books:

1. Principles of Physical Chemistry by B. R. Puri, L. R. Sharma and M. S. Pathania, S. Nagin Chand and Co.

2. Text book of Physical Chemistry by Soni and Dharmatha, S. Chand & Sons.

3. Text book of Polymers science by Gowarikar and Vishwanathan.

4. Corrosion Engineering by M. G. Fontana, Mc Graw Hill Publications.

5. Engineering Chemistry by Jain and Jain.

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** the methods for water hardness and alkalinity testing, and the basics of boiler water treatment.
CO2	**Explain** the process of dissolved oxygen determination and chemical oxygen demand analysis.
CO3	**Determine** various methods to enhance the quantity & quality of Fuel.
CO4	**Identify** between hard and soft water, solve the related numerical problems on water purification and its significance in industry and daily life.
CO5	**Articulate** basic concepts of chemistry in daily life.
CO6	**Design** efficient process for water analysis and purification

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	-	P1
CO2	C3	-	P5
CO3	C2	-	-
CO4	C3	-	P3
CO5	C6	-	P4
CO6	C6	-	P4

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	-	-	-	-	-	-	-	-	-	-	3
CO2	-	-	3	-	2	-	-	-	-	-	-	2
CO3	-	3	-	-	-	-	-	-	-	-	-	2
CO4	-	3	-	-	-	-	2	-	-	-	-	3
CO5	3	-	-	-	-	-	-	-	-	-	-	2

CO6	3	3	-	-	-	-	-	-	-	-	-	3

Justification for mapping must be relevant.

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4
CO1	3	-	-	-
CO2	2	-	-	3
CO3	2	-	-	-
CO4	2	-	2	-
CO5	-	1	-	-
CO6	2	-	3	3

Relevance of the Syllabus to various indicators

Unit I	**Water technology**
Local	Address the evaluation of water characteristics like dissolved oxygen, hardness, alkalinity for specific water bodies such as lakes, rivers, and groundwater sources.
Regional	Address the evaluation of water characteristics like dissolved oxygen, hardness, alkalinity for specific water bodies within a particular geographic area
National	Addresses national problem of water pollution and its impact on society at the national level by analyzing water quality data from various regions across the country.
Global	Addresses transboundary water issues and global water challenges.
Employability	Professionals in the field of water management, environmental science, and sustainability
Entrepreneurship	-

Skill Development	Develops basic knowledge of characteristics of water and methods to remove the hardness
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	To assess and address water-related challenges at various levels for a sustainable environment
Unit II	**Chemical Fuels**
Local	Addresses the immediate consequences of fuel-related pollution and environmental degradation in local region
Regional	-
National	Exploration, extraction, and management of fossil fuel resources, as well as the development and deployment of non-renewable and renewable energy sources,
Global	Aligns with global trends for exploration, extraction, and management of fossil fuel resources, as well as the development and deployment of non-renewable and renewable energy sources,
Employability	Develops skills to use semiconductor photochemistry in energy harnessing and optical sensors
Entrepreneurship	-
Skill Development	Develops basic knowledge about the characteristics of good fuels
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	Assess environmental impact, emissions, and contribution of fuels to climate change to encourage the adoption of renewable and sustainable alternatives.

Unit III	**Battery Technology**
Local	To address the application of fuel cells for power buildings, homes, and small-scale distributed energy systems at local level.
Regional	-
National	To address the applications of battery technology include powering electric vehicles (EVs), storing renewable energy generated from solar panels or wind turbines, and backup power for homes and businesses at national level.
Global	To address the international efforts to reduce greenhouse gas emissions and address climate change challenges.
Employability	Develops skills to fabricate fuel cell-related technologies and their applications
Entrepreneurship	-
Skill Development	Develops knowledge and skills in fuel cell tecchnologies
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Polymer**
Local	Addresses local application of polymer such as for packaging materials, household products, textiles, and construction materials
Regional	-
National	Contributes to synthesis and application of polymer at national level
Global	Aligns with global trends in to synthesis and application of polymer

Employability	Develops skills in synthesis of polymer and its applications
Entrepreneurship	-
Skill Development	Develops knowledge and skills in synthesis and application of polymer
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	

ENGINEERING CHEMISTRY LAB

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
ENGINEERING CHEMISTRY LAB	**ENCH151**	0-0-2	1
Type of Course:	Major		

Proposed Lab Experiments

Defined Course Outcomes

CO1	Students will learn and **apply** various experimental techniques commonly used in chemistry labs, such as titrations, distillations, extractions, chromatography, spectroscopy, and electrochemical methods.
CO2	Students will **acquire** proficiency in handling and operating laboratory equipment, including but not limited to balances, pipettes, burettes, spectrophotometers, pH meters, and other analytical instruments.
CO3	Students will **develop** skills in recording and analysing experimental data, including data interpretation of results.
CO4	Students will gain hands-on experience in synthesizing various chemical compounds and organic polymers
CO5	Students will learn to write **concise** and accurate laboratory reports, including experimental procedures, observations, results, and conclusions.
CO6	Students will **understand** the ethical responsibilities and laboratory safety protocols associated with conducting experiments.

Ex. No	**Experiment Title**	**Mapped CO/COs**

1	Determination of temporary and permanent hardness in water sample using EDTA.	CO1, CO3, CO5
2	Determination of alkalinity in the given water sample.	CO1, CO3, CO5
3	Determination of viscosity of given liquid.	CO2, CO3, CO5
4	Determination of surface tension of given liquid.	CO2, CO3, CO5
5	Determination of pH by pH-metric titration.	CO1, CO3, CO5
6	Preparation of Phenol-formaldehyde and Urea-formaldehyde resin	CO4, CO5, CO6
7	To determine the iron concentration in the given water sample by Spectrophotometer using potassium thiocyanate as colour developing agent.	CO1, CO3, CO5
8	Determination of chloride content in water sample.	CO1, CO3 CO5, CO6
9	Estimation dissolved oxygen (DO) content in the given water sample by Winkler's method.	CO1, CO3,CO5
10	Determination of iron content in the given solution by Mohr's method.	CO1, CO3, CO5
11	Determination of rate constant of hydrolysis of esters.	CO3, CO5
12	To determine the Iron content in the given salt by using external indicator	CO1, CO3, CO5
13	Determination of wavelength of absorption maximum and colorimetric estimation of Fe3+ in solution	CO2, CO3, CO5
14	Determination of molar absorptivity of a compound (KMnO4 or any water-soluble food colorant).	CO2, CO3, CO5
15	Preparation of a nickel complex [Ni(NH3)6]Cl2 and estimation of nickel by complexometric titration.	CO4, CO5, CO6

16	Synthesis of drug like Aspirin, /Paracetamol etc.	CO4, CO5, CO6

BASICS OF ELECTRICAL & ELECTRONICS ENGINEERING

Department:	**Department of Electrical & Electronics Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Basics of Electrical & Electronics Engineering	**ENEE101**	4-0-0	4
Type of Course:	Major		

Pre-requisite(s), if any: NA

Brief Syllabus:

The subject deals with the study and application of basic electrical and electronics devices in daily life and technology integration level. It encompasses various principles and concepts related to electronic devices and systems. Analysis of circuits using theorems for both AC and DC types. Also, it includes familiarization with CRO, function generator and power supplies, electrical measuring meters and different components. Lastly the concept of Integrated circuit and role of semiconductors in major industrial segments are conceptualized with examples.

UNIT WISE DETAILS

Unit Number: 1	**Title: Circuit Analysis:**	**No. of hours: 10**

Content Summary:

Ohm's Law, KCL, KVL Mesh and Nodal Analysis, Circuit parameters, energy storage aspects, Superposition, Thevenin's, Norton's, Reciprocity, Maximum Power Transfer Theorem, Millman's Theorem, Star-Delta Transformation. Application of theorem to the Analysis of D.C. circuits.

Unit Number: 2	**Title: A.C. Circuits & CRO**	**No. of hours: 10**

Content Summary:

A.C. Circuits: R-L, R-C, R-L-C circuits (series and parallel), Time Constant,

Phasor representation, Response of R-L, R-C and R-L-C circuit to sinusoidal input Resonance-series and parallel R-L-C Circuits, Q-factor, Bandwidth.

Cathode Ray Oscilloscope: Basic CRO circuit (Block Diagram), Cathode ray tube (CRT) & its component

Unit Number: 3	**Title: Semiconductor Physics**	**No. of hours: 10**

Content Summary:

Semiconductor Physics: Basic concepts, Intrinsic and extrinsic semiconductors, diffusion and drift currents. P-N junction diode: Ideal diode, P-N junction under open-circuit and closed-circuit, Diode Current Equation, Diode Resistance, Transition and Diffusion Capacitance, Effect of Temperature, Carrier Life Time, Continuity Equation. Special Diodes: Zener Diode, Photodiode, Light Emitting Diodes, applications of Diodes.

Unit Number: 4	**Title: Digital Electronics**	**No. of hours: 10**

Content Summary:

Digital Electronics: Boolean algebra, Truth tables of logic gates (AND, OR, NOT), NAND, NOR as universal gates. Bipolar junction transistor: Introduction to transistors: construction, transistor operations, BJT characteristics, load line, operating point, leakage currents. Application of BJT: CB, CE configurations, Introduction to FETs and MOSFETs.

Self-Learning Components:

Students will be capable of recognizing the small electronics components utilized in their daily households. Also, better classification practices can be developed in students over the types of devices working on AC or DC current. Even inbuilt structures of household devices can also be illustrated all along with studies

Mini Project: Small circuits can be built up using components identified initially and afterword's testing using the mustimeter and CRO can be carried out to justify the working feasibility of components studied in the syllabus.

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning

components

Reference Books:

1. D.P. Kothari & I J Nagrath, Basic Electrical Engineering, Tata McGraw Hill , New Delhi.

2. B L Thareja – A text book of Electrical Technology

3. Boylestad & Nashelsky, "Electronic Devices & Circuits", Pearson Education, 10th Edition.

4. V. K. Mehta & Rohit Mehta, "Principles of Electronics", S. Chand Publishers, 27th Edition.

Web References:

1. https://onlinecourses.nptel.ac.in/noc22_ee113/preview
2. https://www.edx.org/learn/electrical-engineering
3. https://www.classcentral.com/course/youtube-electrical-engineering-basics-54532
4. https://www.electronics-tutorials.ws/

Define Course Outcomes (CO)

COs	Statements
CO 1	**Describe** and learn the basic Knowledge of AC and DC Circuits in making real-time projects to solve engineering difficulties of real time.
CO 2	**Understands its** utilization in day-to-day work dealing with pure AC circuits. R L and basic C circuits effect over-voltage and ac wave will be studied.
CO 3	**Compare** and **Contrast** various logic gates. Demonstrate the ability to identify series, parallel complex circuits. Utilization of the preliminary knowledge gained to obtain real existing power-related problems.
CO 4	**Differentiate** about the understanding of semiconductor devices application to existing apparatuses
CO 5	**Distinguish** the devices used in multipurpose electronics devices.

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	-	-	-	-	-	2	-	1	-	-	-	-
CO2	-	1	1	-	-	-	1	-	1	1	1	-
CO3	2	2	-	2	2	3	2	-	-	-	-	-
CO4	-	-	-	2	2	-	-	2	1	2	2	-
CO5	3	-	-	-	-	-	-	2	-	3	3	-

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	-	-	1	-	1
CO2	-	2	2	-	-
CO3	1	-	2	-	-
CO4	1	-	3	-	1
CO5	2	-	-	-	1

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels (C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1	-	-
C02	C2	-	-
CO3	C3	A4	-
CO4	C6	-	P5
CO5	-	-	-

Relevance of the Syllabus to various indicators

Unit I	**Circuit Analysis:**
Local	In the local context, circuit analysis plays a role in electrical infrastructure, power distribution, and ensuring safe electrical connections within a community. It is crucial for local electricians, engineers, and technicians to understand circuit analysis principles to maintain and troubleshoot electrical systems efficiently.
Regional	In the regional context, circuit analysis is applied to larger electrical networks and power grids that serve multiple cities or areas. Engineers and policymakers use circuit analysis to design and optimize the regional power distribution, taking into account factors like load balancing, transmission losses, and system reliability.
National	At the national level, circuit analysis becomes even more critical for the stability and efficiency of the entire electrical grid. National power grids require careful planning, monitoring, and maintenance, making circuit analysis indispensable for energy security and

	sustainability.
Global	In the global context, circuit analysis is essential for understanding international power interconnections and the challenges of exchanging electricity across borders. Engineers and researchers collaborate to create harmonized standards and protocols for global power transmission.
Employability	Circuit analysis skills are highly sought after in various industries, especially in electrical engineering, electronics, telecommunications, and renewable energy. Proficiency in circuit analysis enhances one's employability, enabling individuals to contribute to the development and maintenance of cutting-edge technologies.
Entrepreneurship	For entrepreneurs in the electronics or renewable energy sector, a solid understanding of circuit analysis is essential. This knowledge allows them to design innovative products, optimize energy-efficient systems, and develop sustainable solutions to real-world problems.
Skill Development	Learning circuit analysis helps individuals develop critical analytical and problem-solving skills. It fosters logical thinking and the ability to approach complex challenges systematically, which is valuable not only in the field of engineering but also in various other domains.
Professional Ethics	Circuit analysis professionals must adhere to ethical principles while designing, implementing, and maintaining electrical systems. Ethical considerations include ensuring safety standards, complying with regulations, and prioritizing the well-being of end-users.
Gender	Encouraging gender diversity and inclusion in circuit analysis and electrical engineering is essential for creating a more balanced and innovative workforce. Efforts to promote equality and eliminate gender biases in STEM fields, including circuit analysis, are crucial for societal progress.
Human Values	Circuit analysis contributes to human values by improving the quality of life through efficient power distribution,

	renewable energy solutions, and advancements in electronic devices that enhance communication and connectivity.
Environment & Sustainability	Understanding circuit analysis is vital for designing energy-efficient systems and integrating renewable energy sources into the power grid. By optimizing energy consumption and promoting sustainable practices, circuit analysis can contribute to environmental preservation.
Unit II	**A.C. Circuits & CRO**
Local	AC Circuits: In the local context, AC circuits are used in various electrical devices and systems found in homes, offices, and local infrastructure. They power lights, appliances, and other electronics, ensuring daily convenience and functionality. CRO: Local technicians and engineers may use CROs to diagnose and troubleshoot electronic equipment and electrical systems in the community, ensuring their proper functioning.
Regional	• AC Circuits: At the regional level, AC circuits are utilized in larger-scale electrical installations, such as power distribution networks, substation systems, and regional electrical infrastructure, ensuring efficient and reliable electricity supply. • CRO: Regional engineering teams might use CROs to analyze complex waveforms in power distribution systems, helping to identify irregularities and optimize electrical performance.
National	• AC Circuits: Nationally, AC circuits play a crucial role in the power generation, transmission, and distribution systems across the country, contributing to the stability and security of the national power grid. • CRO: National laboratories and research institutions may use high-end CROs for advanced studies in electronics, communications, and signal processing, contributing to technological advancements on a national scale.

Global	• AC Circuits: At the global level, AC circuits connect countries through intercontinental power transmission lines, enabling the exchange of electricity across borders and promoting international collaboration in energy distribution. • CRO: Cutting-edge research and development in CRO technology may have global impacts on various fields, such as telecommunications, electronics, and high-tech industries.
Employability	• AC Circuits: Knowledge of AC circuits is essential for electrical engineers, technicians, and electricians in various industries. Employers seek professionals who can design, analyze, and troubleshoot AC circuits efficiently. • CRO: Understanding CRO operation and waveform analysis is valuable for professionals working in electronics, telecommunications, and research fields, making them more employable.
Entrepreneurship	• AC Circuits: Entrepreneurs may explore opportunities in renewable energy solutions, energy-efficient products, or smart grid technologies, which heavily rely on AC circuits for efficient power distribution. • CRO: Entrepreneurs with innovative ideas in electronic instrumentation or specialized CRO applications may start their ventures to cater to niche markets.
Skill Development	• AC Circuits: Learning about AC circuits fosters expertise in electrical engineering, enhancing analytical and problem-solving skills for professionals in the field. • CRO: Skill development in CRO usage enables engineers and researchers to gain insights from complex waveforms, facilitating advanced studies and product development.
Professional Ethics	• AC Circuits: Practicing electrical engineers must adhere to professional ethics to ensure the safety, reliability, and compliance of AC circuit

	installations. • CRO: Ethical considerations in CRO use involve respecting intellectual property rights, conducting honest research, and handling sensitive data responsibly.
Gender	• AC Circuits: Efforts to promote gender diversity and inclusion in electrical engineering professions, including AC circuit design and analysis, contribute to a more balanced and diverse workforce. • CRO: Encouraging women's participation in STEM fields, including electronics and instrumentation (such as CRO technology), helps bridge the gender gap in the industry.
Human Values	• AC Circuits: AC circuits improve human life by providing electricity for everyday needs, enhancing living standards, and facilitating technological advancements that benefit society. • CRO: CRO technology supports various scientific and engineering endeavors that contribute to human knowledge, healthcare, and technological progress.
Environment & Sustainability	• AC Circuits: The efficient design and implementation of AC circuits in renewable energy systems contribute to environmental sustainability by reducing reliance on fossil fuels. • CRO: CROs aid in the development of energy-efficient electronic devices, which align with sustainability goals and eco-friendly practices.
Unit III	Semiconductor Physics
Local	Semiconductor Physics: In the local context, semiconductor physics may be relevant to industries and businesses that use electronic devices and components. Local electronics manufacturers and technology companies may utilize semiconductor physics to design, produce, and troubleshoot electronic devices commonly used in the community.

Regional	Semiconductor Physics: At the regional level, research and development centers and academic institutions may delve into semiconductor physics to develop advanced materials, devices, and technologies. Collaborations among regional industries and research institutions can lead to regional technological advancements.
National	• Semiconductor Physics: Nationally, semiconductor physics plays a vital role in the semiconductor industry, which is critical for various sectors, including electronics, telecommunications, and renewable energy. National investments in semiconductor research can lead to technological breakthroughs and economic growth.
Global	• Semiconductor Physics: Globally, semiconductor physics drives the semiconductor industry, which has significant impacts on various aspects of modern life. It is a foundation for global technological innovation and the development of cutting-edge electronic devices •
Employability	• Semiconductor Physics: Understanding semiconductor physics is highly sought after in industries related to electronics, semiconductor manufacturing, telecommunications, and integrated circuit design. Professionals with expertise in semiconductor physics are in demand for various research and development roles.
Entrepreneurship	• Semiconductor Physics: Entrepreneurs can explore opportunities in the semiconductor industry by starting companies that develop novel semiconductor materials, manufacturing processes, or semiconductor-based devices. •
Skill Development	• Semiconductor Physics: Skill development in semiconductor physics is crucial for researchers, engineers, and scientists working in the semiconductor and electronics fields. It involves understanding semiconductor properties, band theory, carrier dynamics, and quantum mechanics.
Professional Ethics	• Semiconductor Physics: Professionals working in semiconductor research and development must adhere to ethical principles concerning safety, environmental protection, and responsible use of

	semiconductor technology.
Gender	• Semiconductor Physics: Encouraging gender diversity and inclusion in semiconductor physics and related fields is essential for creating a more balanced and diverse workforce. Efforts to promote gender equity in STEM fields help increase representation and opportunities for women.
Human Values	• Semiconductor Physics: The advancements in semiconductor technology have improved human life through various electronic devices, communication systems, medical equipment, and energy-efficient solutions.
Environment & Sustainability	• Semiconductor Physics: Research in semiconductor physics contributes to the development of energy-efficient electronic devices, renewable energy technologies, and environmental monitoring systems, which align with sustainability goals.
Unit IV	**Digital Electronics**
Local	• Digital Electronics: In the local context, digital electronics is used in everyday electronic devices found in homes and businesses. Local electronics retailers and repair services utilize digital electronics knowledge to provide and maintain electronic products in the community. •
Regional	• Digital Electronics: At the regional level, digital electronics plays a role in the development of regional technology hubs and manufacturing centers. Collaboration among regional industries and research institutions may lead to advancements in digital electronics applications. •
National	• Digital Electronics: Nationally, digital electronics is vital for the electronics industry, contributing to the design and production of integrated circuits, microprocessors, and electronic systems. National investments in research and development can lead to technological innovations and economic growth. •

Global	• Digital Electronics: Globally, digital electronics is at the heart of the information age, enabling global communication, computing, and connectivity. Global collaborations and standardizations shape the advancement and implementation of digital technologies worldwide. •
Employability	• Digital Electronics: Understanding digital electronics is essential for professionals in the electronics, computer hardware, and telecommunications industries. Skills in digital logic design, microprocessor programming, and digital system troubleshooting are highly sought after.
Entrepreneurship	• Digital Electronics: Entrepreneurs can explore opportunities in digital electronics by starting companies that develop innovative digital products, embedded systems, or IoT devices. •
Skill Development	• Digital Electronics: Skill development in digital electronics involves learning about binary systems, logic gates, digital circuit design, FPGA programming, and microcontroller applications. These skills are valuable for engineers and technologists in various industries. •
Professional Ethics	• Digital Electronics: Professionals working with digital electronics must adhere to ethical principles concerning data privacy, cybersecurity, and responsible use of digital technology. •
Gender	• Digital Electronics: Efforts to promote gender diversity and inclusion in digital electronics and related fields aim to bridge the gender gap in STEM professions, creating more opportunities for women in technology. •
Human Values	• Digital Electronics: Digital electronics contributes to human values by enabling efficient communication, improved healthcare through medical devices, and enhanced quality of life through various electronic innovations. •
Environment &	• Digital Electronics: Advancements in digital

Sustainability	electronics have led to energy-efficient computing and power management systems, contributing to environmental sustainability by reducing energy consumption and electronic waste.
SDG	SDG 4, SDG 8
NEP 2020	-
POE/4th IR	Aligns with concept of making energy efficient devices and sensor building approaches

BASICS OF ELECTRICAL & ELECTRONICS ENGINEERING LAB

Department:	Department of Electrical & Electronics Engineering		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Basics of Electrical & Electronics Lab	**ENEE151**	0-0-2	1
Type of Course:	Major Course		

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	To **learn** using basic instruments for measuring real time values.
CO 2	To **implement** learned theorems in separate Dc and AC Circuits
CO 3	Capability to **perform** small circuit-based experiments using digital logics
CO 4	Desire to **obtain** wave forms of various device to analyse its actual functioning.

Ex. No.	Experiment Title	Mapped CO/COs
1	To get familiar with the working knowledge of the following instruments: a) Cathode ray oscilloscope (CRO) b) Multimeter (Analog and Digital) c) Function generator d) Power supply	CO4, CO1
2	To measure phase difference between two waveforms using CRO To measure an unknown frequency from Lissajous figures using CRO	CO4, CO1
3	To Verify the Thevenin' s and Norton’s theorem	CO2
4	To Verify the Superposition theorem	CO2

5	To measure voltage, current and power in an A.C. circuit by LCR impedance method	CO3
6	To study the frequency response curve in series and parallel-L-C circuit	CO3
7	a) Plot the forward and reverse V-I characteristics of P-N junction diode b) Calculation of cut-in voltage B.Tech. c) Study of Zener diode in breakdown region	CO3
8	To plot and study the input and output characteristics of BJT in common-emitter configuration.	CO2
9	Verification of truth tables of logic gates (OR, AND, NOT, NAND, NOR).	CO1, CO4
10	To get familiar with the working and use of seven-segment display.	CO4

FRONT END DEVELOPMENT WITH REACTJS

Department:	**Department of Electrical & Electronics Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Front End Development with ReactJS	**ENSP108**	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any: NA			
Brief Syllabus: The course covers the essentials of React programming with details on fundamental architecture of a react application. The learning journey showcases the discovery of how a Single Page React application differs from traditional web development frameworks. The learners will be able to use the new ES6 language features to write code. Create a React application from the ground up.			
UNIT WISE DETAILS			
Unit Number: 1	**Title:** ES6	**No. of hours: 8**	
Content Summary: **ES6:** Specifications and Features, Introduction, The let and const, The arrow functions, New Literal Syntax, Classes, Inheritance using extends, Default Parameter Values, Spread Operator, Iterators and Generators, Introduction to React, Features of React, Why we Need React			
Unit Number: 2	**Title: ECMA, E6**	**No. of hours: 6**	
Content Summary: ECMA Script, ES6 let and const, the arrow functions, New Literal Syntax, Classes, Inheritance using extends, Default Parameter Values, Spread Operator (...), Iterators and Generators, Features of React, Practical Application, Why need React, How React Works,			

Leveraging Virtual DOM, Setting up React

Unit Number: 3	Title: JSX	No. of hours: 8

Content Summary:

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

Unit Number: 4	Title: Elements	No. of hours: 6

Content Summary:

Rendering Element, about render (), Creating React Element, Updating Element, components, Introducing Components, Types of Components, Functional Component, Functional Components as Stateless, Using Functional Component.

Unit Number: 5	Title: Redux	No. of hours: 8

Content Summary:

Redux: Redux Concepts, Redux Principles, Data Flow, Actions, Functions, Reduces, Testing , DevTools, React & Redux Integrate

Self-Learning Components:

- JSX syntax and components
- Styling in React (CSS-in-JS libraries, styled-components)

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components

Reference Books:

1. Anthony Accomazzo, "Fullstack React: The Complete Guide to React JS and Friends" , fullstack.io
2. Robin Wieruch, "The Road to Learn React: Your journey to master plain yet

pragmatic React.js", Independent Publisher
3. Adam Boduch and Roy Derks, "React and React Native", Packt
4. Andrea Chiarelli, "Beginning React", Packt

Define Course Outcomes

COs	
CO 1	Identify the purpose and usage of React and its key features.
CO 2	Implement form handling and controlled components in a React application.
CO 3	Examine the structure and organization of a Redux store and its impact on an application's architecture.
CO 4	Discuss the effectiveness of different testing strategies (unit testing, integration testing) for React applications.

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	1	-	-	-	-	2	-	1	-	-	-	2
CO2	-	1	1	-	-	-	1	-	1	1	1	2
CO3	2	1	-	1	2	2	-	-	-	-	-	2
CO4	-	1	-	2	3	-	-	2	1	3	2	2

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	1	-	1	-	2
CO2	-	2	2	-	2
CO3	1	2	-	1	-
CO4	2	-	1	-	1

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels (C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1	A1	P1
C02	C3	-	-
CO3	C4	A4	-
CO4	C6	A5	-

Relevance of the Syllabus to various indicators

Unit I	**ES6**
Local	-
Regional	-

National	-
Global	Proficiency in modern JavaScript syntax is important for collaborating on international open-source projects and industry standards.
Employability	Mastering the skills outlined in the syllabus enhances employability as modern web development skills are in high demand.
Entrepreneurship	-
Skill Development	The syllabus focuses on developing technical skills in web development and React, which are highly relevant in the technology industry.
Professional Ethics	-
Gender	-
Human Values	The skills gained from the syllabus can be applied to projects that align with human values, such as creating accessible and inclusive web applications.
Environment & Sustainability	-
Unit II	**ECMA, E6**
Local	-
Regional	-
National	-
Global	-
Employability	Proficiency in modern JavaScript and React significantly enhances employability prospects, as these skills are in high demand in the job market.
Entrepreneurship	Knowledge of modern web development and React empowers entrepreneurs to create their own digital products or services.

Skill Development	The syllabus focuses on developing technical skills in modern web development practices and a popular JavaScript library, fostering skill enhancement.
Professional Ethics	Applying best practices and efficient coding techniques aligns with professional ethics, ensuring code quality and maintainability.
Gender	-
Human Values	Proficiency in React and web development allows for the creation of user-friendly and inclusive web applications, adhering to human-centered design principles.
Environment & Sustainability	-
Unit III	**JSX**
Local	-
Regional	-
National	-
Global	-
Employability	Proficiency in JSX and React topics significantly enhances employability prospects, as these skills are highly sought after in the job market.
Entrepreneurship	Knowledge of JSX and React empowers entrepreneurs to create user-centric and engaging digital products or services.
Skill Development	The syllabus focuses on developing practical and technical skills in modern web development, catering to skill enhancement.
Professional Ethics	Applying best practices, clean coding, and maintainable code align with professional ethics, ensuring quality software development.
Gender	-
Human Values	-
Environment &	-

Sustainability	
Unit IV	**Elements**
Local	-
Regional	-
National	-
Global	Proficiency in rendering elements is a standard practice across the global React development community.
Employability	-
Entrepreneurship	Knowledge of creating visually appealing and well-rendered user interfaces empowers entrepreneurs to develop their own digital products or services.
Skill Development	The syllabus focuses on developing technical skills in web development, fostering skill enhancement and industry readiness.
Professional Ethics	Applying best practices in rendering, updating, and component development aligns with professional ethics, ensuring high-quality and maintainable code.
Gender	-
Human Values	Proficiency in rendering, updating, and component development allows for the creation of user-friendly, accessible, and inclusive web applications, aligning with human-centered design principles.
Environment & Sustainability	-
Unit V	**Redux**
Local	-
Regional	-
National	-
Global	Proficiency in integrating React and Redux is valuable in the global web development community.
Employability	Proficiency in Redux concepts and principles enhances employability, as these skills are highly sought after in the job market.

Entrepreneurship	Knowledge of Redux enables entrepreneurs to develop scalable and data-driven digital products or services.
Skill Development	The syllabus focuses on developing technical skills in state management and integration, fostering skill enhancement and industry readiness
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4, SDG 8
NEP 2020	-
POE/4th IR	-

FRONT END DEVELOPMENT WITH REACTJS LAB

Department:	Department of Electrical & Electronics Engineering		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Front End Development With Reactjs Lab	**ENEE151**	0-0-2	1
Type of Course:	MINOR		

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Identify the syntax and structure of JSX for rendering React components.
CO 2	Develop React components using JSX syntax to render dynamic UI elements.
CO 3	Assess the effectiveness of different routing strategies for creating multi-page applications.
CO 4	Construct a complete and functional web application that demonstrates comprehensive React skills.

Ex. No.	Experiment Title	Mapped CO/COs
1	Create a basic React app with "Hello World" text using JSX.	CO1
2	Create a parent component that renders multiple child components with different props.	CO1
3	Create a counter app using state to increment and decrement a value.	CO1

4	Build a form component that captures user input and updates the state.	CO2
5	Style a component using CSS classes and inline styles.	CO2
6	Create a responsive navigation bar using Flexbox or CSS Grid.	CO3
7	Implement a card component with dynamic data, such as images and text.	CO3
8	Set up a Redux store and create actions and reducers for a basic counter.	CO4
9	Implement asynchronous actions using Redux Thunk to fetch data from an API.	CO1
10	Create a multi-page React app using React Router.	CO4
11	Build a navigation menu that highlights the active link using route matching.	CO3
12	Implement a weather app that fetches weather data from an API based on user input.	CO2
13	Write unit tests for React components using Jest and React Testing Library.	CO2
14	Set up continuous integration (CI) for your React app using platforms like Travis CI or GitHub Actions.	CO3
15	Deploy your React app to a hosting platform like Netlify, Vercel, or GitHub Pages.	CO4

Semester: 3

DISCRETE MATHEMATICS

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Discrete Mathematics	**ENCS203**	3-1-0	4
Type of Course:	Programme Core		
Pre-requisite(s), if any: Basic of Mathematics			
Brief Syllabus: This course will discuss fundamental concepts and tools in discrete mathematics with emphasis on their applications to computer science. Topics include logic and Boolean circuits, sets, functions, relations, deterministic algorithms and randomized algorithms, analysis techniques based on counting methods and recurrence relations, trees and graphs etc.			
UNIT WISE DETAILS			
Unit Number: 1	**Propositional Logics & Relations**	**No. of hours: 12**	
Content Summary: **Mathematical Logic:** Introduction to Mathematical Thinking , Propositional and Predicate Logic, Propositional Equivalences, Sets, Binary Relation, Equivalence Relation, Logical operations, Conditional Statements, Tautologies, Contradictions, Logical Equivalence, The use of Quantifiers, Normal Forms, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference. Sets **and Relations**: Set Operations, Representation and Properties of Relations & Functions, Equivalence Relations, Partially Ordering.			
Unit Number: 2	**Title: Counting, Mathematical Induction and Discrete Probability**	**No. of hours: 12**	

Content Summary: Basics of Counting, Pigeonhole Principle, Permutations and Combinations, Inclusion-Exclusion Principle, Mathematical Induction, Probability, Bayes' Theorem, Discrete Probability Theory, Discrete Structures in Computing, Counting Principles, Permutations and Combinations, Probability Theory, Discrete Random Variables, Discrete Optimization - Optimization Problems and Algorithms, Linear Programming, Integer Programming, Algebraic Structures - Groups (Definition, Properties, Subgroups, Cyclic Groups), Rings (Definition, Properties, Integral Domains, Fields), Isomorphisms and Homomorphisms, Counting and combinatorics.

Unit Number: 3	**Title: Group Theory & Discrete Probability**	**No. of hours: 8**

Content Summary: Groups, Subgroups, Semi Groups, Product and Quotients of Algebraic Structures, Isomorphism, Homomorphism, Automorphism, Rings, Integral Domains, Fields, Applications of Group Theory, Combinatorial optimization: basic concepts and algorithms, Sample spaces, events, and probability axioms, Conditional probability and Bayes' theorem.

Unit Number: 4	**Title: Graph Theory**	**No. of hours: 8**

Content Summary: Simple Graph, Multigraph, Weighted Graph, Paths and Circuits, Shortest Paths in Weighted Graphs, Eulerian Paths and Circuits, Hamiltonian Paths and Circuits, Planner graph, Graph Coloring, Bipartite Graphs, Trees and Rooted Trees, Prefix Codes, Tree Traversals, Spanning Trees and Cut-Sets, digraphs, Graph Coloring, Euler's formulae, Graph Theory, Networks and Flows.

***Self-Learning Components:**

Topics (with book references):

1. Applications of Graph Coloring: Time table Scheduling ("Discrete Mathematics and Its Applications" by Kenneth H. Rosen: Chapter 10.3: Graph Coloring)
2. Network Analysis, Routing & Optimization, using graph theory.(Introduction to Graph Theory" by Richard J. Trudeau)
3. Combinatorial Optimization & Error Detection & correction using The Pigeonhole Principle ("Combinatorial Optimization: Algorithms and Complexity" by Christos H. Papadimitriou and Kenneth Steiglitz)
4. Scheduling and Task Prioritization, using Partial ordering. ("Introduction to Scheduling" by Yves Robert and Frederic Vivien)
5. Rules based system and Algorithm design using conditional statements. (Chapter 10, 22, 23, of Artificial Intelligence: A Modern Approach" by Stuart Russell and Peter Norvig).

Online Certification Courses for Discrete Mathematics (With Links):

1. Discrete Mathematics: https://www.coursera.org/learn/discrete-mathematics
2. Mathematics For Computer Science, https://ocw.mit.edu/courses/6-042j-mathematics-for-computer-science-fall-2010/
3. Introduction to Discrete Mathematics for Computer Science Specialization, https://www.coursera.org/specializations/discrete-mathematics
4. Discrete Math Series : Propositional Logic masterclass https://www.udemy.com/course/discretemathematics/
5. Master Discrete Mathematics: Sets, Math Logic, and More: https://www.udemy.com/course/master-discrete-mathematics/
6. Master Math by Coding in Python: https://www.udemy.com/course/math-with-python/
7. Discrete Mathematics for Computer Science in C, Java, Python: https://www.udemy.com/course/discrete-mathematics-and-its-applications/
8. Discrete Mathematics - Complete Course: https://www.udemy.com/course/discrete-mathematics-complete-course/
9. Discrete Optimization: https://www.coursera.org/learn/discrete-optimization
10. Introduction to Discrete Mathematics for Computer Science Specialization: https://www.coursera.org/specializations/discrete-mathematics

NPTEL Lecture Links for Discrete Mathematics (With Links):

1. Discrete Mathematics _ IIITB, IIIT Bangalore, Prof. Ashish Choudhury: https://nptel.ac.in/courses/106108227
2. Discrete Mathematics, IIT Ropar: https://nptel.ac.in/courses/106106183

Reference Books of Discrete Mathematics:

1. Elements of Discrete Mathematics, C. L Liu, McGraw-Hill Inc, 1985. Applied Combinatorics, Alan Tucker.
2. Concrete Mathematics, Ronald Graham, Donald Knuth, and Oren Patashnik, 2nd Edition - Pearson Education Publishers.
3. Combinatorics: Topics, Techniques, Algorithms by Peter J. Cameron, Cambridge University Press.
4. Topics in Algebra, I.N. Herstein, Wiley.
5. Kenneth H. Rosen, Discrete Mathematics and its Applications, Tata McGraw – Hill
6. Satinder Bal Gupta: A Text Book of Discrete Mathematics and Structures, University Science Press, Delhi.

E-Books of Discrete Mathematics (with Links):

1. Discrete Mathematics: An open Introduction, by Oscar Levin, 3rd Edition: https://discrete.openmathbooks.org/pdfs/dmoi-tablet.pdf

2. Lecture Notes on Discrete Mathematics, IITK, https://home.iitk.ac.in/~arlal/book/mth202.pdf
3. Mathematical Foundations And Aspects of Discrete Mathematics, Jean Gallier and Jocelyn Quaintance, https://www.cis.upenn.edu/~jean/discmath-root-b.pdf
4. Discrete Mathematics for Computer Science, Gary Haggard, John Schlipf, Sue Whitesides, https://www2.cs.uh.edu/~arjun/courses/ds/DiscMaths4CompSc.pdf
5. DISCRETE MATHEMATICS FOR COMPUTER SCIENCE, Herbert Edelsbrunner and Brittany Fasy, https://courses.cs.duke.edu/spring09/cps102/Lectures/Book.pdf
6. Discrete Mathematics and its Applications, Rosen, https://faculty.ksu.edu.sa/sites/default/files/rosen_discrete_mathematics_and_its_applications_7th_edition.pdf

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** foundational concepts: Gain a solid understanding of fundamental concepts in discrete mathematics, including logic, sets, relations, and functions
CO2	**Express** proficiency in logical reasoning and constructing mathematical proofs using various proof techniques such as direct proofs, proof by contradiction, and mathematical induction.
CO3	**Determine** methods to Explore various discrete structures, such as sets, sequences, functions, relations, and formal languages. Understand the properties and applications of these structures.
CO4	**Identify** and develop problem-solving skills by applying discrete mathematics concepts to solve mathematical problems and real-world scenarios. Enhance logical thinking and analytical reasoning abilities.
CO5	**Articulate** real-world applications of discrete mathematics in computer science, cryptography, network analysis, optimization problems, scheduling, and decision-making.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A1	P1
C02	C3	A2	P2
CO3	C3	A5	P5
CO4	C6	A5	P5
CO5	C2	A5	P1

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	-	2	-	2	-	-	-	-	2
CO2	1	2	-	1	3	2	1	-	-	-	-	2
CO3	-	-	-	1	3	-	2	-	-	3	-	2
CO4	-	2	-	-	3	1	2	-	-	3	-	2
CO5	-	2	-	-	3	-	2	-	-	3	-	2

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PO1	PO2	PO3	PSO4	PSO5
CO1	3	2	2	1	-
CO2	2	3	2	1	-
CO3	2	3	2	1	1
CO4	2	3	2	1	-
CO5	2	3	2	1	1

Relevance of the Syllabus to various indicators

Unit I	**Introduction**
Local	-
Regional	-
National	-
Global	It lays a solid foundation for further studies in mathematics, computer science, and related fields while fostering critical thinking and analytical skills.
Employability	Equips with problem-solving techniques to analyse and process data, design algorithms, and make informed decisions.
Entrepreneurship	-
Skill Development	Discrete mathematics allows students to think abstractly, develop formal mathematical arguments, and engage in rigorous problem-solving.
Professional Ethics	-
Gender	-
Human Values	-

Environment & Sustainability	-
Unit II	**Counting, Mathematical Induction and Discrete Probability**
Local	-
Regional	-
National	-
Global	Probability, Bayes' theorem, and statistical analysis provide a framework for understanding and interpreting real-world phenomena that involve uncertainty and data.
Employability	It is beneficial in areas such as probability theory, statistics, optimization, cryptography, and network analysis
Entrepreneurship	skills obtained are valuable in various fields, including computer science, mathematics, law, and philosophy.
Skill Development	Enhances your ability to analyze problems logically, identify patterns, and draw logical conclusions. These skills are valuable in various fields, including computer science, mathematics, law, and philosophy.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Group Theory**
Local	-
Regional	-
National	-
Global	Group theory is widely used in physics, chemistry,

	crystallography, and other fields where symmetry is a fundamental concept.
Employability	This develops ability to think conceptually, make connections between different mathematical structures, and develop a broader perspective on mathematics as a whole.
Entrepreneurship	-
Skill Development	Group theory, in particular, is essential for studying symmetry and transformations. It provides a framework for analysing the symmetries of objects, understanding transformational properties, and solving problems related to symmetry.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Graph Theory**
Local	-
Regional	-
National	-
Global	By studying these topics, the students will gain the ability to model and analyse various real-world scenarios, including social networks, transportation networks, communication networks, and data dependencies.
Employability	Understanding concepts such as shortest paths, network connectivity, and digraphs allows students to design efficient and reliable routing algorithms, analyze network performance, and ensure optimal data
Entrepreneurship	-
Skill	Graph theory provides a powerful framework for representing

Development	and analyzing relationships between objects or entities.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 9
NEP 2020	-
POE/4th IR	Aligns with the concepts of Design, Efficiency, Problem Solving

BACKEND DEVELOPMENT WITH NODEJS

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Backend Development with NodeJS	**ENSP201**	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any: Basics of Computer Programming			
Brief Syllabus: The course on Node JS teaches you how to use JavaScript to create network applications quickly and efficiently. The training is intended to assist developers in understanding and developing web applications using JavaScript.			
UNIT WISE DETAILS			
Unit Number: 1	**Title:** Introduction to Node.JS	**No. of hours: 8**	
Content Summary: What is Node.js, History of Node.js, Why Node.js, Node.js Architecture, Working and Features, Installation and Setup, Installing Node.js, Launching REPL, Environment, Installing Visual Studio, Code Editor, Components of Node.js			
Unit Number: 2	**Title: Modules and Files**	**No. of hours: 10**	
Content Summary: Module Exports, Export Object, Export Functions, Export Functions as Class, Loading module from, Separate Folder, Modules, File System Module, Reading and Writing into, Files, Appending and Opening Files, Events and Event Emitters, Handling Events, Customized Class for Handling Events, Inbuilt Modules File Systems, Operating System.			
Unit Number: 3	**Title: Buffer**	**No. of hours: 8**	

Content Summary:

Buffers: Writing to Buffers, Reading from Buffers, Concatenating Buffers, Copying Buffers, Slicing Buffers, The Stream Module, Reading From Stream, Writing to Stream, Pipes, Pipe Chaining

Unit Number: 4	**Title: Rest API**	**No. of hours: 8**

Content Summary:

REST API: Explain REST API, Describe Node.js express, Discuss the importance of express, Explain the installing process of express, Learn express request and response, Describe routing, REST API : Intro to API, History of API Development, Development of AJAX, CRUD

***Self-Learning Components:**

- Study of MongoDB
- Performing CRUD (Create, Read, Update, Delete) operations on databases.
- Study of integrating Node.js applications with CI/CD pipelines

Reference Books:

1. Manuel Kiessling, '*The Node Beginner Book'*, Leanpub
2. Griggs Bethany, '*Node Codebook'*, Packt
3. Ethan Brown, 'Web Development with Node and Express', O Reilly
4. David Herron, '*Node.JS Web Development*', Packt

Define Course Outcomes (CO)

COs	Statements
CO1	Compare and contrast SQL and NoSQL databases in the context of Node.js applications.
CO2	Implement user authentication and session management using Passport.js.
CO3	Build routes to handle different HTTP requests and responses.
CO4	Examine the security vulnerabilities in a Node.js application and propose

	security measures.
CO5	Design and develop a complete RESTful API with user authentication and CRUD operations using Express.js and Node.js.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels (C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels (A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels (P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A1	P1
C02	C3	A2	P2
CO3	C3	A5	P5
CO4	C6	A5	P5
CO5	C2	A5	P1

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	2	-	2	-	2	-	-	-	-	2
CO2	1	2	-	1	3	2	1	-	-	-	-	2
CO3	-	-	-	1	3	-	2	-	-	3	-	2
CO4	-	2	-	-	3	1	2	-	-	3	-	2

CO5	-	2	-	-	3	-	2	-	-	3	-	2

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PO1	PO2	PO3	PSO4	PSO5
CO1	3	2	2	1	1
CO2	2	3	2	1	1
CO3	2	3	2	1	1
CO4	2	3	2	1	-
CO5	2	3	2	1	-*

Relevance of the Syllabus to various indicators

Unit I	**Introduction to Node.JS**
Local	-
Regional	-
National	-
Global	-*
Employability	Mastery of module exports, file system operations, event handling, and using inbuilt modules directly enhances students' employability by providing practical coding skills.
Entrepreneurship	Knowledge of module exports, file manipulation, event-driven programming, and utilizing built-in modules empowers students to create innovative software solutions, supporting entrepreneurship.
Skill	Understanding module exports, file interactions, event-driven

Development	programming, and built-in modules is globally relevant, as these concepts are fundamental in software development worldwide.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Modules and Files**
Local	-
Regional	-
National	-
Global	Understanding module exports, file interactions, event-driven programming, and built-in modules is globally relevant, as these concepts are fundamental in software development worldwide.
Employability	Mastery of module exports, file system operations, event handling, and using inbuilt modules directly enhances students' employability by providing practical coding skills.
Entrepreneurship	Knowledge of module exports, file manipulation, event-driven programming, and utilizing built-in modules empowers students to create innovative software solutions, supporting entrepreneurship
Skill Development	Proficiency in module exports, file operations, event handling, and using inbuilt modules contributes significantly to the technical skill development of students.
Professional Ethics	-
Gender	-

Human Values	-
Environment & Sustainability	-
Unit III	**Buffer**
Local	-
Regional	-
National	-
Global	Understanding buffer operations, stream handling, and pipes is globally relevant, as these concepts are fundamental to software development practices worldwide
Employability	Proficiency in buffer management, stream processing, and pipe operations directly enhances students' employability, as these skills are widely sought after by technology companies.
Entrepreneurship	-
Skill Development	Proficiency in working with buffers, streams, and pipes is globally relevant, as these are fundamental concepts in software development.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Rest API**
Local	-
Regional	-
National	-
Global	

Employability	Proficiency in REST API and Express can enhance students' employability as many organizations across the country require professionals with expertise in these technologies.
Entrepreneurship	Understanding REST API and Express can empower students to create innovative web applications and services, fostering entrepreneurship at a regional level.
Skill Development	Knowledge of REST API and Node.js Express can enhance the skill set of students and make them competitive in the regional job market.
Professional Ethics	Understanding REST API development and Node.js Express includes awareness of best practices, security considerations, and ethical handling of user data, aligning with professional ethics.
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 9
NEP 2020	-
POE/4th IR	Aligns with the concepts of Design, Efficiency, Problem Solving

BACKEND DEVELOPMENT WITH NODEJS LAB

Department:	Department of Electrical & Electronics Engineering		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Basics of Electrical & Electronics Lab	**ENEE151**	0-0-2	1
Type of Course:	Minor		

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Compare and contrast various approaches to error handling and debugging in Node.js.
CO 2	Create and manage middleware functions to intercept and modify incoming requests and outgoing responses.
CO 3	Evaluate the effectiveness of error handling and debugging strategies in identifying and resolving issues in a Node.js application
CO 4	Develop a comprehensive backend application from scratch, integrating multiple components and implementing advanced features.

Ex. No.	Experiment Title	Mapped CO/COs
1	Install Node.js and npm. Set up a development environment (IDE, code editor, terminal).	CO4, CO1
2	Explore the event-driven architecture of Node.js. Understand the role of the V8 engine in Node.js.	CO4, CO1
3	Create custom modules and import/export functions.	CO2

4	Handle multiple asynchronous operations using callbacks.	CO2
5	Create routes, request handlers, and responses. Implement middleware for request processing.	CO3
6	Implement CRUD operations using Express.js. Validate and sanitize user input in API requests.	CO3
7	Perform CRUD operations on a database. Implement data validation and error handling.	CO3
8	Use authentication middleware for secure routes. Manage user sessions and tokens.	CO2
9	Use debugging tools and techniques for troubleshooting.	CO1, CO4
10	Use testing frameworks (e.g., Mocha, Chai) to test APIs.	CO4
11	Explore strategies for scaling and managing traffic.	CO4
12	Secure API endpoints using HTTPS and SSL certificates.	CO4

K.R. MANGALAM UNIVERSITY

DATA STRUCTURES

Department:	**Department of Computer Science and Engineering**		
Course Name: **Data Structure**	**Course Code**	**L-T-P**	**Credits**
	ENCS205	4-0-0	4
Type of Course:	Major		
Pre-requisite(s), if any: Basics of Computer Programming			

Brief Syllabus:

Solving computational problems requires the knowledge of efficient data organization and the ability to make effective choices among multiple solutions. In this course, we will explore several fundamental data structures in computer science and learn to implement them. The course aims to teach the fundamentals of data structures, their design, implementation and effective use in problem solving approach. With the knowledge of data structures and practical experience in implementing them, students can become much more effective designer and developer. The course will start with the basic introduction of linear such as arrays, stack and queues as well as non-linear data structures such as trees and graphs. They will further proceeds with the programming intensive task of implementing them.

UNIT WISE DETAILS

Unit Number: 1	**Title: Introduction to Data Structure**	**No. of hours: 12**

Content Summary:

Introduction to Data Structures: Definition of data structures and abstract data types, Static and Dynamic implementations, Examples and real life applications; Arrays: ordered lists, representation of arrays in memory

Basic Analysis: Differences among best, average, and worst case behaviours of an algorithm, Asymptotic analysis of upper and expected complexity bounds, Big O notation: formal definition and use, big omega and big theta notation , Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential, Time and space trade-offs in algorithms, Recurrence relations , Analysis of iterative and recursive algorithms.		
Unit Number: 2	**Title: Stacks, Queues and Linked List**	**No. of hours: 12**
Content Summary: **Stacks:** ADT Stack and its operation, Array based implementation of stacks, Examples: Infix, postfix, prefix representation, Conversions of an arithmetic expression from Infix to postfix, Applications: Tower of Hanoi Problem, Algorithms and their complexities **Queues:** ADT Queue and its operation, Array based implementation of linear Queues, Circular Queues, Priority queues, Application of queues: Process Scheduling. **Linked List:** Definition, Components of linked list, Representation of linked list, Advantages and Disadvantages of linked list. Types of linked list: Singly linked list, Doubly linked list, Circular linked list and circular doubly linked list. Operations on different types of linked list : creation, insertion, deletion, search and display (based on the different position as specified by the user).Linked representation of Stacks & Queues, Algorithms and their complexities		
Unit Number: 3	**Title: Trees and Graphs**	**No. of hours: 12**

Content Summary:

Trees: Basic Terminology, Binary Trees and their representation, expression evaluation, Complete Binary trees, Extended binary trees, traversing binary trees, Searching, Insertion and Deletion in binary search trees. AVL trees, Threaded trees, algorithms and their analysis.

Graphs: Terminology and Representations, Graphs & Multigraphs, Directed Graphs, Sequential representation of graphs, Adjacency matrices, Transversal Connected Component and Spanning trees, algorithms and their analysis.

Unit Number: 4	Title: Sorting and Searching	No. of hours: 8

Content Summary:

Sorting Algorithms: Introduction, Sorting by exchange, selection sort, insertion sort, Bubble sort, Selection sort, Efficiency of above algorithms, Shell sort, Performance of shell sort, Merge sort, Quick sort Algorithm analysis, Heap sort: Heap Construction, Heap sort, bottom – up, Top – down Heap sort approach

Searching Algorithms: Straight Sequential Search, Binary Search (recursive & non–recursive Algorithm

***Self-Learning Components:**

1. Students should explore Platforms like LeetCode, HackerRank for Data structure

2. Students can refer the following courses as per the **Open Source University Curriculum**

- "Algorithms, Part I" by Robert Sedgewick and Kevin Wayne (available on Coursera)
- "Algorithms, Part II" by Robert Sedgewick and Kevin Wayne (available on Coursera)

Reference Books:

1. E. Horowitz and S. Sahani, "Fundamentals of Data Structures", Galgotia Book source Pvt. Ltd.
2. Data Structures & Algorithms in Python by John Canning, Alan Broder, Robert Lafore Addison-Wesley Professional ISBN: 9780134855912.
3. "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein.
4. Problem Solving with Algorithms and Data Structures Using Python" by Brad Miller and David Ranum.

Define Course Outcomes (CO)

COs	Statements
CO1	**Analyze** the algorithms to determine the time and space complexity and justify the correctness.
CO2	**Design** a given Search problem (Linear Search and Binary Search).
CO3	**Articulate** Data Structure concepts such as Stack, Queue, Linked list, Graph and traversal techniques and their use in programs
CO4	**Design & implement** the algorithm for Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap sort. Compare their performance in term of Space and time complexity

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels (C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1	A3	P5
C02	C2	A3	P4
CO3	C3,C4	A4	P3
CO4	C5	A2	P2

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	3	-	-	1	-	-	-	-	-	3
CO2	3	3	2	-	-	-	-	-	-	-	-	3
CO3	3	3	3	-	-	-	-	-	-	-	-	3
CO4	3	3	3	-	-	-	-	-	-	-	-	3

Justification for mapping must be relevant

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	2	-	-
CO2	3	3	1	-	-

CO3	3	3	-	-	1
CO4	3	3	-	-	-

Relevance of the Syllabus to various indicators

Unit I	**Introduction to Data Structure**
Local	-
Regional	-
National	It provides foundational knowledge in data structures and algorithm analysis, which are fundamental concepts in computer science and software engineering.
Global	The principles taught in this course are applicable worldwide and form the basis of software engineering practices globally.
Employability	Understanding these concepts is crucial for technical interviews and can enhance job prospects in various technology companies.
Entrepreneurship	It helps in developing efficient and scalable software solutions, which are essential for building successful tech startups or innovative ventures.
Skill Development	Develop skills that are fundamental to computer science and software development and can be applied in various programming languages and contexts.
Professional Ethics	Applying good coding practices and software engineering principles align with professional ethics in the field.
Gender	-
Human Values	It encourages students to approach problems analytically and develop efficient solutions that can positively impact human lives.
Environment & Sustainability	-

Unit II	**Stacks, Queues and Linked List**
Local	-
Regional	-
National	These data structures are used extensively in computer science and software engineering, and the skills learned in this course can be applicable to various industries and sectors across the country.
Global	The principles taught in this course are applicable worldwide and form the basis of software engineering practices globally.
Employability	Understanding these data structures and their operations is important for solving problems efficiently and implementing optimized algorithms, which are highly sought after skills in the job market.
Entrepreneurship	Knowledge of data structures like stacks, queues, and linked lists is valuable for entrepreneurship in the technology sector.
Skill Development	Understanding and applying concepts related to stacks, queues, and linked lists enhances programming skills and helps in developing efficient algorithms to solve real-world problems.
Professional Ethics	-
Gender	-
Human Values	Fostering critical thinking, problem-solving skills, and logical reasoning, which are important qualities in a technology-driven society which can improve productivity and streamline processes, thus positively impacting human lives.
Environment & Sustainability	-
Unit III	**Trees and Graphs**
Local	-
Regional	-

National	The skills learned in this course can be applicable to- various industries and sectors across the country.
Global	The principles taught in this course are applicable worldwide and form the basis of software engineering practices globally.
Employability	Knowledge of data structures such as trees and graphs is highly relevant to employability in the field of software development and computer science.
Entrepreneurship	These data structures are commonly used in designing and developing software solutions, and understanding their implementation and applications can help entrepreneurs build innovative and scalable products.
Skill Development	Understanding and applying concepts related to trees and graphs enhances programming skills and helps in developing efficient algorithms to solve real-world problems.
Professional Ethics	Following best practices in data structure implementation and algorithm design promotes code readability, maintainability, and overall software quality.
Gender	-
Human Values	Understanding data structures like trees and graphs enables students to develop efficient algorithms that can improve productivity, streamline processes, and positively impact human lives.
Environment & Sustainability	-
Unit IV	**Sorting and Searching**
Local	
Regional	Understanding and applying these algorithms are fundamental to computer science and software development, making them relevant at a regional level.
National	It provides foundational knowledge in sorting and searching algorithms.
Global	Sorting and searching algorithms are fundamental building

	blocks in computer science and software development, used globally.
Employability	Understanding these algorithms and their efficiency helps in developing optimized software solutions, which are highly sought-after skills in the job market.
Entrepreneurship	These algorithms are used extensively in data processing, information retrieval, and optimization problems, which are essential in building innovative and scalable software products.
Skill Development	Understanding and applying sorting and searching algorithms enhances programming skills and helps in developing efficient algorithms to solve real-world problems.
Professional Ethics	Following best practices in algorithm design and implementation promotes code readability, maintainability, and overall software quality.
Gender	-
Human Values	Understanding sorting and searching algorithms enables students to develop efficient solutions that improve productivity, streamline processes, and positively impact human lives.
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	Aligns with the concepts of Design, Efficiency ,Problem Solving, Abstraction and System Analysis

DATA STRUCTURES LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: **Data Structure lab**	**Course Code**	**L-T-P**	**Credits**
	ENCS253	0-0-2	1
Type of Course:	Major		
Pre-requisite(s), if any: Basics of Computer Programming			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Equip the students with knowledge of algorithms and analysis of space and time complexity of the algorithms
CO 2	Demonstrate the use of stack, queues and linked list
CO 3	Equip the students with tree and graph data structures and their practical applications
CO 4	Implementing and analysing searching and sorting algorithms

Ex No	Experiment Title	Mapped CO/COs
1	To design, implement and analyze the complexity of Linear search algorithm	CO4
2	To design, implement and analyze the complexity of Binary search algorithm	CO4
3	Implement and compare the time complexity of bubble sort, insertion sort and selection sort. Calculate their running times	CO4

	for best, worst & best cases. Draw the three cases in a single graph to justify its observed time complexities.	
4	Implement and analyse the working of Recursive Algorithms	CO1
5	Implement Quick sort algorithm and calculate its running times for best, worst & best cases. Draw the three cases in a single graph to justify its observed time complexities.	CO4
6	Implement the linear data structure: Stack by performing Push and Pop operation	CO2
7	Implement Postfix and Prefix Expression using Stack	CO2
8	Implement reverse of a String using Stack	CO2
9	Implement the linear data structure: Queue by performing Insertion and Deletion operation	CO2
10	Implement Circular Queue by performing Insertion and Deletion operation	CO3
11	Implement the dynamic data structure: single linked list also analyse their time complexities in three cases: a) Inserting a new node at the beginning b) Inserting a new node at the end c) Deleting a node from the beginning	CO2
12	Consider a linked list L reverse the linked list	CO2
13	Implement the dynamic data structure : doubly linked list also analyse their time complexities in three cases: a) Inserting a new node at the beginning b) Inserting a new node in the middle c) Deleting a node from the end	CO2
14	Implement the dynamic data structure : circular linked list also analyse their time complexities in three cases: a) Inserting a new node at the beginning b) Inserting a new node in the middle c) Deleting a node from the end	CO2
15	Implement and analyse Stack implementation using Linked list	CO2
16	Implement and analyse Queue implementation using Linked	

	list	
17	Implement and analyse the tree traversal algorithms 1) Inorder 2) Preorder 3) Post order	CO3
18	Implement and analyse the following operations of Binary Search tree. a) Creating and inserting a new node b) Searching a node c) Deleting an existing node from BST	CO3
19	Implement AVL tree with insertion, deletion and searching operation	CO3
20	Implement the graph traversal techniques: Depth First search and Breadth First search algorithms	CO3
21	To understand and implement the minimum spanning tree in Graphs using Kruskal Algorithm	CO3
22	To understand and implement the minimum spanning tree in Graphs using Prims Algorithm	CO3
23	Implement Merge sort algorithm and calculate its running times for best, worst & best cases. Draw the three cases in a single graph to justify its observed time complexities.	CO4
24	Implement Heap sort algorithm and calculate its running times for best, worst & best cases. Draw the three cases in a single graph to justify its observed time complexities.	CO4
25	Implement a priority queue using a heap and calculate its running times for best, worst & best cases. Draw the three cases in a single graph to justify its observed time complexities.	CO4
	Mini Project 1: Create a student management system that stores and manages student records using various data structures. The system should allow users to perform operations such as adding new students, searching for students, deleting students, and displaying all student records.	CO4
	Mini Project 2: Implement a maze solver using data structures like stacks or queues. The program should take an	CO3

	input maze, find a path from the starting point to the goal, and output the solution. You can use depth-first search (DFS) or breadth-first search (BFS) algorithms to solve the maze.	
	Mini Project 3: Implement a social network analysis tool using data structures like graphs. The tool should be able to read a network of users and their connections and perform operations like finding the shortest path between two users, identifying influential users, or recommending friends.	CO2

LIFE SKILLS FOR PROFESSIONALS-I

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Life Skills for Professionals - I	**AEC011**	3-0-0	3
Type of Course:	AEC		
Pre-requisite(s), if any:			
Brief Syllabus: Through this comprehensive course, the learners will develop a solid foundation in communication skills, enabling them to express themselves confidently, listen actively, and build strong relationships in personal and professional contexts.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Communication: An Introduction**		**No. of hours: 6**
Content Summary: Definition, Nature and Scope of Communication, Importance and Purpose of Communication, Process of Communication, Types of Communication, Barriers to Communication, Essentials of Effective Communication			
Unit Number: 2	**Non-Verbal Communication**		**No. of hours: 6**
Content Summary: Personal Appearance, Gestures, Postures, Facial Expression, Eye Contacts, Body Language (Kinesics) Time language, Tips for Improving Non-Verbal Communication			
Unit Number: 3	**Title: Basic number system**		**No. of hours: 6**
Content Summary: Divisibility, Unit digit, Last two digit, Remainder, Number of zero, Factor, LCM & HCF, Simplification, Mixture, Average, Ratio, and Partnership.			

Unit Number: 4	Title: Number system	No. of hours: 6
Content Summary: Factor, LCM & HCF, Simplification, Mixture, Average, Ratio, and Partnership.		
Unit Number: 5	**Title: Time Management**	**No. of hours: 6**
Content Summary: Time management strategies, setting goals, organizing, and planning ahead, Making the most of your time Deal with distractions, Procrastination and Avoiding distractions.		
***Self-Learning Components:** **https://onlinecourses.nptel.ac.in/noc21_hs02/preview** **Please Note:** 1)Students are supposed to learn the components on self-basis 2) At least 5-10 % syllabus will be asked in end term exams from self-learning components		
Reference Books: Aggarwal, R. S. (2014). Quantitative aptitude (Revised edition). Gladwell, M. (2021). Talking to strangers. Scott, S. (2004). Fierce conversations.		

Define Course Outcomes (CO)

COs	Statements
CO1	**Perform** calculations related to number systems, percentages and averages, quickly and accurately.
CO2	**Exhibit** confidence in tackling multiple-choice questions, time-constrained tests and competitive examinations.
CO3	**Demonstrate** active listening techniques, including attentive listening and reflection

CO4	**Articulate** and speak with confidence and express ideas clearly and coherently.
CO5	**Improve** confidence and display open and positive non-verbal communication.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels (C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C3	-	-
CO2	C2	-	-
CO3	C2	-	P4
CO4	C6	-	-
CO5	C6	-	P5

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 1	3	-	2	-	-	-	-	-	1		1	3
CO 2	-	3	-	-	1	-	-	-	-	-	-	3
CO 3	-	1	-	1	-	-	2	-	-	-	1	3
CO 4	-	2	-	-	2	-	-	-	-	3	-	3
CO 5	-	-	3	2			1	-	-	-	1	3

Justification for mapping must be relevant.

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO 1	3	-	1	-	1
CO 2	3	1	-	-	1
CO 3	1	1	-	-	1
CO 4	2	2	3	-	2
CO 5	3	1	-	-	-

Relevance of the Syllabus to various indicators

Unit I	**Communication : An Introduction**
Local	Improve number sense, enhance basic communication skills.
Regional	Recognize the importance of continuous learning and practice to maintain and further develop mental ability.
National	Practice time management strategies for solving problems within time constraints, as in competitive exams.
Global	Aligns with global trends in employment
Employability	Develop skills in real-life situations, such as academic exams, job interviews, and problem-solving scenarios.
Entrepreneurship	Learn to share ideas, listen to others, build consensus, and manage conflicts to achieve common goals in collaborative settings.
Skill Development	Develops Skills in public speaking, interpersonal communication, professional writing, and persuasive communication.
Professional Ethics	-
Gender	-

Human Values	-
Environment & Sustainability	-
Unit II	**Non-Verbal Communication**
Local	Recognize the importance of continuous learning and practice to maintain and further develop mental ability.
Regional	Practice attentive listening techniques, such as paraphrasing and asking clarifying questions.
National	Attentively listen to others, understand their perspectives, and respond appropriately while exhibiting techniques such as maintaining eye contact, asking clarifying questions, and paraphrasing.
Global	Aligns with global trends in employment
Employability	Develop skills in participating and contributing to group discussions, meetings, or presentations.
Entrepreneurship	Learn to share ideas, listen to others, build consensus, and manage conflicts to achieve common goals in collaborative settings.
Skill Development	Apply skills in real-life situations, such as academic exams, job interviews, and problem-solving scenarios.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Number System**
Local	Improve number sense, enhance basic arithmetic skills and strengthen mental math abilities and speed.
Regional	-
National	Learn about number systems, ratios, proportions, and percentages
Global	Recognize the importance of continuous learning and practice to maintain and further develop mental ability.
Employability	Develop skills in participating and contributing to group discussions, meetings, or presentations.
Entrepreneurship	-

Skill Development	Recognize the importance of continuous learning and practice to maintain and further develop mental ability.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Time Management**
Local	Attentively listen to others, understand their perspectives, and respond appropriately with timelines
Regional	-
National	Contributes to develop skill and improved productivity
Global	Aligns with global trends in understanding the deadlines.
Employability	Enhance the employability of individuals by developing essential skills and competencies sought by employers
Entrepreneurship	-
Skill Development	Strengthening critical thinking, problem-solving, memory, and other cognitive functions to improve overall mental agility and performance.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	Aligns with the concepts employability

SUMMER INTERNSHIP / PROJECT-I

Department:	**Department of Computer Science and Engineering**		
Course Name: Summer Internship / Project-I	**Course Code**	**L-T-P**	**Credits**
	ENSI251	0-0-0	2
Type of Course:	INT		
Pre-requisite(s), if any: NA			

The duration of the internship will be two weeks. It will be after completion of 2nd Semester and before the commencement of Semester III.

The following options can be opted by the students:

1. Offline internship in industry - Student is supposed to produce a joining letter and relieving letter once the internship is over in case of Offline internship in any industry.

2. Online internships – with organizations /institutions those are approved /supported / recommended by the All-India Council of Technical Education for Internship (like SWAYAM, NPTEL, Internshala etc.).

Report Submission and Evaluation Guidelines:

• Student must prepare a detailed report and submit the report. A copy of the report can be kept in the departments for record.

• Each student must be assigned a faculty as a mentor from the university and an Industry Expert as External Guide or Industry Mentor.

• The presentation by student for Internship/ project should in the presence of all students is desirable.

• Student should produce successful completion certificate in case of summer internship in industry.

Course Outcomes:

At the end of the course, students will be able to:

1. Get exposure to the industrial environment, which cannot be simulated in the classroom and hence creating competent professionals for the industry.
2. Get possible opportunities to learn, understand and sharpen the real time technical / managerial skills required at the job(s).
3. Gain experience in writing technical reports / projects and presentation of it.
4. Learn and gain exposure to the engineer's responsibilities and ethics.
5. Understand the social, economic, and administrative considerations that influence the working environment of industrial organizations

Semester: 4

ANALYSIS AND DESIGN OF ALGORITHMS

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Analysis and Design of Algorithms	**ENCS202**	4-0-0	4
Type of Course:	**Major**		
Pre-requisite(s), if any: - Data Structure			

Brief Syllabus:

The analysis and design of algorithm course introduce students to the design of computer algorithms, as well as analysis of sophisticated algorithms. Students will learn how to analyze the asymptotic performance of algorithms as well as provides familiarity with major algorithms and data structures. This course introduces basic methods for the design and analysis of efficient algorithms emphasizing methods useful in practice. Different algorithms for a given computational task are presented and their relative merits evaluated based on performance measures. The following important computational problems will be discussed: sorting, searching, elements of dynamic programming and greedy algorithms, advanced data structures, graph algorithms (shortest path, spanning trees, and tree traversals), string matching, elements of computational geometry, NP completeness.

UNIT WISE DETAILS

Unit Number: 1	**Title:** Introduction to Algorithms	**No. of hours: 8**

Content Summary:

Characteristics of algorithm. Analysis of algorithm: Asymptotic analysis of complexity bounds – best, average and worst-case behavior, Performance measurements of Algorithm, Time and Time and space trade- offs, Analysis of recursive algorithms through recurrence relations: Substitution method, Recursion tree method and Masters' theorem.

Unit Number: 2	**Title:** Fundamental Algorithmic Strategies	**No. of hours: 4**

Content Summary:

Brute -Force, Greedy, Dynamic Programming, Branch-and-Bound and Backtracking methodologies for the design of algorithms; Illustrations of these techniques for Problem-Solving, Bin Packing, Knap Sack. Heuristics – characteristics and their application domains. Heaps and priority queues, Hash tables and hash functions. String matching

Unit Number: 3	**Title:** Graph and Tree Algorithms	**No. of hours: 8**

Content Summary:

Traversal algorithms: Depth First Search (DFS) and Breadth First Search (BFS); Shortest path algorithms, Transitive closure, Minimum Spanning Tree, Topological sorting, Network Flow Algorithm. Graph Coloring and matching algorithms.

Unit Number: 4	**Title**: Tractable and Intractable Problems	**No. of hours: 4**

Content Summary:

Computability of Algorithms, Computability classes – P, NP, NP complete and NP-hard. Cook's theorem, Standard NP-complete problems and Reduction techniques. Advanced Topics: Approximation algorithms, Randomized algorithms, Online algorithms, Quantum algorithms.

Self-Learning Components

Container loading problem, stable marriage problem, Coin Change problem

Reference Books

1. Introduction to Algorithms, 4TH Edition, Thomas H Cormen, Charles E Lieserson, Ronald L Rivest and Clifford Stein, MIT Press/McGraw-Hill.
2. Fundamentals of Algorithms – E. Horowitz et al.

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** fundamental algorithmic concepts and how to analyze Complexities.
CO2	**Analyze** and evaluate algorithm performance.
CO3	**Design** efficient algorithms in terms of space and time.
CO4	**Apply** algorithmic problem-solving strategies.
CO5	**Develop** algorithm implementation skills.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2		P1
C02	C4		P2
CO3	C3		P3
CO4	C4		-
CO5	C6		P3

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	-	-	-	-	-	-	-	-	-	2
CO2	-	3	-	3	2	-	-	-	-	-	-	1
CO3	-	-	3	-	-	-	-	-	-	-	-	3
CO4	-	-	-	-	2	-	-	-	2	-	-	-
CO5	-	-	-	3	-	-	-	-	-	2	-	-

Justification for mapping must be relevant

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	3	-	-	1
CO2	-	3	-	3	1
CO3	3	2	-	-	1
CO4	-	3	3	-	2
CO5	-	2	3	-	2

Relevance of the Syllabus to various indicators

Unit I	**Introduction to algorithm**
Local	Addresses local understanding of the problems and how to find its solutions
Regional	Addresses regional understanding of the problems and how to find its solutions

National	Addresses national understanding of the problems and how to find its solutions
Global	Addresses global understanding of the problems and how to find its solutions
Employability	After having knowledge about how to solve real world problems, new problems can be addressed to develop their algorithms.
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills to develop analytical skills
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	Fundamental Algorithmic Strategies
Local	Understanding and applying algorithm design methodologies enhances programming and problem-solving skills at the local level.
Regional	Knowledge of algorithm design methodologies allows individuals to develop innovative solutions and potentially start their own businesses in the region
National	Adhering to ethical principles in algorithm design ensures professionalism and ethical practices at the national level.
Global	Employability: Proficiency in algorithm design techniques enhances employability opportunities globally.
Employability	Employability: Proficiency in algorithm design techniques enhances employability opportunities globally.
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills to develop analytical

	skills
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Graph and Tree Algorithms**
Local	Addresses local understanding of the problems and how to find its solutions
Regional	Addresses regional understanding of the problems and how to find its solutions
National	Addresses national understanding of the problems and how to find its solutions
Global	Addresses global understanding of the problems and how to find its solutions
Employability	After having knowledge about how to solve real world problems, new problems can be addressed to develop their algorithms.
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills to develop analytical skills
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Tractable and Intractable Problems**
Local	Addresses local understanding of the problems and how to

	find its solutions
Regional	Addresses regional understanding of the problems and how to find its solutions
National	Addresses national understanding of the problems and how to find its solutions
Global	Addresses global understanding of the problems and how to find its solutions
Employability	After having knowledge about how to solve real world problems, new problems can be addressed to develop their algorithms.
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills to develop analytical skills
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4, 8, 9
NEP 2020	-
POE/4th IR	-

ANALYSIS AND DESIGN OF ALGORITHMS LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: **Analysis and Design of Algorithms**	**Course Code**	**L-T-P**	**Credits**
	ENCS256	0-0-2	1
Type of Course:	**Programme Core**		
Pre-requisite(s), if any: - Data Structure			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	**Analyze** the time and space complexities of algorithms and evaluate their performance
CO 2	**Apply** algorithmic problem-solving strategies to solve complex computational problems
CO 3	**Design** and develop innovative algorithms for solving complex computational problems.
CO 4	**Generate** algorithmic solutions that consider trade-offs between time complexity, space complexity, and problem constraints.

Ex. No	Experiment Title	Mapped CO/COs
1	Sort a given set of elements using the Quicksort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator.	CO1
2	Design an algorithm to find the maximum and minimum elements in an unsorted array.	CO1
3	Implement Largest Common Subsequence.	CO1
4	Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.	CO1
5	Find minimum cost spanning tree of a given undirected graph using Prim’s algorithm.	CO2
6	To implement Optimal Binary Search Tree.	CO2
7	To implement Strassen’s matrix multiplication Algorithm	CO2
8	Design an algorithm to find the maximum subarray sum in an array.	CO2
9	From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.	CO2
10	Implement 0/1 Knapsack Problem using Dynamic algorithm concepts.	CO2
11	To implement Bellman Ford’s Algorithm.	CO2
12	To implement Depth First Search and Breadth First Search Algorithm.	CO2
13	To implement Naïve String-matching Algorithm.	CO3
14	Implement N Queen's problem using Back Tracking.	CO3
15	Design an algorithm to check if a given graph is acyclic (a DAG).	CO3

16	Obtain the Topological ordering of vertices in a given digraph.	CO3
17	Compute the transitive closure of a given directed graph using Warshall's algorithm	CO3
18	Design an algorithm to find the nth Fibonacci number using dynamic programming.	CO3
19	Design an algorithm to solve the 3-SAT problem using a backtracking approach.	CO4
20	Implement the brute-force algorithm to solve the Subset Sum Problem.	CO4
21	Design an algorithm to solve the Independent Set Problem using the branch and bound approach.	CO4
22	Design an algorithm to solve the Vertex Cover Problem using the 2-approximation algorithm.	CO4

DATABASE MANAGEMENT SYSTEMS

Department:	**Department of Computer Science and Engineering**		
Course Name: **Database Management System**	Course Code	L-T-P	Credits
	ENCS204	4-0-0	4
Type of Course:	Major		
Pre-requisite(s), if any: **Nil**			
Brief Syllabus: Introduction to database, Database modelling languages, E-R modelling, Transaction Processing, Database security.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction**	**No. of hours: 12**	
Content Summary: Introduction to DBMS: Database system architecture: Data Abstraction, Data Independence, Data models: network model, relational and object-oriented data models, Entity-relationship model: Relationship model, constraints, keys, Design issues, Extended E-R features- Generalization, Specialization, Aggregation, Translating E-R model into Relational model, integrity constraints in relational model.			
Unit Number: 2	**Title: Relational Query Languages**	**No. of hours: 8**	
Content Summary: Relational query languages: Relational algebra: Tuple and domain relational calculus, SQL, DDL, DML and DCL constructs. Query processing and optimization: Evaluation of relational algebra expressions, Query equivalence, Join strategies, Query optimization algorithms. Open source and Commercial DBMS - MYSQL, ORACLE, DB2, SQL server. Relational database design: Database anomalies, Domain and data			

dependency, Armstrong's axioms, Normal forms (1NF,2NF, 3NF, Boyce/Codd Normal, 4NF), Dependency preservation, Lossless design.

Unit Number: 3	Title: Transaction Processing	No. of hours: 12

Content Summary:

Storage strategies: File Organization, Indices, B-tree and B+ trees, hashing, Transaction processing: Concurrency control, ACID property, Serializability of scheduling, Locking and timestamp-based schedulers, multi-version and optimistic Concurrency Control schemes, Database recovery.

Unit Number: 4	Title: Database Security	No. of hours: 8

Database Security: Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection, SQL injection. Advanced topics: Object oriented and object relational databases, Logical databases, Web databases, Distributed databases, Data warehousing and data mining.

***SELF-LEARNING COMPONENTS:**

https://onlinecourses.nptel.ac.in/noc22_cs91/preview

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components.

Reference Books:

1."Database System Concepts", 6th Edition by Abraham Silberschatz, Henry F. Korth, S. Sudarshan, McGraw-Hill.

2. "Principles of Database and Knowledge – Base Systems", Vol 1 by J.D. Ullman, Computer Science Press.

Define Course Outcomes (CO)

COs	Statements
CO 1	**Summarize** the concepts of database objects; enforce integrity constraints on a database using RDBMS.
CO 2	**Use** Structured Query Language (SQL) for database manipulation
CO 3	**Understand** basic database storage structures and access techniques
CO 4	**Analyze** and **implement** transaction processing, concurrency control and database recovery protocols in databases.

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	-	-	-	-	-	2	-	1	-	-	-	-
CO2	-	1	1	-	-	-	1	-	1	1	1	-
CO3	2	2	-	2	2	3	2	-	-	-	-	-
CO4	-	-	-	2	2	-	-	2	1	2	2	-

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	-	-	1	-	1
CO2	-	2	2	-	1
CO3	1	-	2	-	-
CO4	1	-	3	-	-

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels (C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1	-	-
C02	C2	-	-
CO3	C3	A4	P4
CO4	C5	-	P5

Relevance of the Syllabus to various indicators

Unit I	**Introduction**
Local	Understanding the fundamentals of DBMS can aid in solving local data management challenges.
Regional	Understanding DBMS architecture can be applied to regional projects or research in data management fields.
National	DBMS plays a significant role in various national data management systems and applications.
Global	DBMS is fundamental to global data management practices and technologies.
Employability	Knowledge of DBMS enhances employability in various technical and IT professions.
Entrepreneurship	Knowledge of DBMS can inspire entrepreneurial opportunities in IT-based ventures.
Skill Development	Learning DBMS develops analytical and problem-solving skills for data management tasks.
Professional Ethics	Applying DBMS principles with integrity ensures ethical

	practices in data handling.
Gender	DBMS education is equally important for individuals of all genders pursuing IT careers.
Human Values	Applying DBMS with ethical considerations contributes to responsible data management.
Environment & Sustainability	-
Unit II	**Relational Query Languages**
Local	Understanding relational query languages can aid in solving local data retrieval challenges.
Regional	Understanding relational query languages can be applied to regional projects or research in data management.
National	Relational query languages are widely used in various national data management systems and applications.
Global	Relational query languages are fundamental to global data retrieval and processing practices.
Employability	Knowledge of relational query languages enhances employability in IT and database-related professions.
Entrepreneurship	Knowledge of relational query languages can inspire entrepreneurial opportunities in IT-based ventures.
Skill Development	Learning relational query languages develops analytical and query optimization skills.
Professional Ethics	Applying relational query languages with integrity ensures ethical data retrieval and manipulation.
Gender	Relational query languages education is equally important for individuals of all genders pursuing IT careers.
Human Values	Applying relational query languages with ethical considerations promotes user privacy and data security.
Environment & Sustainability	-

Unit III	**Transaction Processing**
Local	Understanding transaction processing aids in managing local data operations and concurrency control.
Regional	Understanding transaction processing can be applied to regional data management systems and applications.
National	Transaction processing is crucial for various national data-intensive applications and systems.
Global	Transaction processing is fundamental to global data management and processing practices.
Employability	Knowledge of transaction processing enhances employability in IT and database management professions.
Entrepreneurship	Knowledge of transaction processing can inspire entrepreneurial opportunities in data management ventures.
Skill Development	Learning transaction processing develops skills in data concurrency control and recovery mechanisms.
Professional Ethics	Applying transaction processing with integrity ensures data consistency and reliability.
Gender	Transaction processing education is equally important for individuals of all genders pursuing IT careers.
Human Values	Applying transaction processing with ethical considerations maintains data integrity and security.
Environment & Sustainability	-
Unit IV	**Database Security**
Local	Understanding database security aids in protecting local data from unauthorized access and attacks.
Regional	Understanding database security can be applied to regional data management systems and applications.
National	Database security is essential for safeguarding national data and information systems.
Global	Database security is fundamental to global data protection and privacy practices.
Employability	Knowledge of database security enhances employability in IT security and data protection professions.

Entrepreneurship	Knowledge of database security can inspire entrepreneurial opportunities in cybersecurity ventures.
Skill Development	Learning database security develops skills in access control and intrusion detection.
Professional Ethics	Applying database security with integrity ensures ethical data protection and privacy.
Gender	Database security education is equally important for individuals of all genders pursuing IT security careers.
Human Values	Applying database security with ethical considerations promotes data confidentiality and integrity.
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	-

DATABASE MANAGEMENT SYSTEMS LAB

Department:	Department of Computer Science and Engineering		
Course Name:	**Course Code**	L-T-P	Credits
Database Management System Lab	**ENCS254**	0-0-2	1
Type of Course:	Major		

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Apply the basic concepts of Database Systems and create model using ER Diagrams
CO 2	Understand the basics of SQL and construct queries for database creation and interaction
CO 3	Understand PL/SQL statements: Exception Handling, Cursors, and Triggers
CO 4	Analyse and implement Join operations to extract information from more than one table.

Ex. No.	**Experiment Title**	**Mapped CO/COs**
1	Consider following databases and draw ER diagram and convert entities and relationships to relation table for a given scenario: COLLEGE DATABASE: STUDENT (USN, SName, Address, Phone, Gender) SEMSEC (SSID, Sem, Sec) CLASS (USN,	CO1

	SSID) SUBJECT (Subcode, Title, Sem, Credits) IAMARKS (USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)	
2	Consider following databases and draw ER diagram and convert entities and relationships to relation table for a given scenario: COMPANY DATABASE: EMPLOYEE (SSN, Name, Address, Sex, Salary, SuperSSN, DNo) DEPARTMENT (DNo, DName, MgrSSN, MgrStartDate) DLOCATION (DNo,DLoc) PROJECT (PNo, PName, PLocation, DNo) WORKS_ON (SSN, PNo, Hours)	CO1
3	Consider the below Database: Movies (title, director, making_year, rating), actors (actor, acting_year), acts(actor, title), directors (director, director_year) Write relation algebra queries for given relations: 1. Find movies made after 1997 2. Find movies made by Hanson after 1997 3. Find all movies and their ratings 4. Find all actors and directors Find Coen's movies with McDormand	CO2
4	**Database Schema for a customer-sale scenario** Customer(**Cust id : integer,** cust_name: string) Item(**item_id: integer,** item_name: string, price: integer) Sale(**bill_no: integer,** bill_data: date, **cust_id: integer, item_id: integer**, qty_sold: integer) For the above schema, perform the following— i. Create the tables with the appropriate integrity constraints. ii. Insert around 10 records in each of the tables. iii. List all the bills for the current date with the customer names and item numbers. iv. List the total Bill details with the quantity sold, price of the item and the final amount. v. List the details of the customer who have bought a product which has a price>200. vi. Give a count of how many products have been bought by each customer vii. Give a list of products bought by a customer having cust_id as 5. viii. List the item details which are sold as of today. ix. Create a view which lists out the bill_no, bill_date, cust_id, item_id, price, qty_sold, amount. Create a view which lists the daily sales date wise for the	CO2

last one week

<table>
<tr>
<td>5</td>
<td>Consider the following table:
Table: CLASS
<table>
<tr><td>Id</td><td>Name</td></tr>
<tr><td>1</td><td>Bravo</td></tr>
<tr><td>2</td><td>Alex</td></tr>
<tr><td>4</td><td>Cheng</td></tr>
</table>
Give the output of the following SQL script:
> INSERT INTO class VALUES (5,'Rahul');
> COMMIT;
> UPDATE class SET name = 'Abhijeet' WHERE id= '5';
> SAVEPOINT A;
> INSERT INTO class VALUES (6, 'Chris');
> SAVEPOINT B;
> INSERT INTO class VALUES (7, 'Bravo');
> SAVEPOINT C
> SELECT * FROM class;
> ROLLBACK TO B;
> SELECT * FROM class;
> ROLLBACK TO A;</td>
<td>CO2</td>
</tr>
<tr>
<td>6</td>
<td>(Exercise on retrieving records from the table) EMPLOYEES (Employee_Id, First_Name, Last_Name, Email, Phone_Number, Hire_Date, Job_Id, Salary, Commission_Pct, Manager_Id, Department_Id)
(a) Find out the employee id, names, salaries of all the employees
(b) List out the employees who works under manager 100
(c) Find the names of the employees who have a salary greater than or equal to 4800
(d) List out the employees whose last name is 'AUSTIN'
(e) Find the names of the employees who works in departments 60,70 and 80
(f) Display the unique Manager_Id.</td>
<td>CO2</td>
</tr>
<tr>
<td>7</td>
<td>(Exercise on updating records in table) Create Client_master with the following fields(ClientNO, Name, Address, City, State, bal_due)
(a) Insert five records
(b) Find the names of clients whose bal_due> 5000 .
(c) Change the bal_due of ClientNO " C123" to Rs. 5100
(d) Change the name of Client_master to Client12.
(e) Display the bal_due heading as "BALANCE"</td>
<td>CO2</td>
</tr>
</table>

8	Rollback and Commit commands Create Teacher table with the following fields(Name, DeptNo, Date of joining, DeptName, Location, Salary) (a) Insert five records (b) Give Increment of 25% salary for Mathematics Department . (c) Perform Rollback command (d) Give Increment of 15% salary for Commerce Department (e) Perform commit command	CO2
9	(Exercise on order by and group by clauses) Create Sales table with the following fields(Sales No, Salesname, Branch, Salesamount, DOB) (a) Insert five records (b) Calculate total salesamount in each branch (c) Calculate average salesamount in each branch (d) Display all the salesmen, DOB who are born in the month of December as day in character format i.e. 21-Dec-09 (e) Display the name and DOB of salesman in alphabetical order of the month.	CO2
10	Consider the following tables namely "DEPARTMENTS" and "EMPLOYEES" Their schemas are as follows, Departments (dept _no , dept_ name , dept_location); Employees (emp_id , emp_name , emp_salary,dept_no); a) Develop a query to grant all privileges of employees table into departments table b) Develop a query to grant some privileges of employees table into departments table c) Develop a query to revoke all privileges of employees table from departments table d) Develop a query to revoke some privileges of employees table from departments table e) Write a query to implement the save point.	CO2
11	Using the tables "DEPARTMENTS" and "EMPLOYEES" perform the following queries a) Display the employee details, departments that the departments are same in both the emp and dept. b) Display the employee name and Department name by implementing a left outer join. c) Display the employee name and Department name by implementing a right outer join. d) Display the details of those who draw the salary greater than the average salary	CO4

12	Employee Database An Enterprise wishes to maintain a database to automate its operations. Enterprise is divided into certain departments and each department consists of employees. The following two tables describes the automation schemas Dept (deptno, dname, loc) Emp (empno, ename, job, mgr, hiredate, sal, comm, deptno) a) Update the employee salary by 15%, whose experience is greater than 10 years. b) Delete the employees, who completed 30 years of service. c) Display the manager who is having maximum number of employees working under him? d) Create a view, which contain employee names and their manager	CO2
13	Using Employee Database perform the following queries a) Determine the names of employee, who earn more than their managers. b) Determine the names of employees, who take highest salary in their departments. c) Determine the employees, who are located at the same place. d) Determine the employees, whose total salary is like the minimum Salary of any department. e) Determine the department which does not contain any employees.	CO2
14	Write a PL/SQL program to demonstrate Exceptions.	CO3
15	Write a PL/SQL program to demonstrate Cursors.	CO3
16	Write a PL/SQL program to demonstrate Functions.	CO3
17	Write a PL/SQL program to demonstrate Packages.	CO3
18	Write PL/SQL queries to create Procedures.	CO3
19	Write PL/SQL queries to create Triggers.	CO3

TEST AUTOMATION

Department:	**Department of Computer Science and Engineering**		
Course Name: **Test Automation**	**Course Code**	**L-T-P**	**Credits**
	ENSP208	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any: nil			

Brief Syllabus:

This course material includes theory and lab. The course comprises five modules. The main objective of this course is to impart knowledge on Test Automation and Working with Selenium. The course details out the testing life cycle and essentials of test design.

UNIT WISE DETAILS

Unit Number: 1	**Title: Introduction to Software Testing**	**No. of hours: 12**

Content Summary:

Seven principles of Software Testing, SDLC vs STLC, Testing Life Cycle, Usability Testing, Why do we need Usability Testing, How to do Usability testing, Advantages & Disadvantages, Functional Testing, End to End Testing,Methods, Advantages & Disadvantages, Compatibility Testing, Types GUI testing, Techniques API testing, Advantages.

Unit Number: 2	**Title: Test Automation**	**No. of hours: 6**

Content Summary:

Selenium: Selenium components, Selenium Architecture, TestNGInstalling TestNg in Eclipse, TestNG annotations – Understanding usage, Setting priority of execution for test cases, Hard Assertion, Soft Assertion, TestNG Reports

Unit Number: 3	**Title: Selenium 3.x**	**No. of hours: 8**

Content Summary:

Describe Selenium 3.x advantages and implementation, Define drivers for Firefox, IE, chrome, Iphone, Android etc, Analyse first Selenium Code, Differentiate between Close and Quit, Describe Firepath and firebug Add-ons installation in Mozilla, Inspect elements in Mozilla, Chrome and IE, Identifying WebElements using id, name, class, Generate own CssSelectors. Differentiate between performance of CssSelectors as compared to Xpaths, Define class attribute, Handle Dynamic objects/ids on the page, Analyse whether object is present on page or not.

Unit Number: 4	**Title: Manual Testing**	**No. of hours: 6**

Content Summary:

Manual Testing, Manual Testing – How to Approach?, Manual Testing – Myth and fallacy, Defect Life Cycle, Qualities of a good Manual Tester, Manual Testing vs Automation Testing, Types, System Testing, Acceptance Testing, Unit Testing, Techniques, Integration Testing, Smoke- Sanity Testing.

Unit Number: 5	**Title: Introduction to Test Design**	**No. of hours: 8**

Content Summary:

Test Scenario, Test Case Design, Test Basis Traceability Matrix

***Self-Learning Components:**

- Anomaly detection techniques to identify unusual behaviour or deviations from expected outcomes
- Integrate human feedback into automated testing process.
- Version control for test scripts and machine learning models

Reference Books:

1. Kamber and Han, "Data Mining Concepts and Techniques", Hartcourt India P.Ltd
2. W. H. Inmon, "Building the operational data store", 2nd Ed., John Wiley.
3. Paul Raj Poonia, "Fundamentals of Data Warehousing", John Wiley & Sons.
4. Sam Anahony, "Data Warehousing in the real world: A practical guide for building decision support systems", John Wiley.

Define Course Outcomes (CO)

COs	
CO1	Understand the fundamental principles of software testing and differentiate between Software Development Life Cycle (SDLC) and Software Testing Life Cycle (STLC).
CO2	**Demonstrate** knowledge of various types of testing, including Usability Testing, Functional Testing, End-to-End Testing, Compatibility Testing, GUI Testing, and API Testing, along with their advantages and disadvantages.
CO3	**Gain** proficiency in test automation using Selenium, including an understanding of Selenium components, architecture, and TestNG framework for test automation.
CO4	**Implement** Selenium 3.x for web automation, including setting up drivers for different browsers and mobile devices, inspecting elements, using CSS selectors, and handling dynamic objects on web pages.
CO5	**Design** effective and comprehensive test scenarios and test cases, and understand the importance of test basis and traceability matrix in test design.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels (C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C4	A1	P4
CO2	C3	A1	P2, P3
CO3	C4	A2	P3
CO4	C1	A5	P3, P4
CO5	C6	A5	P3

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	-	-	-	-	-	-	-	-	-	-	2
CO2	2	1	-	3	-	-	-	-	-	-	-	2
CO3	-	-	2	-	3	-	-	-	-	-	-	2
CO4	2	2	3	-	-	-	-	-	-	-	-	2
CO5	2	3	3	-	-	-	-	-	-	-	-	2

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	-	-	1	-
CO2	3	-	-	1	-
CO3	3	-	-	-	1
CO4	3	-	-	-	1
CO5	2	-	2	2	1

Relevance of the Syllabus to various indicators

Unit I	Introduction to Software Testing
Local	-
Regional	-
National	-
Global	-

Employability	Knowledge of software testing principles, methods, and techniques enhances employability for software testing roles.
Entrepreneurship	Understanding SDLC vs STLC and Testing Life Cycle equips individuals with skills valuable for entrepreneurship, especially in software development and quality assurance ventures.
Skill Development	The syllabus covers a wide range of testing methods and techniques, contributing to the development of diverse technical skills.
Professional Ethics	The syllabus emphasizes the importance of ensuring software quality, which aligns with professional ethics in delivering reliable and trustworthy software solutions.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Test Automation: Selenium**
Local	-
Regional	-
National	-
Global	Knowledge of Selenium is applicable in software testing at different geographical levels, from local to global, across various industries.
Employability	Knowledge of Selenium and TestNG enhances employability in software testing and quality assurance roles.
Entrepreneurship	Understanding TestNG annotations and reports equips individuals with skills valuable for entrepreneurship, especially in software development and automation ventures.
Skill Development	Learning Selenium components, architecture, and TestNG

	usage contributes to the development of practical skills in software testing and test automation.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	Introduction to Selenium 3.x
Local	-
Regional	-
National	-
Global	The ability to identify web elements is essential for software testing and quality assurance worldwide.
Employability	Proficiency in Selenium and web element identification enhances employability in software testing and automation roles.
Entrepreneurship	Knowledge of these topics supports entrepreneurship by enabling individuals to develop and offer automated testing solutions.
Skill Development	Develops knowledge and skills in data handling & map associated use cases.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	Manual Testing
Local	-

Regional	-
National	-
Global	-
Employability	Proficiency in manual testing techniques enhances employability in quality assurance and testing roles.
Entrepreneurship	Expertise in manual testing can support entrepreneurship by offering testing services to various businesses.
Skill Development	Learning manual testing techniques and methodologies contributes to practical skill development in software testing and quality assurance.
Professional Ethics	Following proper testing methodologies, understanding defect life cycle, and ensuring accurate reporting align with professional ethics by promoting software quality.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit V	Introduction to Test Design
Local	
Regional	
National	
Global	
Employability	
Entrepreneurship	
Skill Development	
Professional Ethics	Adhering to proper test scenario and test case design methodologies aligns with professional ethics by promoting software quality.

Gender	
Human Values	
Environment & Sustainability	
SDG	The course on software testing and test automation indirectly contributes to the United Nations' Sustainable Development Goals (SDGs) by promoting quality education (Goal 4) through equipping students with essential skills in the software development field, thereby enhancing employability and supporting decent work and economic growth (Goal 8). Moreover, the course encourages innovation and improvements in software industry practices, aligning with the SDG of industry, innovation, and infrastructure (Goal 9).
NEP 2020	The above course on software testing and test automation aligns with the National Education Policy (NEP) of India by promoting skill development and vocational education. By offering practical training in software testing principles and test automation techniques, the course contributes to equipping students with industry-relevant skills, fostering employability, and supporting the NEP's vision of providing quality and holistic education that prepares learners for real-world challenges. Moreover, the emphasis on practical learning and hands-on experience in the course aligns with the NEP's focus on experiential learning and promoting critical thinking and problem-solving abilities among students, preparing them for the demands of the modern workforce and contributing to the overall development of the country's education system.

TEST AUTOMATION LAB

Department:	Department of Computer Science and Engineering		
Course Name: **Test Automation Lab**	**Course Code**	L-T-P	Credits
	ENSP258	0-0-2	1
Type of Course:	Minor		

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Describe the process of selecting appropriate test cases for automation.
CO 2	Apply exception handling techniques in automated test scripts.
CO 3	Propose strategies to integrate test automation into the software development life cycle.
CO 4	Design and implement end-to-end automated test suites for different application modules.

Ex. No	Experiment Title	Mapped CO/COs
1	Understanding the need for test automation.	CO1
2	Introduction to popular test automation tools (e.g., Selenium, Appium, JUnit, TestNG). Installation and setup of selected test automation tools.	CO1
3	Creating and implementing an automation framework (e.g., Page Object Model, Data-Driven Framework, Keyword-	CO1

	Driven Framework)	
4	Writing basic scripts to automate test cases in scripting language for automation (e.g., Java, Python, C#).	CO1
5	Automating UI testing using Selenium WebDriver and handling different types of web elements (e.g., buttons, forms, dropdowns)	CO2
6	Writing scripts for mobile app testing on Android and iOS platforms	CO2
7	Implementing assertions to validate expected outcomes using verification techniques to compare actual and expected results.	CO2
8	Dealing with AJAX-based updates on web pages and techniques for handling dynamic IDs, XPath, and CSS selectors	CO2
9	Implementing synchronization to handle timing issues	CO2
10	Creating detailed test reports using frameworks (e.g., TestNG, ExtentReports). Logging test execution activities and errors.	CO2
11	Integrating test automation with CI/CD pipelines (e.g., Jenkins).	CO2
12	Using version control systems (e.g., Git) for test scripts.	CO2
13	Refactoring automation code for better maintainability.	CO3
14	Writing reusable and modular test scripts.	CO3
15	Addressing challenges such as synchronization issues, dynamic content, and cross-browser testing.	CO3
16	Applying test automation concepts to a real project.	CO3
17	Designing and automating end-to-end test scenarios.	CO3
18	Conducting code reviews for automation scripts.	CO3
19	Analyzing real-world case studies of successful test automation implementations.	CO4

20	Working on hands-on projects to apply learned concepts.	CO4

LIFE SKILLS FOR PROFESSIONALS-II

<table>
<tr><td>Department:</td><td colspan="3">Department of Computer Science and Engineering</td></tr>
<tr><td rowspan="2">Course Name:

Life Skills for Professionals - II</td><td>Course Code</td><td>L-T-P</td><td>Credits</td></tr>
<tr><td>AEC012</td><td>3-0-0</td><td>3</td></tr>
<tr><td>Type of Course:</td><td colspan="3">AEC</td></tr>
<tr><td colspan="4">Pre-requisite(s), if any:</td></tr>
<tr><td colspan="4">Brief Syllabus:

This course is a multifaceted initiative designed to enhance and optimize learner’s communication practices across various platforms. This program integrates a range of strategies, tools, and techniques to foster effective communication, facilitate collaboration, and promote a cohesive information flow within the learner’s area. This course is structured and comprehensive initiative designed to develop and improve individuals' aptitude across various cognitive and behavioral domains. This course incorporates a range of assessments, training modules, and activities to enhance critical thinking, problem-solving, decision-making, and other essential aptitudes required for personal and professional success.</td></tr>
<tr><td colspan="4">UNIT WISE DETAILS</td></tr>
<tr><td>Unit Number: 1</td><td colspan="2">Title: Personality Improvement</td><td>No. of hours: 6</td></tr>
<tr><td colspan="4">Content Summary: Asking for and giving information, Offering and responding to offers, Requesting and responding to requests, Congratulating people on their success, Asking questions and responding politely, Apologizing and forgiving</td></tr>
<tr><td>Unit Number: 2</td><td colspan="2">Title: Ratio & its application</td><td>No. of hours: 6</td></tr>
<tr><td colspan="4">Content Summary: Time & Work, Time & Distance, Train, Boat & Stream, Permutation & combination, Probability</td></tr>
<tr><td>Unit</td><td colspan="2">Title: Arithmetic</td><td>No. of hours: 6</td></tr>
</table>

Number: 3		
Content Summary: Inequalities, Log, progression, Mensuration, BODMAS		
Unit Number: 4	**Title: Presentation Skills**	**No. of hours: 6**
Content Summary: Presentation Skills, Telephone etiquettes, LinkedIn Profile and professional networking, Video resumes & Mock interview sessions.		
Unit Number: 5	**Title: Leadership skills**	**No. of hours: 6**
Content Summary: Nurturing future leaders, increasing productivity of the workforce, Imparting Self-leadership, Executive leadership.		
***SELF-LEARNING COMPONENTS:** **https://onlinecourses.nptel.ac.in/noc21_hs02/preview** **Please Note:** 1)Students are supposed to learn the components on self-basis 2) At least 5-10 % syllabus will be asked in end term exams from self-learning components.		
Reference Books: Aggarwal, R. S. (2014). Quantitative aptitude (Revised edition). Gladwell, M. (2021). Talking to strangers. Scott, S. (2004). Fierce conversations.		

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** and apply the fundamental theories, models, and principles of communication.
CO2	**Apply** ability to communicate effectively through spoken and written forms. It includes developing skills in public speaking, interpersonal communication, professional writing, and persuasive communication.

CO3	**Evaluate** the development of teamwork and collaboration skills. It includes activities such as group projects, team-building exercises, and simulations that allow students to practice effective communication and collaboration within diverse teams
CO4	**Improve** their communication skills in different professional and personal contexts, such as interviews, networking events, customer interactions, and interpersonal relationships
CO5	**Analyze** ideas and information clearly and concisely through spoken language. They will develop the ability to articulate their thoughts, use appropriate vocabulary, and convey their message with clarity.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	-	-
C02	C3	-	-
CO3	C5	-	-
CO4	-	-	P5
CO5	C5	-	P5

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 1	3	3	-	-	-	-	-	-	-	-	3	3
CO 2	3	2	-	-	-	-	-	-	-	-		3
CO 3	-	2	3	-	-	2	-	-	-	-	-	2

CO 4	-	3	2	-	-	-	-	2	-	-	-	3
CO 5	-	-	3	2	-	-	-		-	-	-	3

Justification for mapping must be relevant.

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO 1	3	2	-	-	1
CO 2		2	-	-	-
CO 3	1	-	-	3	1
CO 4		2	-	-	-
CO 5	-	-	3	-	1

Relevance of the Syllabus to various indicators

Unit I	**Personality Improvement**
Local	Improve personality, enhance basic communication skills.
Regional	Recognize the importance of continuous learning and practice to maintain and further develop interpersonal ability.
National	Practice leadership strategies for solving problems within time constraints, as in competitive exams.
Global	Aligns with global trends in employment
Employability	Develop skills in real-life situations, such as academic exams, job interviews, and problem-solving scenarios.
Entrepreneurship	Learn to share ideas, listen to others, build consensus, and manage conflicts to achieve common goals in collaborative settings.

Skill Development	Develops Skills in public speaking, interpersonal communication, professional writing, and persuasive communication.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Ratio and it's proportion**
Local	Recognize the importance of continuous learning and practice to maintain and further develop mental ability.
Regional	Practice attentive listening techniques, such as paraphrasing and asking clarifying questions.
National	Attentively listen to others, understand their perspectives, and respond appropriately.
Global	Aligns with global trends in employment
Employability	Develop skills in participating and contributing to group discussions, meetings, or presentations.
Entrepreneurship	Learn to share ideas, listen to others, build consensus, and manage conflicts to achieve common goals in collaborative settings.
Skill Development	Apply skills in real-life situations, such as academic exams, job interviews, and problem-solving scenarios.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Arithmetic**
Local	Improve number sense, enhance basic arithmetic skills and strengthen mental math abilities and speed.
Regional	-

National	Learn about Inequalities, Log, progression, Mensuration, BODMAS
Global	Recognize the importance of continuous learning.
Employability	Develop skills in participating and contributing to group discussions, meetings, or presentations.
Entrepreneurship	-
Skill Development	Recognize the importance of continuous learning and practice to maintain and further develop mental ability.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Presentation Skills**
Local	Attentively listen to others, understand their perspectives, and respond appropriately with timelines
Regional	-
National	Contributes to develop skill and improved productivity
Global	Aligns with global trends in understanding the deadlines.
Employability	Enhance the employability of individuals by developing essential skills and competencies sought by employers
Entrepreneurship	-
Skill Development	Strengthening critical thinking, problem-solving, memory, and other cognitive functions to improve overall mental agility and performance.
Professional Ethics	-
Gender	-
Human Values	-

Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	Aligns with the concepts employability

MINOR PROJECT-I

Department:	**Department of Computer Science and Engineering**		
Course Name: Minor Project-I	**Course Code**	**L-T-P**	**Credits**
	ENSI252	---	2
Type of Course:	Project		
Pre-requisite(s), if any: NA			

- Students expected to develop a basic project that demonstrates the application of learning from studied subjects.
- Students are required to submit a hard copy of project file as per the template (Provided at the end of Handbook). File needs to be submitted in spiral bind.
- Project will be evaluated on the scale of 100 with following evaluation criteria.
  - Project idea & features (10)
  - Literature review (10)
  - Tools & Techniques employed (10)
  - Methodology (10)
  - Presentation of Results and its usefulness (20)
  - Implementation and its understandability (10)
  - Meetings & comments by guide (20)
  - Research paper (10)

File format for Minor project

	Abstract	Page No.
1.	Introduction (description of broad topic)	
2.	Motivation	
3.	Literature Review	
4.	Gap Analysis	
5.	Problem Statement	

6.	Objectives	
7.	Tools/platform used	
8.	Methodology	
9.	Experimental Setup	
10.	Evaluation Metrics	
11.	Results And Discussion	
12.	Conclusion & Future Work	
13.	References	
14.	Annexure I: Responsibility Chart	
15.	Annexure II: Screenshots of all the MS-Team Meetings with links (online)/ handwritten comments(offline) from guide	
16.	Annexure III Complete implementation code	
17.	Annexure IV Research Paper (Published/Submitted)	

Semester: 5

THEORY OF COMPUTATION

<table>
<tr><td>Department:</td><td colspan="3">Department of Computer Science and Engineering</td></tr>
<tr><td rowspan="2">Course Name: Theory of Computation</td><td>Course Code</td><td>L-T-P</td><td>Credits</td></tr>
<tr><td>ENCS301</td><td>3-1-0</td><td>4</td></tr>
<tr><td>Type of Course:</td><td colspan="3">Major</td></tr>
<tr><td colspan="4">Pre-requisite(s), if any:</td></tr>
<tr><td colspan="4">Brief Syllabus:

This course provides a formal connection between algorithmic problem solving and the theory of languages and automata and develops them into a mathematical view towards algorithmic design and in general computation itself. The course should in addition clarify the practical view towards the applications of these ideas in the engineering part of computer science.</td></tr>
<tr><td>Unit Number: 1</td><td colspan="2">Title: Introduction to Finite automata</td><td>No. of hours: 12</td></tr>
<tr><td colspan="4">Content Summary:

Finite automata: Review of Automata, its types and regular expressions, Equivalence of NFA, DFA and €-NFA, Conversion of automata and regular expression, Applications of Finite Automata to lexical analysis</td></tr>
<tr><td>Unit Number: 2</td><td colspan="2">Title: PDA and Parser</td><td>No. of hours: 10</td></tr>
<tr><td colspan="4">Content Summary:

PDA and Parser: Parse Trees, Ambiguity in grammars and languages, Push down automata, Context Free grammars, Top down and Bottom up parsing. Closure Properties of CFL.</td></tr>
<tr><td>Unit Number: 3</td><td colspan="2">Title: Chomsky hierarchy and Turing Machine</td><td>No. of hours: 08</td></tr>
</table>

Content Summary:

Chomsky hierarchy and Turing Machine: Chomsky hierarchy of languages and recognizers, Context Sensitive features like type checking, Turing Machine as language acceptors and its design.

Unit Number: 4	Title: Code generation and optimization	No. of hours: 10

Content Summary:

Code generation and optimization: Syntax directed translation, S-attributed and L-attributed grammars, Intermediate code generation, type conversions, and equivalence of type expression, Code generation and optimization.

Text Books

1. J.E. Hopcroft, R. Motwani and J.D. Ullman, "Introduction to Automata Theory, Languages and Computations", second Edition, Pearson Education.

Reference Books/Materials

1. H.R. Lewis and C.H. Papadimitriou, "Elements of the theory of Computation", Second Edition, Pearson Education.

2. Thomas A. Sudkamp," An Introduction to the Theory of Computer Science, Languages and Machines", Third Edition, Pearson Education.

3. Raymond Greenlaw an H.James Hoover, "Fundamentals of Theory of Computation, Principles and Practice", Morgan Kaufmann Publishers.

4. MichealSipser, "Introduction of the Theory and Computation", Thomson Brokecole.

5. J. Martin, "Introduction to Languages and the Theory of computation" Third Edition, Tata Mc Graw Hill.

Define Course Outcomes (CO)

COs	Statements
CO1	To solve the problems related to regular expression, regular grammar, and Finite Automata
CO2	To write a formal notation for strings, languages and machines
CO3	To identify the phases of compilers for a programming language and construct the parsing table for a given syntax
CO4	To discover syntax directed translation rules for a given context free grammar by examining S-attributed and L-attributed grammars
CO5	To construct grammars and machines for a context free and context sensitive languages
CO6	To build the intermediate code by applying various code optimization strategies.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C3	A2	P3
CO2	C2	A1	P4
CO3	C4	A2	P3
CO4	C4	A4	P4
CO5	C5	A3	P2
CO6	C6	A5	P5

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	-	-	-	-	-	-	-	-	-	1
CO2	2	2	-	-	-	-	-	-	-	-	-	1
CO3	2	3	3	3	-	-	3	-	-	-	-	1
CO4	3	3	3	3	-	-	3	3	3	-	-	1
CO5	1	-	-	2	3		1	-	-	-	-	1
CO6	-	-	-	-	-	3	-	-	-	-	-	1

Justification for mapping must be relevant

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	1	2	1	-
CO2	3	2	2	2	-
CO3	3	3	2	2	-
CO4	2	3	2	3	1
CO5	3	3	2	3	1
CO6	2	3	2	3	1

Relevance of the Syllabus to various indicators

Unit I	**Introduction to formal proof**
Local	-
Regional	-
National	-

Global	In the global context, formal proof and finite automata have significant relevance as they are fundamental concepts in computer science and mathematics. The global technology industry heavily relies on formal proof techniques for developing secure software systems, cryptography, and algorithmic design.
Employability	-
Entrepreneurship	-
Skill Development	Studying formal proof and finite automata develops critical thinking, logical reasoning, and problem-solving skills. These skills are transferable and applicable to various domains beyond computer science, including mathematics, engineering, and natural sciences. They enhance overall skill development and promote a deeper understanding of computational processes and structures.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Regular Expression**
Local	-
Regional	-
National	-
Global	In the global context, regular expressions and automata have significant relevance as they are fundamental concepts in computer science and information processing.
Employability	-
Entrepreneurship	-
Skill Development	Studying regular expressions and automata develops critical thinking, problem-solving, and logical reasoning skills. These

	skills are transferable and applicable to various domains beyond computer science, including linguistics, mathematics, and data analysis.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Context-Free Grammar (CFG)**
Local	-
Regional	-
National	-
Global	In the global context, context-free grammars and automata play a significant role in programming language design, parsing algorithms, and language translation tools.
Employability	-
Entrepreneurship	-
Skill Development	Studying context-free grammars and automata develops critical thinking, problem-solving, and algorithmic design skills. These skills are transferable and applicable to various domains beyond computer science, such as linguistics, mathematics, and formal reasoning.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**A language that is not Recursively Enumerable (RE)**

Local	-
Regional	-
National	-
Global	The study of languages that are not recursively enumerable (RE) has global significance as it forms the basis of theoretical computer science. Researchers and academics worldwide collaborate to explore undecidable problems, develop new mathematical models, and advance the understanding of computability.
Employability	-
Entrepreneurship	-
Skill Development	Studying languages that are not recursively enumerable (RE) enhances critical thinking, problem-solving, and analytical skills. The exploration of undecidable problems requires creativity, logical reasoning, and the ability to work with complex mathematical concepts.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4, 9
NEP 2020	Context-free grammars and automata align with NEP 2020 in the following ways: Integration of Emerging Technologies, Multidisciplinary Approach and Skill Development.
POE/4th IR	Context-free grammars and automata contribute to the Fourth Industrial Revolution (IR 4.0) in the following ways: Advancements in Computing, Data Processing and Analysis, Technological Disruption and Interdisciplinary Collaboration.

OPERATING SYSTEMS

Department:	**Department of Computer Science and Engineering**		
Course Name: OPERATING SYSTEM	**Course Code**	**L-T-P**	**Credits**
	ENCS303	4-0-0	4
Type of Course:	Major		
Pre-requisite(s), if any: Basics of programming			

Brief Syllabus:

The Operating systems course is intended as a general introduction to the techniques used to implement operating systems and related kinds of systems software. The topics covered will be functions and structure of operating systems, process management (creation, synchronization, and communication); processor scheduling; deadlock prevention, avoidance, and recovery; main-memory management; virtual memory management (swapping, paging, segmentation and page-replacement algorithms); control of disks and other input/output devices; file-system structure and implementation; and protection and security.

UNIT WISE DETAILS

Unit Number: 1	**Title: Introduction to Operating System and Process Scheduling**	**No. of hours: 8**

Content Summary:

Introduction to Operating System: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Functions of an Operating System Structure of an OS-Layered, Monolithic, Microkernel Operating Systems, Concept of Virtual Machine.

The Abstraction: The Processes definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching.

Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads.

Process Scheduling: Foundation and Scheduling objectives, Types of Schedulers, Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time.

Scheduling Algorithms: Pre-emptive and Non-preemptive, FCFS, SJF, RR.

Unit Number: 2	Title: Memory & File Management	No. of hours: 12

Content Summary:

Memory Management: Basic concept, Logical and Physical address map, Memory allocation: Contiguous Memory allocation – Fixed and variable partition–Internal and External fragmentation and Compaction; Paging: Principle of operation – Page allocation – Hardware support for paging, Protection and sharing, Disadvantages of paging.

Virtual Memory: Basics of Virtual Memory – Hardware and control structures – Locality of reference, Page fault, Working Set, Dirty page/Dirty bit– Demand paging, Page Replacement algorithms: Optimal, First in First Out (FIFO), Second Chance (SC), Not recently used (NRU) and Least Recently used (LRU).

Unit Number: 3	Title: Process-Synchronization, Deadlocks & I/O Systems	No. of hours: 10

Content Summary:

Process-Synchronization & Deadlocks: Inter-process Communication: Critical Section, Race Conditions, Mutual Exclusion, Peterson's Solution, The Producer\ Consumer Problem, Semaphores, Event Counters, Monitors, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem etc.

Deadlocks: Definition of Deadlocks, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery. Memory and I/O Management: Introduction Memory Allocation Techniques: Fragmentation, Segmentation.

I/O Systems: I/O devices, Device controllers, Direct memory access Principles of I/O Software: Goals of Interrupt handlers.

Unit Number: 4	Title: Distributed Operating Systems & Concurrent System	No. of hours: 10

Content Summary:

Distributed Operating Systems: Introduction, Issues, Communication Primitives, Distributed Deadlock Detection, Issues, Centralized Deadlock-Detection Algorithms Distributed Deadlock-Detection Algorithms. Agreement Protocols, Classification-Solutions, Applications. Distributed Resource Management: Distributed File systems, Architecture, Mechanisms, Design Issues, Distributed Shared Memory, Architecture, Algorithm, Protocols-Design Issues. Distributed Scheduling, Issues, Components,

Algorithms.

Concurrent System: Failure Recovery and Fault Tolerance: Basic Concepts-Classification of Failures, Basic Approaches to Recovery; Recovery in Concurrent System; Synchronous and Asynchronous Check-pointing and Recovery; Check pointing in Distributed Database Systems; Fault Tolerance.

***Self-Learning Components:**

1. Case study on UNIX and WINDOWS Operating System.

2. Practice of System calls

3. Students can refer the following book as well:

Operating Systems: Three Easy Pieces by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau

https://pages.cs.wisc.edu/~remzi/OSTEP/

4. Students can refer the following courses as per the Open-Source University Curriculum

- "Operating system courses" on Udemy.
- "Introduction to Operating Systems Specialization" Coursera.
- "Introduction to Operating Systems" by Udacity.

Reference Books:

1. MukeshSinghal and N. G. Shivaratri, "Advanced Concepts in Operating Systems", McGrawHill, 2000
2. Abraham Silberschatz, Peter B. Galvin, G. Gagne, "Operating System Concepts", Sixth Addison Wesley Publishing Co., 2003.
3. Andrew S. Tanenbaum, "Modern Operating Systems", Second Edition, Addison Wesley, 2001.
4. Tannenbaum, "Operating Systems", PHI, 4th Edition.
5. William Stallings, "Operating Systems Internals and Design Principles", PHI

Define Course Outcomes (CO)

COs	Statements
CO1	Recall and explain the fundamental concepts and principles of operating systems.
CO2	Compare and contrast different types of operating systems, their architectures, and their services.
CO3	Apply knowledge of process management and scheduling algorithms to solve problems.
CO4	Evaluate the performance of scheduling algorithms and memory management techniques by analyzing system metrics, such as CPU utilization, throughput, turnaround time, waiting time, and response time.
CO5	Design and create programs to simulate file management, virtual memory, and distributed operating systems concepts.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A1	P4
CO2	C4	A3	P4
CO3	C3	A4	P2
CO4	C5	A4	P3

CO5	C6	A5	P5

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	-	-	-	-	-	-	1	-	-	1
CO2	3	2	-	-	-	1	-	-	-	-	-	1
CO3	3	3	2	-	-	-	-	-	-	-	-	1
CO4	-	3	-	3	-	-	-	-	-	2	-	-
CO5	3	-	3	-	2	-	-	-	1	-	-	1

Justification for mapping must be relevant

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	1	2	1	1
CO2	3	2	2	2	1
CO3	3	3	2	2	-
CO4	2	3	2	3	-
CO5	3	3	2	3	1

Relevance of the Syllabus to various indicators

Unit I	**Introduction to Operating System and Process Scheduling**
Local	Can help students to build a strong foundation in computer science.

Regional	
National	Widely used across industries and organizations
Global	Applicable in various global industries and organizations.
Employability	Covers essential concepts and skills related to operating systems.
Entrepreneurship	Understanding of operating systems can be beneficial for entrepreneurs in the technology industry.
Skill Development	Students will develop skills in understanding operating system concepts, system calls, and kernel functionalities
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Memory & File Management**
Local	-
Regional	To meet the demand for skilled professionals in the region.
National	It provides fundamental knowledge about processes, threads, and process scheduling, which are essential for the functioning of computer systems in various national industries and organizations.
Global	Relevant globally as processes, threads, and process scheduling are fundamental concepts in operating systems used worldwide.
Employability	It covers essential concepts and skills related to processes, threads, and process scheduling in operating systems.
Entrepreneurship	-

Skill Development	Provides foundational knowledge and skills related to processes, threads, and process scheduling.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Process-Synchronization, Deadlocks & I/O Systems**
Local	-
Regional	-
National	It is important for national educational institutions to offer this course to produce skilled graduates who can contribute to the national workforce.
Global	Fundamental concepts are applicable in various global industries and organizations.
Employability	Concepts are crucial for various roles in software development
Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Distributed Operating Systems & Concurrent System**
Local	-

Regional	-
National	Can contribute to the national workforce and address the challenges of concurrent programming.
Global	It can be applied globally in various industries and organizations that deal with concurrent programming and need professionals who understand these concepts.
Employability	Graduates with knowledge of these concepts are highly sought after by companies that develop concurrent software applications.
Entrepreneurship	Can be beneficial for entrepreneurs in the technology industry, especially those involved in developing software systems that require efficient concurrent processing.
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4, 8, 9, 11
NEP 2020	OS supports collaborative learning environments, which are encouraged under NEP 2020 to promote interactive and engaging teaching practices.
POE/4th IR	OS contributes to the development of smart systems, autonomous devices, and intelligent algorithms that are central to the 4IR and POE.

OPERATING SYSTEM LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: OPERATING SYSTEM LAB	**Course Code**	**L-T-P**	**Credits**
	ENCS351	0-0-2	1
Type of Course:	MAJOR		
Pre-requisite(s), if any: Basics of programming			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	**Recall** the concepts and principles of CPU scheduling algorithms used in operating systems.
CO 2	**Compare** and contrast different CPU scheduling algorithms and their advantages and disadvantages.
CO 3	**Implement** CPU scheduling algorithms, such as Round Robin and Priority, using Python programming.
CO 4	**Evaluate** the performance of CPU scheduling algorithms by analyzing and interpreting the generated Gantt charts and calculating average waiting time and turnaround time.
CO 5	**Design** Python programs to simulate various file allocation strategies and memory management techniques, such as sequential, indexed, linked, and paging.

List of Programs

Ex No	Experiment Title	Mapped CO/COs
1	Write Python programs to simulate the following CPU Scheduling algorithm: First-Come, First-Served (FCFS)	CO1
2	Write Python programs to simulate the following CPU Scheduling algorithm: Shortest Job First (SJF)	CO1
3	Write Python programs to simulate the following CPU Scheduling algorithms: Round Robin	CO1
4	Write Python programs to simulate the following CPU Scheduling algorithms: Priority	CO1
5	Given the list of processes, their CPU burst times, and arrival times, write a Python program to display/print the Gantt chart for Priority and Round Robin scheduling algorithms. Compute and print the average waiting time and average turnaround time for each scheduling policy.	CO4
6	Write a Python program to simulate the following file allocation strategies like Sequential	CO5
7	Write a Python program to simulate the following file allocation strategies like Indexed	CO5
8	Write a Python program to simulate the following file allocation strategies like linked.	CO5
9	Write Python programs to simulate the following contiguous memory allocation techniques: a) Worst-fit b) Best-fit c) First-fit	CO5
10	Write programs using the I/O system calls of UNIX/Linux operating system (open, read, write, close, fcntl, seek, stat, opendir, readdir).	CO1
11	Write program to simulate the MVT (Multiple Variable Tasks) memory management technique.	CO5
12	Write program to simulate the MFT (Multiple Fixed Tasks) memory management technique.	CO5

13	Write program to simulate the Banker's Algorithm for Deadlock Avoidance and Prevention.	CO5
14	Write program to implement the Producer-Consumer problem using semaphores using UNIX/Linux system calls.	CO3
15	Write programs to illustrate the following IPC (Inter-Process Communication) mechanisms: a) Pipes	CO3
16	Write programs to illustrate the following IPC (Inter-Process Communication) mechanisms: a) FIFOs (Named Pipes)	CO3
17	Program to implement process synchronization using semaphores in Python.	CO4
18	Program to implement a basic File allocation strategy like sequential file allocation in Python.	CO5
19	Program to demonstrate the use of signals in Python for process management.	CO1
20	Program to create and manipulate threads in Python.	CO3
21	Program to implement memory management techniques (e.g., paging, segmentation) in Python.	CO5
22	Program to simulate file system operations (e.g., open, read, write, close) in Python.	CO1
23	Program to implement process synchronization using mutex locks in Python.	CO4
24	Program to simulate the working of virtual memory in Python.	CO5
25	Program to simulate disk file management operations (e.g., allocation, deallocation) in Python.	CO5
26	Program to implement file locking mechanisms (e.g., advisory, mandatory) in Python.	CO5
27	Write a Python program to simulate the following file organization techniques Two level directories	CO5
28	Write Python programs to simulate the paging in memory management techniques	CO5
29	Write Python programs to simulate the segmentation in memory management techniques	CO5
30	Write a Python program to simulate the following file organization technique: Single level directory	CO5

SOFTWARE ENGINEERING

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Software Engineering	**ENCS305**	4-0-0	4
Type of Course:	Programme Core / Programme Elective /Open Elective		
Pre-requisite(s), if any:			
Brief Syllabus: Importance of Software Engineering, Software Development Lifecycle and its models, Agile vs. Plan Based development, Development of Software Documents, Version Control system using GitHub and Eclipse IDE, Requirements Engineering technique, Development of UML Diagrams, Software Architecture and Design patterns, Software Testing- Black Box and White Box, Developing Test cases using Equivalence and Boundary value partitioning techniques, Test Driven Development with Junit in Eclipse, Software Refactoring.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction to Software Engineering**		**No. of hours: 6**
Content Summary: Importance of Software Engineering, Discipline of Software Engineering; Eclipse Introduction, Overview, and Demo; Lifecycle models: Requirements Engineering, Design and Implementation, Maintenance, Software Process Model Introduction, Waterfall Process, Spiral Process, Evolutionary Prototyping Process, Agile Process, Choosing a Model, Lifecycle Documents.			
Unit Number: 2	**Title: Engineering Requirements**		**No. of hours: 8**
Content Summary: Requirements Engineering: General RE Definition, Functional and Non-functional Requirements, User and System Requirements, Modelling Requirements, Analyzing Requirements, Requirements Prioritization, Requirements Engineering Process and			

steps; Creating SRS and performing requirements inspections. Engineering standards in building, testing, operation and maintenance of the computer and software systems. Requirements analysis using DFD, ER Diagrams, Requirement documentation, Nature of SRS, Characteristics & organization of SRS.

Unit Number: 3	**Title: Software Metrics and UML**	**No. of hours: 14**

Content Summary:

Software Metrics: Size Metrics like LOC, Token Count, Function Count, Design Metrics, Data Structure Metrics, Information Flow Metrics. Cost Estimation Models: COCOMO, COCOMO-II.

Object Orientation Introduction, UML Structural Diagrams: Class Diagrams, Component Diagram, UML Structural Diagram: Deployment Diagram. UML creation tips; UML Behavioural Diagram: Use Case, Use Case Diagram: Creation Tips, UML Behavioural Diagrams: Sequence, UML Behavioural Diagrams: State Transition Diagram. UML creation tips; Software Architecture: What is Software Architecture? Advantages and use of architectural models. Architectural patterns. Designing architectural patterns. Design Patterns: Patterns Catalogue, Pattern Format, Factory Method Pattern, Strategy Pattern, Choosing a Pattern, Negative Design Patterns.

Unit Number: 4	**Title: Software Testing and Maintenance**	**No. of hours: 12**

Content Summary:

Testing: Black Box Testing Failure, Fault and Error, Verification Approaches, Pros and Cons of Approaches, Testing Introduction, Testing Granularity Levels, Alpha and Beta Testing, Black-Box Testing, Systematic Functional Testing Approach; Test Data Selection, Equivalence Partitioning and Boundary Value Analysis, Create and Evaluate Test Case Specifications, Generate Test Cases from Test Case Specifications. White-Box Testing: Coverage Criteria Intro, Statement Coverage, Control Flow Graphs, Test Criteria, MC/DC Coverage.

Software Maintenance: Management of Maintenance, Maintenance Process, Maintenance Models, Regression Testing, Reverse Engineering, Software Re-engineering.

Reference Books:

a) R. Pressman, Software Engineering A Practitioner's Approach (8 ed.), McGraw Hill International, 2019. ISBN 978-1259253157.

b) Sommerville, Software Engineering (10 ed.), Person Publications Publishing Company, 2017. ISBN 978-9332582699.

c) K. K. Aggarwal & Yogesh Singh, "Software Engineering", New Age International.

d) W.S. Jawadekar, "Software Engineering – Principles and Practices", McGraw Hill.

Define Course Outcomes (CO)

COs	Statements
CO1	Demonstrate understanding of Software Engineering as an iterative and systematic process.
CO2	Recall the lifecycle models of software engineering.
CO3	Design the software development process to complement technical understanding of software products.
CO4	Analyze requirements using modeling techniques such as DFD and ER diagrams.
CO5	Generate test case specifications and test cases from given requirements.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels (C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A3	P4
CO2	C1	A1	P1
CO3	C6	A4	P5
CO4	C4	A4	P2
CO5	C3	A2	P3

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	-	2	-	-	2	-	2	-	-	-	2
CO2	3	2	3	-	2	-	-	-	-	-	-	2
CO3	3	-	3	-	2	-	-	-	2	-	2	-
CO4	3	3	2	2	2	-	-	-	-	-	-	-
CO5	2	2	2	-	3	-	-	-	-	2	-	-

Justification for mapping must be relevant

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	2	2	1
CO2	2	2	2	2	1
CO3	3	3	2	2	1
CO4	2	3	2	2	1
CO5	2	3	2	2	2

Relevance of the Syllabus to various indicators

Unit I	**Introduction to Software Engineering**
Local	Can help students understand the local software industry and its specific challenges.
Regional	Can provide insights into the regional software development practices and challenges.
National	Address the broader context of software engineering within a country, including its impact on the economy and society
Global	Explore the global nature of software development and its impact on various industries and sectors worldwide.
Employability	Provide students with a foundational understanding of software engineering concepts and practices, which are valuable skills in the job market.
Entrepreneurship	Provide insights into the software industry, its challenges, and potential opportunities for innovation and business ventures.
Skill Development	Introduce fundamental concepts and techniques used in software engineering.
Professional Ethics	Consideration of ethical issues in software development,

	such as privacy, security, and responsible use of technology.
Gender	-
Human Values	Impact of software on individuals, societies, and ethical considerations related to human well-being.
Environment & Sustainability	-
Unit II	**Engineering Requirements**
Local	Help in assessing the complexity and quality of software developed within the local context.
Regional	Provide insights into the software development practices and trends within a specific region.
National	Contribute to evaluating software quality and productivity within a country's software industry.
Global	Provide standardized measures for assessing software complexity and quality, regardless of the geographical location.
Employability	Commonly used in software development organizations to measure productivity, quality, and project estimation.
Entrepreneurship	Evaluating the feasibility, cost estimation, and risks associated with software development projects.
Skill Development	By enhancing the ability to measure, analyze, and improve software quality and productivity.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Software Metrics and UML**

Local	Standardized techniques can be employed by local software development teams.
Regional	Provide a common language and methodology for software development, facilitating collaboration and communication among regional software development teams.
National	Provide a standardized framework for software development, promoting consistency and interoperability among national software projects.
Global	Widely adopted internationally, allowing for effective communication and collaboration among software development teams across different countries.
Employability	Commonly used in industry, and proficiency in these techniques is valued by employers.
Entrepreneurship	Aiding entrepreneurs in planning, designing, and communicating their software ideas.
Skill Development	Enhancing students' proficiency in software modeling and design.
Professional Ethics	Address the importance of developing reliable software and adhering to quality standards in the software engineering profession.
Gender	-
Human Values	-
Environment & Sustainability	Development of reliable software that reduces wastage, energy consumption, and potential negative environmental impacts.
Unit IV	**Software Testing and Maintenance**
Local	Provide practical knowledge and techniques for testing software developed within the local context.
Regional	Address common testing challenges and practices in software development within the region.

National	Provide essential knowledge and skills required for testing software developed within the country.
Global	Testing is an integral part of software development across different countries and industries worldwide.
Employability	As software testing skills are in high demand by employers seeking quality assurance in software development projects.
Entrepreneurship	Provide knowledge and techniques for ensuring the quality and reliability of software products developed by entrepreneurs.
Skill Development	Introduce essential concepts, methodologies, and tools used in software testing.
Professional Ethics	Addressing ethical considerations in software testing, such as ensuring impartiality, confidentiality, and integrity in the testing process.
Gender	-
Human Values	-
Environment & Sustainability	
SDG	SDG 4,8,9
NEP 2020	-
POE/4th IR	Emphasizes the responsible and ethical development and deployment of the systems.

MOBILE APPLICATION DEVELOPMENT

<table>
<tr><td>Department:</td><td colspan="3">Department of Computer Science and Engineering</td></tr>
<tr><td rowspan="2">Course Name:

Mobile Application Development</td><td>Course Code</td><td>L-T-P</td><td>Credits</td></tr>
<tr><td>ENSP313</td><td>4-0-0</td><td>4</td></tr>
<tr><td>Type of Course:</td><td colspan="3">Minor</td></tr>
<tr><td colspan="4">Pre-requisite(s), if any: Basics of Android</td></tr>
<tr><td colspan="4">Brief Syllabus:

This course on mobile app development is based on Android acting as a mobile device software stack that includes an operating system, middleware, and key applications. This rich source of software is used in Mobile Technology via the Android Software Development Kit's innovation module (SDK). As basics you learn to create new interface for the app and perform different interactive operations</td></tr>
<tr><td colspan="4">UNIT WISE DETAILS</td></tr>
<tr><td>Unit Number: 1</td><td colspan="2">Title: Introduction to Android</td><td>No. of hours: 10</td></tr>
<tr><td colspan="4">Content Summary:

Android Introduction, Android Flavors, Android OS Architecture, Gradle Build System, Setting up Android Development Environment, System Requirements, Android Studio Installation, Create First Android Application, Understand Project Hierarchy, Layouts & Views, Resources, User Input Control, List View and Scrolling Views, Recycler View & Card View, Themes & Styles, Material Design, Providing Resources for adaptive layouts, Dialogs – Alert, Progress, and Custom, Floating Action Button, Localization</td></tr>
<tr><td>Unit Number: 2</td><td colspan="2">Title: Activities & Intents</td><td>No. of hours: 10</td></tr>
</table>

Content Summary:

Activity Lifecycle, Activity State, Explicit Intent, Implicit Intent, Intent resolution, Detail Activity, Menu and Icons, Passing data between activities with intents, Activity Navigation, Data Back from Activity

Unit Number: 3	Title: Fragments	No. of hours: 10

Content Summary:

What is a Fragment? Creating a Fragment, Fragment Lifecycle, Handling Fragment Events, Fragment Communication, Fragment Manager, Transactions, Saving State, Communication between fragments and activities, Debugging and Testing

Unit Number: 4	Title: Advance Concepts	No. of hours: 10

Content Summary:

Getting Access to the Action Bar, Option Menus, Context Menus, Popup Menus, Navigation Drawer, Tab Navigation, Swipe View with View Pager, Advantages and Disadvantages of the Action Bar, Action Bar Components, Menu with Action Bar (Example Simplified)

Unit Number: 5	Title: Broadcast & Services	No. of hours: 10

Content Summary:

Broadcast Receivers & Notification, Broadcast Receiver, Sending a Broadcast, Classes of Broadcasts, Context-registered receivers, Manifest Declared Receivers, Permissions in Broadcasts, Sending & Receiving with Permissions, Creating a Receiver, Registering a Broadcast Receiver, Notification, Integrating Notification with a Broadcast Receiver, Services, Types of Services, Service Declaration, Creating a Service, Extending the Service Class, Starting a Service, Stopping a Service, Creating a Bound Service,
The lifecycle of a ServiceRemove axes spine: Despine Size and aspect: Non grid plot, Grid type plot Scale and Context: Poster, paper, notebook and talk.

***Self-Learning Components:**

- **Self-learn programming languages relevant to mobile app development, such as Swift (iOS), Kotlin (Android), or JavaScript (React Native).**

- **Explore frameworks like Flutter, React Native, or Xamarin to build cross-platform mobile applications efficiently.**
- **Explore how to integrate external APIs and services to add functionality and data to your mobile app.**

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components.

Text Books

- Arnon Axelrod, "Complete Guide to Test Automation",Apress

Reference Books

- Elfriede Dustin, Thom Garrett, Bernie Gauf, "Implementing Automated Software Testing: How to Save Time and Lower Costs While Raising Quality", Addison Wesley
- Matsinopoulos, Practical Test Automation, Apress
- Dorothy Graham, Mark Fewster, 'Experiences of Test Automation: Case Studies of Software Test Automation', Addison Wesley
-

Course Outcomes (CO)

COs	
CO1	Recall the fundamental concepts of mobile application development, including programming languages and development environments.
CO2	Understand the principles of mobile app security and data protection.
CO3	Evaluate security practices and propose measures to enhance the security of a mobile app's user data
CO4	Develop a testing and deployment plan that includes continuous integration and continuous deployment (CI/CD) practices.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels(C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A1	P1
C02	C3	A4	P1
CO3	C5	A2	P2
CO4	C6	A1	P4

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	1	2	-	-	-	-	-	-	-	-	3
CO2	-	3	2	-	2	-	-	-	-	-	-	3
CO3	-	2	2	3	1	-	-	-	-	-	-	3
CO4	-	2	3	-	-		-	-	-	-	-	2

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2		-	3	-
CO2	2	2	-	2	-
CO3	-	2	-	-	1
CO4	-	-	-	3	2

Relevance of the Syllabus to various indicators

Unit I	**Introduction to Andriod**
Local	-
Regional	-
National	-
Global	Competence in Android app development positions individuals and businesses competitively in the global tech market.
Employability	Proficiency in Android app development enhances employability in software development roles.
Entrepreneurship	Learning Android app development enables entrepreneurs to create innovative mobile applications, fostering entrepreneurship in the tech sector.
Skill Development	Android app development enhances technical skills and contributes to a well-rounded skill set in software development.
Professional Ethics	Teaching Android app development aligns with professional ethics by promoting responsible and ethical software development practices.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Activities & Intents**
Local	-
Regional	-
National	-
Global	-
Employability	Proficiency in activity lifecycle, intents, and navigation

	enhances employability in software development roles.
Entrepreneurship	Learning about activity lifecycle, intents, and navigation enables entrepreneurs to create innovative and user-friendly mobile applications, fostering entrepreneurship in the tech sector.
Skill Development	Understanding activity lifecycle, intents, and navigation contributes to a well-rounded skill set in Android app development.
Professional Ethics	Teaching activity lifecycle, intents, and navigation aligns with professional ethics by promoting responsible and user-centered software development practices.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Fragments**
Local	-
Regional	-
National	-
Global	-
Employability	-
Entrepreneurship	Learning about Fragments enables entrepreneurs to create versatile and user-friendly mobile applications, fostering entrepreneurship in the tech sector.
Skill Development	Understanding Fragments contributes to a comprehensive skill set in Android app development.
Professional Ethics	Teaching Fragments promotes responsible and user-centered software development practices.
Gender	-

Human Values	-
Environment & Sustainability	-
Unit IV	**Advance Concepts**
Local	-
Regional	-
National	-
Global	-
Employability	Proficiency in designing user interfaces and implementing navigation enhances employability in software development roles.
Entrepreneurship	Learning about user interface components and navigation enables entrepreneurs to create user-friendly and innovative mobile applications, fostering entrepreneurship in the tech sector.
Skill Development	Understanding user interface components and navigation contributes to a well-rounded skill set in Android app development.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit V	**Broadcast & Services**
Local	-
Regional	-
National	-

Global	Understanding Broadcast Receivers, Notifications, and Services positions individuals and businesses competitively in the global app development market.
Employability	Proficiency in creating Broadcast Receivers, Notifications, and Services enhances employability in software development roles.
Entrepreneurship	Learning about Broadcast Receivers, Notifications, and Services enables entrepreneurs to create innovative and user-engaging mobile applications, fostering entrepreneurship in the tech sector.
Skill Development	Understanding Broadcast Receivers, Notifications, and Services contributes to a well-rounded skill set in Android app development.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4, SDG 8, SDG 9
NEP 2020	
POE/4th IR	The 4th Industrial Revolution's technological advancements and POE's engineering principles provide a strong foundation for mobile app development, enabling developers to create innovative, efficient, and ethical applications that contribute to the digital transformation of society.

MOBILE APPLICATION DEVELOPMENT LAB

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Mobile Application Development Lab	**ENSP363**	0-0-2	1
Type of Course:	Minor		
Pre-requisite(s), if any: Basics of Android			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Implement user input controls, like buttons and text fields, to gather user data and interactions.
CO 2	Assess the effectiveness of using notifications and Broadcast Receivers for communicating app updates and events.
CO 3	Construct a mobile app that demonstrates the integration of various Android components, such as services and content providers.
CO 4	Organize their code professionally using objects and blocks, prototype several entry- level apps and try to publish on App store.

Ex. No.	Experiment Title	Mapped CO/COs
1	Case Study of Objective-C language. Create a simple "Hello World" app to get familiar with the Android development environment, project structure, and building and running apps.	CO2

2	Develop a basic calculator app that performs arithmetic operations like addition, subtraction, multiplication, and division.	CO2
3	Build an app that allows users to create, edit, and delete tasks, demonstrating UI elements, data handling, and user interaction.	CO2
4	Create an app that fetches weather data from an API and displays current weather conditions based on user location or input.	CO1
5	Develop an app that converts currency values based on real-time exchange rates retrieved from an API.	CO1
6	Build a quiz app with multiple-choice questions, scoring, and feedback, testing the user's knowledge on a specific topic.	CO1
7	Design an app that helps users learn new vocabulary or concepts through flashcards, with features like flipping and shuffling cards.	CO1
8	Create an app for taking and organizing notes, incorporating features like categorization, searching, and editing.	CO1
9	Develop an app that tracks and displays user's daily steps, distance, and calories burned using built-in sensors.	CO3
10	Design an app that captures photos or records videos, allowing users to apply filters, effects, and save or share media.	CO3
11	Build a basic music player app that allows users to play, pause, skip, and shuffle songs from a predefined playlist.	CO3
12	Create an app that uses location services to find nearby restaurants, displays their details, reviews, and allows users to make reservations.	CO3

13	Develop a simple shopping app that lists products, allows users to add items to the cart, and proceed to checkout.	CO3
14	Design a real-time chat app that enables users to send and receive messages, incorporating features like typing indicators and message notifications.	CO3
15	Build an app that sets daily fitness challenges for users and tracks their progress over time, motivating them to stay active.	CO3
16	Create an app that teaches basic phrases and vocabulary in a foreign language, incorporating interactive exercises and quizzes.	CO3
17	Develop an app that allows users to draw and sketch using touch gestures, offering various brush sizes, colors, and undo/redo functionality.	CO4
18	Design an app that sets location-based reminders for users, sending notifications when they approach or leave a specific area.	CO4
19	Build an app that provides users with a collection of recipes, including ingredients, instructions, and cooking timers.	CO4
20	Create an app that helps users plan their trips by suggesting destinations, providing travel information, and offering itinerary customization.	CO4

NEW AGE PROGRAMMING LANGUAGES LAB

<table>
<tr><td>Department:</td><td colspan="3">Department of Computer Science and Engineering</td></tr>
<tr><td rowspan="2">Course Name:

New-Age programming languages (GO, F#, Clojure, Kotlin)</td><td>Course Code</td><td>L-T-P</td><td>Credits</td></tr>
<tr><td>SEC042</td><td>0-0-4</td><td>2</td></tr>
<tr><td>Type of Course:</td><td colspan="3">SEC</td></tr>
<tr><td colspan="4">Pre-requisite(s), if any:</td></tr>
<tr><td colspan="4">Brief Syllabus:

New-Age programming languages (GO, F#, Clojure, Kotlin) provides an introduction to the concepts and applications of modern programming languages. It explore the features and benefits of GO, F#, Clojure, and Kotlin, and develop practical skills in programming using these languages. The course will cover language syntax, data types, control structures, functional programming concepts, concurrency, and integration with other technologies.</td></tr>
<tr><td colspan="4">UNIT WISE DETAILS</td></tr>
<tr><td>Unit Number: 1</td><td colspan="2">Title: GO programming Language</td><td>No. of hours: 10</td></tr>
<tr><td colspan="4">Content Summary:

Overview of GO, F#, Clojure, and Kotlin, Comparison with traditional programming languages, Installation and setup of development environment, Introduction to GO syntax and data types, Control structures, Functions and packages in GO, Arrays, slices, and maps in GO, Structs and custom data types, Pointers and memory management, Concurrency and parallelism in GO, Error Handling, Concurrent Programming in GO, Advanced GO Concepts- Function closures and anonymous functions, Reflection and type introspection, Testing and benchmarking in GO, Writing concurrent and parallel programs.</td></tr>
<tr><td>Unit</td><td colspan="2">Title: F# Programming</td><td>No. of hours: 10</td></tr>
</table>

Number: 2	Language	

Content Summary:

Introduction to F# syntax and functional programming concepts, Data Types, Variables, Operators, Decision Making, Loops, Functions, Strings, Options, Immutable data types and pattern matching, Higher-order functions and currying, Asynchronous and parallel programming in F#, Object-Oriented Programming with F#, Database access with F#, Querying and manipulating data using F#, Integration with relational and NoSQL databases

Unit Number: 3	Title: Introduction to Clojure Programming	No. of hours: 10

Content Summary:

Overview of Clojure and its features, Setting up the development environment, Basic syntax and data structures in Clojure, Functional Programming in Clojure, Immutable data and pure functions, Higher-order functions and recursion, Collections and sequence operations in Clojure, Destructuring and pattern matching, Macros and metaprogramming in Clojure, Concurrency models in Clojure, Asynchronous programming with core.async, Parallel programming with reducers and pmap, Interacting with Java libraries and APIs, Java interoperability in Clojure, Working with Java collections and objects, Web Development with Clojure, Building web applications using Clojure and Ring, Database access and persistence in Clojure, Error Handling and Testing: Exception handling and error management in Clojure, Testing strategies and frameworks in Clojure, Data Manipulation and Transformation: Data manipulation with Clojure's sequence functions, Data transformation with transducers, Data-driven development with data literals and data readers

Unit Number: 4	Title: Introduction to Kotlin Programming	No. of hours: 10

Content Summary:

Overview of Kotlin and its advantages, Setting up the development environment, Basic syntax and data types in Kotlin, Conditional statements and loops, Function declarations and parameters, Lambda expressions and higher-order functions, Object-Oriented Programming in Kotlin: Classes, objects, and inheritance, Properties and access modifiers, Interfaces and abstract classes, Understanding nullable and non-nullable types, Safe calls and the Elvis operator, Type inference and smart casting, Collections and Functional Programming: Working with lists, sets, and maps in Kotlin, Collection operations and transformations, Introduction to

functional programming concepts in Kotlin, Creating extension functions in Kotlin, Using DSLs for domain-specific problems, Builder pattern and DSL implementation.

***Self-Learning Components:**

1. Web programming with GO
2. F# for Data Science and Machine Learning:
3. Metaprogramming and DSLs in Clojure:
4. Android App Development with Kotlin:

References:

1. Building Modern Web Applications with Go (Golang) by Udemy
2. https://www.jetbrains.com/academy/
3. https://www.classcentral.com/subject/f-sharp
4. https://www.classcentral.com/subject/clojure

Please Note:

At least 5-10 % syllabus will be asked in end term exams from self-learning components

Reference Books:

1. The Go Programming Language, Alan A. A. Donovan and Brian W. Kernighan, Addison-Wesley Professional.
2. An Introduction to Programming in Go, Caleb Doxsey, CreateSpace Independent Publishing.
3. Real-World Functional Programming: With Examples in F# and C#, Tomas Petricek and Jon Skeet, Manning.
4. Programming F# 3.0: A Comprehensive Guide for Writing Simple Code to Solve Complex Problems, Chris Smith, O'Reilly Media.
5. Getting Clojure: Build Your Functional Skills One Idea at a Time, Russ Olsen, O'Reilly.
6. The Joy of Clojure, Michael Fogus and Chris Houser, Manning Publication.
7. Atomic Kotlin, Bruce Eckel and Svetlana Isakova, Mindview LLC.
8. Kotlin in Action, Dmitry Jemerov and Svetlana Isakova, Manning Publication.

Online References:

1. https://gobyexample.com/ [
2. https://golang.org/doc/
3. https://www.youtube.com/playlist?list=PLlxmoA0rQ-LwgK1JsnMsakYNACYGa1cjR
4. https://kotlinlang.org/docs/home.html
5. https://docs.microsoft.com/en-us/dotnet/fsharp/
6. https://www.udemy.com/course/learning-functional-programming-with-f/

7. https://clojure.org/guides/getting_started

Course Outcomes (CO)

COs	Statements
CO1	**Understand** the fundamental principles and paradigms of modern programming languages, including functional programming, object-oriented programming, and concurrent programming.
CO2	**Develop** proficiency in using the syntax, data structures, and control flow constructs of each language (GO, F#, Clojure, and Kotlin) to solve programming problems.
CO3	**Explore** the unique features and strengths of each language, such as Go's focus on concurrency, F#'s functional programming capabilities, Clojure's emphasis on immutability and simplicity, and Kotlin's interoperability with existing Java code.
CO4	**Apply** the languages' respective development tools, such as Go's gofmt and go vet, F#'s F# Interactive (FSI), Clojure's Leiningen or Boot, and Kotlin's integrated development environment (IDE) support, to improve code quality and productivity.
CO5	**Design and implement** projects that integrate multiple programming languages, using appropriate inter-language communication mechanisms and libraries (e.g., Go and Kotlin interacting via REST APIs, F# and Clojure communicating via message queue

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels(C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A1	-
C02	C3	A2	P2
CO3	C2	A3	-
CO4	C3	A4	P3
CO5	C6	-	P4

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	-	-	2	-	-	-	-	2	-	-
CO2	2	2	-	-	2	-	-	-	-	2	-	-
CO3	2	2	-	3		-	-	-	-		-	3
CO4	-	-	-	-	3	-	-	-	-	3	-	2
CO5	-	-	-	-	-	2	-	-	-	2	-	-

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	-	-	3	1
CO2	3	2	-	-	1
CO3	-	2	-	3	-
CO4	-	2	-	3	-
CO5	-	2	2	2	2

Relevance of the Syllabus to various indicators

Unit I	**Introduction to New-Age Programming Languages and GO programming Language**
Local	-
Regional	-
National	Provides essential knowledge and skills related to modern programming languages like GO, F#, Clojure, and Kotlin, which are widely used in national software development projects and initiatives.
Global	Relevant in the global software development community, as these languages and concepts have international adoption and usage.
Employability	Highly valued in the job market, as these languages are used in various industries and offer opportunities for software development roles.
Entrepreneurship	Equips students with knowledge of modern programming languages and advanced concepts, enabling them to explore entrepreneurial opportunities in software development and innovation using these languages.
Skill Development	Enhances students' technical skills in software development, making them more competent in the field of computer science.

Professional Ethics	While not directly related to professional ethics, the syllabus indirectly promotes ethical practices by emphasizing the importance of error handling, writing efficient and concurrent programs, and following best practices in software development.
Gender	-
Human Values	Indirectly supports human values by fostering the development of software solutions that are efficient, maintainable, and user-friendly, aligning with values such as accessibility, usability, and user-centric design.
Environment & Sustainability	-
Unit II	**F# Programming Language**
Local	-
Regional	-
National	Provides essential knowledge and skills which are relevant at the national level for software development, data analysis, and database management projects.
Global	As a globally recognized programming language used in various industries and research domains.
Employability	Highly valued in the job market, as these skills are sought after by companies and organizations involved in software development, data analysis, and database management.
Entrepreneurship	Equips with knowledge that allows to explore entrepreneurial opportunities in software development, data-centric applications, and database-driven solutions.
Skill Development	Enhances technical skills in programming, functional programming paradigms, and data manipulation, making them more competent in the field of computer science.
Professional Ethics	Indirectly promotes ethical practices by emphasizing good programming practices, data integrity, and security considerations

Gender	-
Human Values	Indirectly supports human values by fostering the development of software solutions that are efficient, maintainable, and user-friendly, aligning with values such as accessibility, usability, and user-centric design.
Environment & Sustainability	-
Unit III	**Introduction to Clojure Programming**
Local	-
Regional	-
National	Provides essential knowledge and skills which are relevant at the national level for software development, web application development, and data-driven solutions.
Global	As a globally recognized programming language used in various industries and research domains.
Employability	Language is highly valued in the job market. The skills are sought after by companies and organizations involved in software development, web application development, and data analysis.
Entrepreneurship	Explores entrepreneurial opportunities in software development, web application startups, and data-centric solutions
Skill Development	Enhances technical skills, making more competent in the field of computer science
Professional Ethics	Indirectly promotes ethical practices by emphasizing good programming practices, data integrity, and security considerations.
Gender	-
Human Values	Indirectly supports human values by fostering the development of software solutions that are efficient, maintainable, and user-friendly, aligning with values such as accessibility, usability, and user-centric design. .

Environment & Sustainability	-
Unit IV	**Introduction to Kotlin Programming**
Local	-
Regional	-
National	Provides essential knowledge relevant at the national level for software development, mobile app development, and general-purpose programming.
Global	Language has global applicability.
Employability	Highly valued in the job market. These skills are sought after by companies and organizations involved in software development, mobile app development, and DSL-based solutions.
Entrepreneurship	Explore entrepreneurial opportunities in software development, mobile app startups, and domain-specific language development
Skill Development	Make more competent in the field.
Professional Ethics	Indirectly promotes ethical practices by emphasizing good programming practices, code readability, and modularity.
Gender	-
Human Values	Indirectly by fostering the development of software solutions efficient, maintainable, and user-friendly.
Environment & Sustainability	-
SDG	**SDG 4, SDG 8, SDG 9**
NEP 2020	Quality education, equity, critical thinking, digital literacy, skill development.
POE/4th IR	Technological advancements, digital transformation, and future-ready skills.

K.R. MANGALAM UNIVERSITY

Proposed Lab Experiments

Department:	**Department of Computer Science and Engineering**		
Course Name: New Age Programming languages Lab	**Course Code**	**L-T-P**	**Credits**
	SEC042	0-0-4	2
Type of Course:	Programme Elective-IV (Full Stack Development)		
Pre-requisite(s), if any:			

Course Outcomes (CO)

COs	Statements
CO1	**Understand** the fundamental principles and paradigms of modern programming languages
CO2	**Develop** proficiency in using the syntax, data structures, and control flow constructs of each language
CO3	**Explore** the unique features and strengths of each language, such as Go's focus on concurrency, F#'s functional programming capabilities, Clojure's emphasis on immutability and simplicity, and Kotlin's interoperability with existing Java code.
CO4	**Apply** the languages' respective development tools and best practices.
CO5	**Design and implement** projects that utilize the strengths of each language to tackle complex problems or tasks.

Ex. No	Experiment Title	Mapped CO/COs
	Practical List on GO Programming Language	
1	Write a program that takes user input and performs basic calculations (e.g., addition, subtraction, multiplication) using different data types like integers and floats. Use control structures like if statements and loops to handle different scenarios and validate user input.	CO2
2	Create a package that contains multiple functions to perform common tasks, such as string manipulation or mathematical operations. Use these functions in a separate program to demonstrate their functionality and reusability.	CO1
3	Implement a program that stores a collection of elements using arrays. Perform operations like adding, removing, or updating elements	CO2
4	Define a struct Person with the following members: name, age, job and salary. Create methods associated with the struct to read data in structure and print data.	CO4
5	Develop a program that utilizes pointers to modify and manipulate data in memory. Explore concepts like referencing, dereferencing, and memory allocation/deallocation.	CO2
6	Write a program that demonstrates the use of Go routines and channels to achieve concurrent execution of tasks.	CO3
7	Create a program that handles various error scenarios and provides appropriate error messages or responses. Write unit tests for critical functions and verify their correctness using Go's testing package.	CO5
8	**Mini Project**: Task Manager Application in Go Create a task manager application using the Go programming language. The application should allow users to manage their tasks by adding, updating, and	CO5

		deleting tasks. The tasks should have attributes such as title, description, due date, and status (e.g., "in progress", "completed").	
Practical List on F# Programming Language			
9	a.	WAP to read marks of 4subjects and calculate the Percentage of student and find the result according to given conditions 60>=1st Division 60<&& 50>= 2nd Division 50<&& 40>=3rd Division 40<=fail.	CO2
	b.	WAP to accept an integer and check whether it is prime or not.	
10	a.	Write a function that takes a string as input and returns the reverse of the string. Also check if a given string is a palindrome	CO2
	b.	Create a function that takes a string as input and performs the following transformations: i. If the string contains only alphabetic characters, convert it to uppercase. ii. If the string contains only numeric characters, convert it to an integer and double its value. iii. If the string contains a mix of alphabetic and numeric characters, return it as is.	
	c.	Design a function that validates an email address based on specific rules, such as the presence of an '@' symbol and a valid domain name. Use pattern matching to check if the input string matches the expected email format.	
12		Implement a program that performs various operations on lists using higher-order functions (define a list of integers or strings). Write pure functions that demonstrate the map, filter, reduce/fold operations.	CO1
13		Implement a program that performs multiple I/O-bound or computationally intensive tasks concurrently using	CO3

	F#'s asynchronous workflows and parallel programming constructs.	
14	Create a program that demonstrates the object-oriented programming (OOP) capabilities of F#. Define classes, objects, and inheritance hierarchies using F#'s OOP syntax.	CO3
15	Create a program that demonstrates the following tasks: i. Establish a connection to both the relational and NoSQL databases using appropriate database drivers or libraries. ii. Perform basic CRUD operations (Create, Read, Update, Delete) on the databases.	CO4
16	**Mini Project**: Employee Management System Create an Employee Management System using the F# programming language and a relational database. The system should allow users to perform CRUD (Create, Read, Update, Delete) operations on employee records stored in the database. It should provide functionality to add new employees, retrieve employee information, update employee details, and delete employee records.	CO5
Practical List on Clojure Programming Language		
17	Write a program that demonstrates the basic syntax and data structures in Clojure, such as lists, vectors, maps, and sets.	CO1
18	Write functions that manipulate and transform sequences using operations such as map, filter, reduce, and take.	CO2
19	Implement a program that showcases asynchronous programming using the core.async library.	CO3
20	Write code that calls Java methods, creates Java objects, and works with Java collections and objects from Clojure.	CO4
21	Develop a web application using Clojure and the Ring library. Set up routes, handle HTTP requests and	CO5

		responses, and render dynamic content.	
22		Write functions that interact with the database, perform CRUD operations, and handle transactions.	CO5
23		Implement error handling mechanisms, such as exception handling and error management, in Clojure.	CO4
24		**Mini Project**: Blogging Platform with Clojure Create a Blogging Platform using the Clojure programming language. The platform should allow users to create and publish blog posts, manage user accounts, and provide functionality for reading and commenting on blog posts. It should utilize a relational database for data storage and retrieval.	CO5
Practical List on Kotlin Programming Language			
25	a	WAP for print following o/p Hello Kotlin!!!	CO2
	b	WAP to take employee's basic salary, dept_code and experience. Calculate bonus according to following criteria i. dept_code = 101 && exp <= 2 bonus = 3% ii. dept_code = 102 && exp <= 4 bonus = 5% iii. dept_code = 103 && exp <= 7 bonus = 8%	
	c	WAP to accept an integer and display average of digit.	
26		Write a program in Kotlin that demonstrates various aspects of function declarations, parameters, and higher-order functions. a. Implement a function that takes two integer parameters and returns their sum. b. Create a function that has default parameter values for an optional third parameter, which is a string representing a greeting. If no greeting is provided, the function should use a default greeting. c. Explore named parameters by creating a function that takes multiple parameters and demonstrate how to call the function by specifying the parameter names explicitly. d. Implement a variable-length argument function	CO2

	that takes a variable number of integers and calculates their average. e. Utilize a higher-order function by creating a function that accepts a lambda expression as a parameter. The lambda should take an integer parameter and return the square of that integer.	
27	WAP to create a class Student with data members' rollno, student name, course and percentage and member functions to accept and display the details of student. a. Implement properties, methods, and constructors in classes. b. Explore access modifiers and visibility scopes in Kotlin.	CO1
28	Implement a program that demonstrates the declaration and usage of nullable and non-nullable variables. Utilize safe calls (?.) and the Elvis operator (?:) to handle nullable values and provide alternative values or perform fallback actions.	CO3
29	WAP to implement various collections like lists, sets, and maps in Kotlin and perform common operations on them. Use collection functions and transformations such as map, filter, and reduce to manipulate data.	CO2
30	Implement a DSL for a domain-specific problem, showcasing Kotlin's expressive syntax and extension functions.	CO5
31	Implement a program that demonstrates the creation and usage of extension functions in Kotlin(Choose a specific class or data type, such as String). For example, you can create an extension function that counts the number of vowels in a string or reverses the string.	CO3
32	**Mini Project**: Quiz App Build a quiz application that presents users with multiple-choice questions on various topics. Users can select their answers, and the app provides instant feedback on correctness. Keep track of the user's score and display the result at the end of the quiz. Include features like a timer, score calculation, and a database of questions.	CO5

LIFE SKILLS FOR PROFESSIONALS-III

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Life Skills for Professionals -III	**AEC013**	3-0-0	3
Type of Course:	AEC		
Pre-requisite(s), if any:			
Brief Syllabus: This Course designed to enhance the employability of individuals by developing essential skills and competencies sought by employers. This program equips participants with a wide range of skills necessary for success in the modern job market. To engage in interactive workshops, practical exercises, role-playing, and real-world simulations to reinforce their learning. The course is designed to be inclusive and caters to individuals from diverse backgrounds and career aspirations. The course is designed to enhance and develop various cognitive skills and mental abilities. This course focuses on strengthening critical thinking, problem-solving, memory, and other cognitive functions to improve overall mental agility and performance			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Data interpretation**		**No. of hours: 6**
Content Summary: Table chart, Line graph, Bar graph, Pie chart			
Unit Number: 2	**Title: Logical Reasoning**		**No. of hours: 6**
Content Summary: Coding & Decoding, Sitting arrangement, Calendar, Clock, Direction Sense, Blood relation, Syllogism.			
Unit Number: 3	**Title: Logical & Non-verbal reasoning**		**No. of hours: 6**

Content Summary: Series, Puzzle Text, Statement & Arguments, Cube & Dice, Non-verbal Reasoning		
Unit Number: 4	**Title: Understanding Stress**	**No. of hours: 6**
Content Summary: Introduction to Stress (i) Introduction to stress: Meaning, Definition, Eustress, Distress, (ii) Types of stress: Acute stress, Episodic Acute stress and chronic stress, signs and Symptoms Sources of stress (i) Psychological, Social, Environmental (ii) Academic, Family and Work stress Impact of stress		
Unit Number: 5	**Title: Employability skills**	**No. of hours: 6**
Content Summary: Identifying job openings, Enhancing interpersonal skills, including teamwork, Applying for a job, Preparing Cover letters, preparing a CV/Resume and Effective Profiling, Group Discussions, Preparing for and Facing a Job Interview, Mock Interview, Feed Back – Improvement		
***Self-Learning Components:** **https://onlinecourses.nptel.ac.in/noc21_hs02/preview** **Please Note:** 1)Students are supposed to learn the components on self-basis 2) At least 5-10 % syllabus will be asked in end term exams from self-learning components.		
Reference Books: Aggarwal, R. S. (2014). Quantitative aptitude (Revised edition). Gladwell, M. (2021). Talking to strangers. Scott, S. (2004). Fierce conversations.		

K.R. MANGALAM UNIVERSITY

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** their critical thinking skills and become adept at analyzing and evaluating information, identifying problems, generating innovative solutions, and making informed decisions.
CO2	**Apply** digital literacy skills necessary for the modern workplace and become proficient in using online platforms relevant to their field.
CO3	**Evaluate** Contribute positively, respect different perspectives, resolve conflicts, and achieve shared goals.
CO4	**Improve** and develop skills related to career planning, job search strategies, and personal branding
CO5	**Create** leadership skills and to motivate and inspire others, manage projects effectively, and demonstrate a proactive and responsible approach to their spoken language.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	-	-
C02	C3	-	-
CO3	C5	-	-
CO4	-	-	P5
CO5	C6	-	-

CO-PO Mapping

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO 1	3	3	-	-	-	-	-	-	-	-	-	3
CO 2	3	2		1	1	-	-	-	-	-	-	3
CO 3	2	3	-	-	-	-	1	-	-	-	-	3
CO 4	2	2	1	1	-	-	-	-	-	3	-	3
CO 5	-	-	1	3	3	-	-	-	-	-	-	3

Justification for mapping must be relevant

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PO Mapping

CO	PSO1	PSO2	PSO3	PSO4	PSO5
CO 1	3	1	-	-	1
CO 2	3	-	-	-	1
CO 3	2	2	-	-	-
CO 4	3	-	-	2	1
CO 5	3	2	3	-	-

Relevance of the Syllabus to various indicators

Unit I	Data interpretation
Local	Improve personality, enhance basic mental ability skills.
Regional	Recognize the importance of continuous learning and practice to maintain and further develop mental ability.
National	Practice leadership strategies for solving problems within time constraints, as in competitive exams.
Global	Aligns with global trends in employment
Employability	Develop skills in real-life situations, such as academic exams, job interviews, and problem-solving scenarios.
Entrepreneurship	Learn to share ideas, listen to others, build consensus, and manage conflicts to achieve common goals in collaborative settings.
Skill Development	Develops Skills in public speaking, interpersonal communication, professional writing, and persuasive communication.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	Logical Reasoning
Local	Recognize the importance of continuous learning and practice to maintain and further develop mental ability.
Regional	Practice attentive listening techniques, such as paraphrasing and asking clarifying questions.
National	Attentively listen to others, understand their perspectives, and respond appropriately.
Global	Aligns with global trends in employment
Employability	Develop skills in participating and contributing to group discussions, meetings, or presentations.
Entrepreneurship	Learn to share ideas, listen to others, build consensus, and

	manage conflicts to achieve common goals in collaborative settings.
Skill Development	Apply skills in real-life situations, such as academic exams, job interviews, and problem-solving scenarios.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	Logical & Non-verbal reasoning
Local	Improve number sense, enhance basic arithmetic skills and strengthen mental math abilities and speed.
Regional	-
National	Learn about Series, Puzzle Text, Statement & Arguments, Cube & Dice, Non-verbal Reasoning
Global	Recognize the importance of continuous learning.
Employability	Develop skills in participating and contributing to group discussions, meetings, or presentations.
Entrepreneurship	-
Skill Development	Recognize the importance of continuous learning and practice to maintain and further develop mental ability.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	Understanding Stress
Local	Attentively listen to others, understand their perspectives, and respond appropriately
Regional	-
National	Contributes to develop skill and improved productivity
Global	Aligns with global trends in encouraged to establish professional connections and learn effective techniques for engaging in informational interviews or networking events
Employability	Enhance the employability of individuals by developing essential skills and competencies sought by employers
Entrepreneurship	-
Skill Development	Strengthening critical thinking, problem-solving, memory,

	and other cognitive functions to improve overall mental agility and performance.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit V	Employability skills
Local	Attentively listen to others, understand their perspectives, and respond appropriately with timelines
Regional	-
National	Contributes to develop skill and improved productivity
Global	Aligns with global trends in understanding importance of networking during the job search process
Employability	Enhance the employability of individuals by developing essential skills and competencies sought by employers
Entrepreneurship	-
Skill Development	Strengthening critical thinking, problem-solving, memory, and other cognitive functions to improve overall mental agility and performance.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	Aligns with the concepts employability

SUMMER INTERNSHIP / PROJECT-II

Department:	**Department of Computer Science and Engineering**		
Course Name: Summer Internship / Project-II	**Course Code**	**L-T-P**	**Credits**
	ENSI351	0-0-0	2
Type of Course:	INT		
Pre-requisite(s), if any: NA			

The duration of the internship will be two weeks. It will be after completion of 2nd Semester and before the commencement of Semester III.

The following options can be opted by the students:

1. Offline internship in industry - Student is supposed to produce a joining letter and relieving letter once the internship is over in case of Offline internship in any industry.

2. Online internships – with organizations /institutions those are approved /supported / recommended by the All-India Council of Technical Education for Internship (like SWAYAM, NPTEL, Internshala etc.).

Report Submission and Evaluation Guidelines:

- Student must prepare a detailed report and submit the report. A copy of the report can be kept in the departments for record.
- Each student must be assigned a faculty as a mentor from the university and an Industry Expert as External Guide or Industry Mentor.
- The presentation by student for Internship/ project should in the presence of all students is desirable.
- Student should produce successful completion certificate in case of summer internship in industry.

Course Outcomes:

At the end of the course, students will be able to:

1. Get exposure to the industrial environment, which cannot be simulated in the classroom and hence creating competent professionals for the industry.
2. Get possible opportunities to learn, understand and sharpen the real time technical / managerial skills required at the job(s).
3. Gain experience in writing technical reports / projects and presentation of it.
4. Learn and gain exposure to the engineer's responsibilities and ethics.
5. Understand the social, economic, and administrative considerations that influence the working environment of industrial organizations.

Semester: 6

COMPUTER ORGANIZATION & ARCHITECTURE

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Computer Organization & Architecture	**ENCS302**	3-1-0	4
Type of Course:	Major		
Pre-requisite(s), if any: Concepts of Digital Electronics			

Brief Syllabus:

Computer Organization & Architecture (COA) covers topics in computer architecture and organization focusing on multicore, graphics-processor unit (GPU), and heterogeneous SOC multiprocessor architectures and their implementation issues (architect's perspective). The objective of the course is to provide in-depth coverage of current and emerging trends in computer organization and architecture focusing on performance and the hardware/software interface. The course emphasis is on analysing fundamental issues in architecture design and their impact on application performance.

UNIT WISE DETAILS

Unit Number: 1	**Title: Introduction**	**No. of hours: 10**

Content Summary:

Role of abstraction, basic functional units of a computer, Von-Neumann model of computation, A note on Moore's law, Notion of IPC, and performance. Data representation and basic operations.

Unit Number: 2	**Title: Instruction Set Architecture (RISC-V)**	**No. of hours: 10**

Content Summary:

CPU registers, instruction format and encoding, addressing modes, instruction set, instruction types, instruction decoding and execution, basic instruction cycle, Reduced Instruction Set Computer (RISC), Complex Instruction Set Computer (CISC), RISC-V instructions; X86 Instruction set.

Unit Number: 3	Title: The Processor	No. of hours: 10

Content Summary:

Revisiting clocking methodology, Amdahl's law, Building a data path and control, single cycle processor, multi-cycle processor, instruction pipelining, Notion of ILP, data and control hazards and their mitigations.

Unit Number: 4	Title: Memory hierarchy, Storage and I/O	No. of hours: 10

Content Summary:

SRAM/DRAM, locality of reference, Caching: different indexing mechanisms, Trade-offs related to block size, associativity, and cache size, Processor-cache interactions for a read/write request, basic optimizations like writethrough/write-back caches, Average memory access time, Cache replacement policies (LRU), Memory interleaving.

Introduction to magnetic disks (notion of tracks, sectors), flash memory. I/O mapped, and memory mapped I/O. I/O data transfer techniques: programmed I/O, Interrupt-driven I/O, and DMA.

***Self-Learning Components:**

- BSim Documentation

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components.

References:

1. https://www.nand2tetris.org/
2. https://www.coursera.org/learn/computer-organization-design

3. https://www.geeksforgeeks.org/computer-organization-and-architecture-tutorials/
4. https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-823-computer-system-architecture-fall-2005/

Textbook:

1. "Computer Organization and Design: The Hardware/Software Interface", David A. Patterson and John L. Hennessy, 5th Edition, Elsevier.

Reference Books:

9. "Computer Organization & Architecture", Smruti Ranjan Sarangi, McGraw Hill
10. "Computer System Architecture", Mano M. Morris, Pearson.
11. "Computer Organization and Embedded Systems", 6th Edition by Carl Hamacher, McGraHill Higher Education
12. "Computer Architecture and Organization", 3rd Edition by John P. Hayes, WCB/McGraw-Hill
13. "Computer Organization and Architecture: Designing for Performance", 10th Edition by William Stallings, Pearson Education.

Online References:

1. https://learning.edx.org/course/course-v1:MITx+6.004.2x+3T2015/block-v1:MITx+6.004.2x+3T2015+type@sequential+block@c3s1/block-v1:MITx+6.004.2x+3T2015+type@vertical+block@c3s1v1
2. RIPES: https://freesoft.dev/program/108505982
3. GEM5: https://www.gem5.org/documentation/learning_gem5/introduction/
4. CACTI: https://github.com/HewlettPackard/cacti
5. PIN: https://www.intel.com/content/www/us/en/developer/articles/tool/pin-a-binary-instrumentation-tooldownloads.html
6. TEJAS: https://www.cse.iitd.ac.in/~srsarangi/archbooksoft.html
7. XILINX(VHDL/Verilog tools): https://www.xilinx.com/support/university/students.html

Course Outcomes (CO)

COs	Statements
CO1	Understand the basics of instructions sets and their impact on processor design
CO2	**Demonstrate** an understanding of the design of the functional units of a digital computer system

CO3	**Evaluate** cost performance and design trade-offs in designing and constructing a computer processor including memory.
CO4	**Design** a pipeline for consistent execution of instructions with minimum hazards
CO5	**Manipulate** representations of numbers stored in digital computers using I/O devices and store them into memory

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels(C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A1	P1
C02	C3	A4	P1
CO3	C5	A2	P2
CO4	C6	A1	P4
CO5	C4	A2	P3

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	-	-	-	-	-	-	-	-	-	3
CO2	-	3	2	-	1	-	-	-	-	-	2	3
CO3	-	1	2	3		-	-	-	-	-	-	3
CO4	-	2	3	-	2		-	-	-	-	1	3

CO5	-	2	3	-	2	-	-	-	-	-	1	3

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	-	-	3	-
CO2	2	2	-	2	-
CO3	-	2	-	-	1
CO4	-	-	-	3	-
CO5	-	2	-	2	1

Relevance of the Syllabus to various indicators

Unit I	**Introduction**
Local	Data representation and basic operations: Local, as it focuses on specific techniques and algorithms used within a computer system. Notion of IPC: Local, as it refers to the communication and interaction between processes or components within a computer system.
Regional	-
National	-
Global	-
Employability	-
Entrepreneurship	-
Skill Development	-

Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Instruction Set Architecture (RISC-V)**
Local	-
Regional	-
National	
Global	Addressing modes: Global, as they are a fundamental concept in computer architecture and are used in various CPU architectures worldwide. Instruction set: Global, as it refers to the collection of instructions supported by a CPU architecture, which is applicable across different computer systems.
Employability	-
Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**The Processor**
Local	-

Regional	-
National	-
Global	Global, as they are techniques used to improve performance and increase instruction-level parallelism, relevant across different computer architectures.
Employability	
Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Memory hierarchy, Storage and I/O**
Local	-
Regional	-
National	-
Global	Introduction to magnetic disks, notion of tracks, sectors, flash memory: Global, as they are fundamental concepts and technologies applicable to computer storage systems worldwide.
Employability	-
Entrepreneurship	-
Skill Development	-
Professional	-

Ethics	
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4, SDG 8, SDG 9
NEP 2020	Promoting universal access to education, holistic development, multidisciplinary approach, skill development, critical thinking, creativity, ICT integration, research and development, global competencies, and professional ethics.
POE/4th IR	Aligns with the concepts of parallel computing, advanced processors, and memory architectures.

COMPUTER NETWORKS

Department:	**Department of Computer Science and Engineering**		
Course Name: **Computer Network**	**Course Code**	**L-T-P**	**Credits**
	ENCS304	4-0-0	4
Type of Course:	Major		
Pre-requisite(s), if any:			
Brief Syllabus: This course provides a comprehensive study of computer networks, covering fundamental concepts, protocols, and technologies. It emphasizes hands-on learning and explores open-source tools commonly used in the field of computer networking. Through practical assignments and projects, students will gain a solid understanding of network design, implementation, security, and management.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Evolution of Computer Networking**	**No. of hours: 6**	
Content Summary: Data communication Components: Representation of data and its flow Networks, Various Connection Topology, Protocols and Standards, OSI model, Access networks, physical media, Forwarding, routing; packet switching; circuit switching; a network of network, packet delay and loss, end-end throughput.			
Unit Number: 2	**Title: Data Link Layer Design Issues**	**No. of hours: 12**	
Content Summary: Data Link Layer and Medium Access Sub Layer: Error Detection and Error Correction - Fundamentals, Block coding, Hamming Distance, CRC; Flow Control and Error control protocols - Stop and Wait, Go back – N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking, Random Access, Multiple access protocols -			

Pure ALOHA, Slotted ALOHA, CSMA/CD,CDMA/CA.

Unit Number: 3	Title: Introduction to Network Layer and Transport Services	No. of hours: 12

Content Summary:

Network Layer: Switching, Logical addressing – IPV4, IPV6; Address mapping – ARP, RARP, BOOTP and DHCP–Delivery, Forwarding and Unicast Routing protocols. Transport Layer: Process to Process Communication, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), SCTP Congestion Control; Quality of Service, QoS improving techniques: Leaky Bucket and Token Bucket algorithm.

Unit Number: 4	Title: Principles of Network Applications	No. of hours: 12

Content Summary:

Application Layer: Domain Name Space (DNS), DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls, Basic concepts of Cryptography.

***Self-Learning Components:**

https://gaia.cs.umass.edu/kurose_ross/videos/1/

Cisco Networking Academy: network fundamentals, routing and switching, and network security. They provide free learning materials and hands-on practice: https://www.netacad.com/

Open-Source Networking Tools and Technologies

- Open-source network monitoring tools (e.g., Nagios, Zabbix)
- Open-source network management tools (e.g., OpenNMS)
- Open-source network security tools (e.g., Snort, Suricata)

Text Book:

1. Computer Networks (Fifth Edition) – Andrew S. Tanenbaum (Prentice Hall of India)
2. Data communication and Networking(Fourth Edition)- Behrouz A

Forouzan(Tata Mcgraw Hill)

Reference Books:

3. Computer Networking A Top-Down Approach(Fifth Edition)-James F. Kurose-Keith W. Ross (Pearson)
4. Computer Networks – Protocols, Standards and Interfaces (Second Edition) – UylessBlack(Prentice Hall of India Pvt. Ltd.)

Define Course Outcomes (CO)

COs	Statements
CO1	Understand the fundamental concepts and principles of computer networks.
CO2	Demonstrate knowledge of network hardware and software components.
CO3	Develop skills in network administration and management.
CO4	Choose appropriate protocol for desired communication service.

COs Mapping with Levels of Bloom's taxonomy

	Cognitive levels (C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1		
C02	C2		
CO3	C3	A4	

CO4	C6		P5

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	-	-	-	-	-	-	-	-	-	-
CO2	-	2	2	-	-	-	-	-	-	-	-	-
CO3	-	-	-	3	-	-	-	-	-	-	-	-
CO4	-	-	-	3	-	-	-	-	-	-	-	-

Justification for mapping must be relevant

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	-	-	-	1
CO2	-	-	-	-	-
CO3	-	-	2	-	-
CO4	-	-	-	-	-

Relevance of the Syllabus to various indicators

Unit I	**Evolution of Computer Networking**
Local	Computer networking enables local communication and connectivity within communities, businesses, and educational institutions. It facilitates information sharing, collaboration, and resource sharing at the local level.
Regional	Networking infrastructure connects regions and nations,

	promoting economic growth, social development, and knowledge sharing. It enables efficient communication, e-commerce, and government services across regions and countries.
National	-
Global	The global network infrastructure, such as the internet, connects people worldwide. It fosters global communication, cultural exchange, international trade, and enables the global dissemination of knowledge and information.
Employability	Computer networking skills are in high demand across industries. Proficiency in networking technologies and protocols enhances employability prospects in fields such as network administration, cybersecurity, cloud computing, and telecommunications.
Entrepreneurship	Networking knowledge is crucial for entrepreneurs to establish and manage their businesses effectively. It enables the creation of scalable, secure, and interconnected systems that support business operations, communication, and data exchange.
Skill Development	Computer networking cultivates essential technical skills, such as network design, configuration, troubleshooting, and optimization. It also fosters critical thinking, problem-solving, and analytical skills required to address complex network challenges.
Professional Ethics	Computer networking professionals must adhere to ethical standards and guidelines. They need to respect user privacy, ensure data security, and practice responsible use of network resources. Professional ethics in networking include principles like confidentiality, integrity, accountability, and respect for intellectual property rights.
Gender	Promoting gender diversity and inclusivity in computer networking is crucial. Encouraging women's participation in networking fields helps bridge the gender gap and fosters diverse perspectives and innovative solutions.
Human Values	Computer networking should prioritize human values, such as accessibility, equity, and social responsibility. Access to

	network resources and services should be inclusive, regardless of geographical location or socioeconomic background. Networking technologies should be leveraged to bridge digital divides and empower underserved communities.
Environment & Sustainability	Networking can contribute to environmental sustainability by enabling remote work, reducing the need for commuting, and minimizing carbon emissions. It also facilitates energy-efficient network infrastructure design and management, leading to reduced power consumption and environmental impact.
Unit II	**Data Link Layer Design Issues**
Local	Data Link Layer design issues are relevant at the local level as they affect the efficiency and reliability of local area networks (LANs). Local network connectivity is vital for businesses, educational institutions, and communities to facilitate communication and data exchange.
Regional	Efficient Data Link Layer design ensures seamless connectivity within regions and nations, enabling smooth data transmission across a wide range of industries and sectors.
National	-
Global	In the global context, the design of the Data Link Layer plays a crucial role in ensuring interoperability and standardization across networks worldwide. Global communication and data exchange rely on well-designed protocols and technologies at this layer.
Employability	Proficiency in Data Link Layer design is valuable for networking professionals seeking employment in roles such as network engineers, system administrators, and network architects. Employers look for individuals with a strong understanding of data link protocols and the ability to design reliable and efficient data link connections.
Entrepreneurship	Data Link Layer design knowledge is also essential for entrepreneurs who need to establish and manage their network infrastructure effectively. It enables the creation of secure and efficient data links to support business operations and communication.

Skill Development	Understanding Data Link Layer design issues contributes to the development of technical skills in network engineering and administration. It involves knowledge of protocols, error detection and correction techniques, flow control, and media access control.
Professional Ethics	Data Link Layer design should adhere to professional ethics, including principles of integrity, privacy, and security. Designers must ensure the confidentiality and integrity of transmitted data, implement appropriate access control mechanisms, and protect against unauthorized access or data breaches.
Gender	Promoting gender diversity and inclusivity in Data Link Layer design is important to ensure a diverse range of perspectives and innovative solutions. Efforts should be made to encourage and support the participation of underrepresented groups in networking fields.
Human Values	Data Link Layer design should consider human values such as accessibility, reliability, and user-friendliness. Networks should be designed to provide reliable and efficient data transmission, ensuring that users have access to network resources without discrimination or unnecessary barriers.
Environment & Sustainability	Sustainable Data Link Layer design involves optimizing network performance and reducing power consumption. Energy-efficient network technologies, such as link aggregation and power-saving modes, can contribute to environmental sustainability by minimizing energy consumption and reducing carbon emissions.
Unit III	**Introduction to Network Layer and Transport Services**
Local	Local: The Network Layer and Transport Services enable local connectivity and communication within communities, organizations, and institutions. They facilitate local data transmission and routing within a network.
Regional	Regional and National: These networking components play a vital role in regional and national connectivity, enabling data transmission across networks and facilitating communication between different regions and countries.

National	-
Global	Global: The Network Layer and Transport Services are essential for global connectivity, enabling data transmission across the internet and connecting individuals and organizations worldwide.
Employability	Proficiency in the Network Layer and Transport Services is highly relevant for networking professionals seeking employment. It enhances employability in roles such as network engineers, network administrators, and system architects. Employers value individuals with expertise in network design, routing, and transport protocol selection.
Entrepreneurship	Understanding the Network Layer and Transport Services is crucial for entrepreneurs who need to design and manage their network infrastructure effectively. It allows for scalable and efficient data transmission, supporting business operations and facilitating communication.
Skill Development	Skill development in these areas also enhances critical thinking, problem-solving, and troubleshooting skills, as network professionals need to analyze and resolve issues related to routing, congestion control, and reliability.
Professional Ethics	Ethical considerations are important when working with the Network Layer and Transport Services. Networking professionals must ensure the confidentiality, integrity, and availability of data during transmission. They must also respect user privacy and adhere to ethical standards in handling network traffic and data.
Gender	Encouraging gender diversity and inclusivity in the Network Layer and Transport Services is crucial to foster diverse perspectives and innovative solutions. Efforts should be made to promote the participation and representation of underrepresented groups in networking fields.
Human Values	The Network Layer and Transport Services should prioritize human values such as accessibility, reliability, and user-friendliness. Networks should be designed to provide reliable and efficient data transmission, ensuring equitable access and usability for all users.

Environment & Sustainability	Sustainable network design involves optimizing the Network Layer and Transport Services to minimize resource consumption and reduce environmental impact. This includes implementing efficient routing algorithms, congestion control mechanisms, and energy-saving techniques to reduce power consumption and promote environmental sustainability.
Unit IV	**Principles of Network Applications**
Local	Principles of computer network applications enable local communication and connectivity within communities, organizations, and institutions. Local network applications facilitate information sharing, collaboration, and resource utilization at the local level.
Regional	These principles play a crucial role in regional and national connectivity, allowing for efficient communication and data exchange across networks within a region or country.
National	-
Global	The principles of computer network applications are essential for global connectivity, enabling the exchange of data and information across the internet on a global scale.
Employability	Proficiency in the principles of computer network applications enhances employability in various roles such as application developers, network engineers, and system administrators. Employers seek individuals with a strong understanding of network protocols, application design, and development.
Entrepreneurship	Understanding these principles is crucial for entrepreneurs who need to develop and manage networked applications effectively. It enables the creation of innovative and scalable applications to support business operations and provide value to users.
Skill Development	The principles of computer network applications contribute to the development of technical skills in application development, network programming, and protocol implementation. It involves knowledge of network protocols, application layer protocols (e.g., HTTP, FTP), and client-server communication.
Professional	Ethical considerations are important when working with

Ethics	computer network applications. Professionals should prioritize user privacy, data security, and responsible use of network resources. Adhering to ethical guidelines ensures the confidentiality, integrity, and availability of data during application communication.
Gender	Promoting gender diversity and inclusivity in computer network applications is important to ensure diverse perspectives and inclusive designs. Efforts should be made to encourage and support the participation of underrepresented groups in application development and networking fields
Human Values	Principles of computer network applications should prioritize human values such as accessibility, usability, and user-friendliness. Applications should be designed to provide seamless and intuitive user experiences, ensuring equitable access and usability for all users.
Environment & Sustainability	Sustainable application design involves optimizing network communication to minimize resource consumption and reduce environmental impact. This includes implementing efficient data transfer mechanisms, minimizing unnecessary data transmission, and promoting energy-efficient application architectures.
SDG	SDG 4
NEP 2020	Design assessments that evaluate students' practical skills, problem-solving abilities, and understanding of networking concepts.
POE/4th IR	Students can gain a deeper understanding of the practical implications of emerging technologies on network design, management, and security in the modern digital era.

COMPUTER NETWORKS LAB

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Computer Network Lab	**ENCS352**	0-0-2	1
Type of Course:	Major		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	**Understand** the fundamental concepts of computer networks, including network devices, IP addressing, VLANs, and routing protocols, through hands-on experimentation and network configuration.
CO 2	**Develop** practical skills in setting up and configuring computer networks, including wired and wireless networks, and troubleshoot basic connectivity issues using network commands and tools.
CO 3	**Acquire** proficiency in configuring advanced network features such as VLANs, inter-VLAN routing, static routing, and Network Address Translation (NAT) to design complex network architectures.
CO 4	**Design** and **implement** error detection and correction mechanisms using Hamming Codes and CRC for 7/8 bits ASCII codes to ensure data integrity over a network.

Ex. No	Experiment Title	Mapped CO/COs
1	Create a simple network with multiple PCs, switches, and routers.	CO 1
2	Assign IP addresses to devices and configure basic connectivity.	CO 1, CO 2
3	Test connectivity between PCs using ping and trace routes.	CO 2
4	Configure VLANs on switches and assign ports to specific VLANs.	CO 2, CO 3
5	Enable inter-VLAN routing using a router or Layer 3 switch.	CO 3
6	Test connectivity between PCs in different VLANs.	CO 3
7	Set up a network with multiple routers.	CO 1, CO 3
8	Configure static routes on routers to enable communication between networks.	CO 3, CO 4
9	Verify routing tables and test connectivity between networks.	CO 4
10	Set up a network with a private IP address space.	CO 4
11	Configure NAT on a router to enable translation between private and public IP addresses.	CO 4
12	Test connectivity between devices on the private network and the Internet.	CO 4
13	Create a wireless network using access points and wireless clients.	CO 1, CO 2
14	Simulate network issues such as connectivity problems, routing errors, or misconfigurations.	CO 2, CO 3
15	Design and implement a network traffic monitoring.	CO 3
16	Setting up small computer networks and Hands on networking commands: Set up a small wired and wireless network of 2 to 4 computers using Hub/Switch/Access point.	CO 1

17	Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes.	CO 4
18	Write a program for error detection and correction for 7/8 bits ASCII codes using CRC.	CO 4
19	Write a program to simulate Go back N and Selective Repeat Modes of Sliding Window Protocol in peer to peer mode. Further extend it to real implementation of Flow Control over TCP protocol.	CO 4
20	Design and deploy TCP based Multithreaded HTTP client server for accessing student activity data in the institute.	CO 3, CO 4
21	Design and deploy TCP based Multithreaded FTP client server to share institute level notices.	CO 3, CO 4
22	Design and deploy TCP based Multithreaded Chat client server for your class.	CO 3, CO 4
23	Design and deploy UDP based Multithreaded Chat client server for your class.	CO 3, CO 4
24	Examining real-world network deployments.	CO 3, CO 4
25	Case studies of network failures and their resolutions.	CO 2, CO 3, CO 4

DEVOPS & AUTOMATION

Department:	**Department of Computer Science and Engineering**		
Course Name: **DevOps & Automation**	**Course Code**	**L-T-P**	**Credits**
	ENSP316	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any: Nil			
Brief Syllabus: Throughout the subject, students will engage in hands-on exercises and projects to gain practical experience with various DevOps tools and practices. By the end of the course, students will be well-equipped to embrace the DevOps culture and apply automation techniques to enhance software development, delivery, and operations processes.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction to DevOps**	**No. of hours: 10**	
Content Summary: Definition of DevOps: Challenges of traditional IT systems & processes, History and emergence of DevOps, DevOps definition and principles governing DevOps, DevOps and Agile,The need for building a business use case for DevOps, Purpose of DevOps, Application Deployment, Automated Application Deployment, Application Release Automation (ARA), Components of Application Release Automation (ARA), Best Practices of CI, Benefits of CI, CAMS Continuous Delivery and Deployment: Implementing CD pipelines for deploying applications to various environments.			
Unit Number: 2	**Title: DevOps Tools & Technologies**	**No. of hours: 10**	
Content Summary: Introduction to Linux (OS) & why it is important to know Linux while working with DevOps, Git & GitHub (SCM), Docker (Containerization), Jenkins (CI/CD Pipelines), Terraform (Provisioning), Maven (Build & Release Management), Ansible (Configuration Management), Selenium (Test Automation), AWS (Cloud Computing), SonarQube (Code Quality Checking), Prometheus/Nagios (Monitoring)			

Unit Number: 3	Title: Configuration Management and Infrastructure as Code (IaC)	No. of hours: 10

Content Summary:
History of Version Control Systems (VCS), Basic operations in a VCS, Examples of version control systems, Subversion (SVN), Features and Limitations, Mercurial, Git, Overview, History - Linux and Git by Linus Torvalds, Advantages of Git, Explain how local version control works, Centralized Version Control Systems (CVCS), Distributed Version Control Systems (DVCS), advantages of DVCS, Private Workspace

Unit Number: 4	Title: Application Containerization	No. of hours: 10

Content Summary:

Understanding Containers: Transporting Goods Analogy, Problems in Shipping Industry before Containers, Shipping Industry Challenges, Container: Virtualization Introduction, Hypervisor, Scope of Virtualisation, 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 Number: 5	Title: Introduction to Containerization	No. of hours: 10

Content Summary:
Docker architecture, Docker Daemon (Container Platform), Docker Rest API , CLI Different environments: (Dev, QA and Prod), Overcoming issues with different environments, Development Environment Docker Swarm and Kubernetes, Architecture, AWS (ECS,EKS), 10 AWS Elastic Container Services Architecture, Azure Kubernetes Services, Openshift, KUBERNETES ON CLOUD, Monitoring of container Jenkins Installation, Jenkins Dashboard & UI understanding, Jenkins Job, Jenkins Triggers, Jenkins Plugins, Multi Node cluster setup with architecture, Installing/Configuring Nexus, Use Jenkins as a Continuous Integration server, Deploying the application to staging/prod environment, Docker integration with Jenkins, Static Pipeline.

***SELF-LEARNING COMPONENTS:**

https://elearn.nptel.ac.in/shop/iit-workshops/completed/cicd-devops-automation-and-devsecops-automation/

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components.
Reference Books: 1. Jez Humble and David Farley, "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation," Pearson Education, Inc., 2011. 2. Nigel Poulton, "The Kubernetes Book," Independently published, 2018. 3. Sam Newman, "Building Microservices: Designing Fine-Grained Systems," O'Reilly Media, Inc., 2015. 4. Eberhard Wolff, "Microservices Patterns: With examples in Java," Manning Publications, 2018. 5. Yevgeniy Brikman, "Terraform: Up & Running: Writing Infrastructure as Code," O'Reilly Media, Inc., 2017.

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** the principles and benefits of DevOps, and its role in enhancing collaboration and efficiency between development and operations teams.
CO2	**Acquire** hands-on experience with popular DevOps tools such as Git, Jenkins, Docker, Kubernetes, and Ansible for implementing continuous integration, continuous delivery, and automated deployment processes.
CO3	**Demonstrate** proficiency in containerization and orchestration techniques using Docker and Kubernetes for efficient and scalable application deployment and management.
CO4	**Implement** configuration management and Infrastructure as Code (IaC) using Ansible and Terraform to automate the provisioning and management of infrastructure resources.

CO5	**Develop** skills in monitoring, logging, and security practices in the context of DevOps, ensuring application performance, resilience, and adherence to security best practices.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	-	-	P1
C02	C2	-	P2
CO3	C3	L3	P3
CO4	C4	-	-
CO5	C5	L5	P5

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	2	-	3	-	3	-	-	-	3
CO2	3	3	-	3	3	-	-	3	-	-	-	3
CO3	3	3	2	2	3	2	2	3	-	-	-	3
CO4	-	3	2	3	3	-	2	3	2	-	-	3
CO5	-	3	2	3	-	-	3	3	-	-	-	3

Justification for mapping must be relevant.

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PO1	PO2	PO3	PSO4	PSO5
CO1	3	-	1	-	1
CO2	2	1	-	1	-
CO3	3	-	-	-	-
CO4	1	1	1	1	1
CO5	-	2	-	-	1

Relevance of the Syllabus to various indicators

Unit I	**Introduction to DevOps**
Local	Addresses local understanding of the Internet and its impact on society
Regional	Addresses regional internet connectivity and network infrastructure requirements
National	Contributes to national digital literacy and internet connectivity strategies
Global	Aligns with global trends in internet technologies and network protocols
Employability	Develops skills in using internet-based services and understanding network protocols
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in internet technologies and network protocols
Professional Ethics	-

Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**DevOps Tools & Technologies**
Local	Addresses local understanding of the Internet and its impact on society
Regional	-
National	Contributes to national digital literacy and internet connectivity strategies
Global	Aligns with global trends in internet technologies and network protocols
Employability	Develops skills in using internet-based services and understanding network protocols
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in internet technologies and network protocols
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Configuration Management and Infrastructure as Code (IaC)**
Local	Addresses local network security needs and practices
Regional	-
National	Contributes to national network security strategies and

	protocols
Global	Aligns with global trends in network security techniques and protocols
Employability	Develops skills in network programming and network security techniques
Entrepreneurship	-
Skill Development	Develops knowledge and skills in client-server programming and network security
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Application Containerization**
Local	Addresses local understanding and implementation of internet-based services
Regional	-
National	Contributes to national digital communication strategies and multimedia applications
Global	Aligns with global trends in internet telephony, multimedia applications, and SEO
Employability	Develops skills in internet telephony, multimedia applications, and SEO
Entrepreneurship	-
Skill Development	Develops knowledge and skills in internet telephony, multimedia applications, and SEO
Professional Ethics	-

Gender	-
Human Values	-
Environment & Sustainability	-
Unit V	**Introduction to Containerization**
Local	-
Regional	-
National	-
Global	Global tech community benefits from individuals skilled in Docker, Jenkins, and cloud-native technologies, facilitating collaboration and knowledge exchange on a global scale.
Employability	Proficiency in Docker, Jenkins, and container orchestration technologies increases individuals' employability in the tech industry, making them valuable assets for potential employers.
Entrepreneurship	Knowledge of these technologies empowers aspiring entrepreneurs to build and scale applications efficiently, fostering innovation and business growth.
Skill Development	Learning Docker, Jenkins, and container orchestration addresses the evolving needs of the IT industry, fostering skill development for individuals seeking to stay relevant in a rapidly changing technological landscape.
Professional Ethics	Implementing robust monitoring practices for containers and adhering to security standards while deploying applications across different environments showcases national-level professional ethics in software development.
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4

NEP 2020	-
POE/4th IR	Aligns with the concepts of internet telephony, multimedia applications, and SEO

DEVOPS & AUTOMATION LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: **DevOps & Automation Lab**	**Course Code**	**L-T-P**	**Credits**
	ENSP366	0-0-2	1
Type of Course:	Minor		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	Course Outcomes (COs)
CO 1	**Gain** hands-on experience in setting up version control using Git and performing collaborative software development with branching and merging techniques.
CO 2	**Acquire** practical knowledge in implementing continuous integration and continuous deployment (CI/CD) pipelines using Jenkins, automating the build, test, and deployment processes.
CO 3	**Develop** proficiency in containerization with Docker, including managing Docker containers and images, and deploying applications on Kubernetes for efficient and scalable orchestration.
CO 4	**Demonstrate** skills in infrastructure automation and configuration management using Ansible and Terraform to provision and manage cloud resources and application configurations.
CO 5	**Understand** and **apply** monitoring, logging, and security practices in DevOps, ensuring application performance, resilience, and adherence to security best practices throughout the software development lifecycle.

Ex. No.	Experiment Title	Mapped CO(s)
1	Setting up version control with Git	CO1
2	Implementing a basic Jenkins CI/CD pipeline	CO2
3	Automating application deployment with Jenkins	CO2
4	Containerizing an application using Docker	CO3
5	Managing Docker containers and images	CO3
6	Deploying applications with Kubernetes	CO3
7	Implementing Kubernetes deployment strategies	CO3
8	Continuous deployment with Kubernetes	CO3
9	Configuring infrastructure with Ansible	CO4
10	Automating application configuration with Ansible	CO4
11	Implementing Infrastructure as Code (IaC) with Terraform	CO4
12	Creating scalable and resilient infrastructure with Terraform	CO4
13	Monitoring application performance with Prometheus	CO5
14	Logging and centralized log management	CO5
15	Implementing security measures in CI/CD pipelines	CO5
16	Implementing feature flags for controlled feature rollout	CO5
17	Load testing and performance optimization	CO5
18	Automating application tests with Selenium	CO2, CO5
19	Integrating automated testing in CI/CD pipelines	CO2, CO5

20	Blue-green deployment for zero-downtime updates	CO3, CO5
21	Canary deployment for testing new features	CO3, CO5
22	Implementing GitOps for application deployments	CO3, CO5
23	Managing secrets and sensitive data securely	CO5
24	Disaster recovery planning and testing	CO5
25	Creating a DevOps project integrating multiple tools	, CO4, CO5

1. **Setting up version control with Git**: Exercise: Initialize a Git repository, create branches, perform commits, and push changes to a remote repository. Project: Collaboratively work on a project using branching and merging techniques in Git.
2. **Implementing a basic Jenkins CI/CD pipeline**: Exercise: Set up a simple Jenkins pipeline to build and test a sample application from version control. Project: Develop a complete CI/CD pipeline that includes code building, automated testing, and deployment to a staging environment.
3. **Automating application deployment with Jenkins**: Exercise: Configure Jenkins to automatically deploy the application to a test server upon successful build. Project: Implement a full-fledged CD pipeline with Jenkins, including deployment to production after successful testing.
4. **Containerizing an application using Docker**: Exercise: Dockerize a basic application and run it in a container. Project: Containerize a multi-service application with Docker Compose for easier deployment.
5. **Managing Docker containers and images**: Exercise: Explore Docker commands to manage containers and images, such as starting, stopping, and cleaning up. Project: Implement a container registry and manage images for different application versions.
6. **Deploying applications with Kubernetes**: Exercise: Set up a Kubernetes cluster and deploy a basic application using YAML manifests. Project: Deploy

a microservices-based application with Kubernetes, configuring services and network policies.

7. **Implementing Kubernetes deployment strategies**: Exercise: Implement rolling updates and rollbacks in Kubernetes. Project: Use Kubernetes deployment strategies like blue-green and canary deployments for a real-world application.

8. **Continuous deployment with Kubernetes**: Exercise: Set up a Jenkins pipeline for continuous deployment to Kubernetes. Project: Create an end-to-end automated CD pipeline with Jenkins and Kubernetes.

9. **Configuring infrastructure with Ansible**: Exercise: Use Ansible to provision and configure virtual machines. Project: Create a playbook to configure a complete development environment for an application.

10. **Automating application configuration with Ansible**: Exercise: Create Ansible playbooks to automate application-specific configurations. Project: Implement dynamic inventory and use Ansible roles for better code organization.

11. **Implementing Infrastructure as Code (IaC) with Terraform**: Exercise: Set up a basic Terraform configuration to create cloud resources. Project: Use Terraform to define infrastructure for a scalable and fault-tolerant application.

12. **Creating scalable and resilient infrastructure with Terraform**: Exercise: Implement auto-scaling and load balancing in Terraform. Project: Design a Terraform template for a highly available architecture using multiple availability zones.

13. **Monitoring application performance with Prometheus**: Exercise: Set up Prometheus for monitoring application metrics. Project: Create custom Prometheus metrics and use Grafana for visualization and alerting.

14. **Logging and centralized log management**: Exercise: Configure centralized log collection using tools like Fluentd or Logstash. Project: Set up ELK (Elasticsearch, Logstash, and Kibana) stack for efficient log analysis.

15. **Implementing security measures in CI/CD pipelines**: Exercise: Use Jenkins plugins to implement security checks in CI/CD pipelines. Project: Implement security scanning tools like SonarQube and integrate them into the pipeline.

16. **Implementing feature flags for controlled feature rollout**: Exercise: Add feature flags to a sample application to enable/disable specific features. Project: Implement a feature flag service for a real-world application and manage feature rollout.

17. **Load testing and performance optimization**: Exercise: Use load testing tools to evaluate application performance under heavy traffic. Project: Analyze performance bottlenecks and optimize the application for scalability.

18. **Automating application tests with Selenium**: Exercise: Use Selenium WebDriver for automating browser-based tests. Project: Develop an automated testing suite covering multiple application features.

19. **Integrating automated testing in CI/CD pipelines**: Exercise: Integrate automated tests into the Jenkins CI/CD pipeline. Project: Implement a complete testing strategy, including unit, integration, and end-to-end tests.

20. **Blue-green deployment for zero-downtime updates**: Exercise: Perform blue-green deployment for a sample application update. Project: Set up a blue-green deployment strategy for a production application.

21. **Canary deployment for testing new features**: Exercise: Implement canary deployment for a specific application feature. Project: Use canary deployment to gradually release new features to a subset of users.

22. **Implementing GitOps for application deployments**: Exercise: Use GitOps principles to manage Kubernetes manifests with Git. Project: Implement a GitOps workflow for application deployment and configuration management.

23. **Managing secrets and sensitive data securely**: Exercise: Utilize Kubernetes secrets or HashiCorp Vault to manage sensitive data. Project: Set up a secure secret management system for a production environment.

24. **Disaster recovery planning and testing**: Exercise: Design a disaster recovery plan for a sample application. Project: Test the disaster recovery plan and validate its effectiveness.

25. **Creating a DevOps project integrating multiple tools**: Exercise: Choose and integrate various DevOps tools into a sample project. Project: Create an end-to-end DevOps project showcasing the integration of tools and best practices.

CI/CD & MICROSERVICES

Department:	**Department of Computer Science and Engineering**		
Course Name: **CI/CD &Micro-services**	**Course Code**	**L-T-P**	**Credits**
	ENSP314	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any: Nil			
Brief Syllabus: Throughout the subject, students will learn, Version control, automated builds, testing, Docker containers, Kubernetes orchestration, infrastructure as code, and continuous integration and deployment pipelines for efficient software delivery and Microservices architecture, API design, containerization, inter-service communication, resilience, security, observability, and deployment strategies for scalable, modular, and agile application development.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction**		**No. of hours: 8**
Content Summary: In the first unit, you will be introduced to continuous integration (CI), continuous delivery (CD),_and continuous deployment. You will learn about the benefits of CI, how it benefits distributed teams, and the steps involved in CICD. The unit also covers the concept of pipelines and provides a checklist and prerequisites for implementing CI/CD. Furthermore, you will explore the business drivers and advantages of continuous deployment through a case study of the HP Laserjet.			
Unit Number: 2	**Title: Stages of Continuous Integration and Delivery**		**No. of hours:** 8
Content Summary: This unit delves into the stages of a continuous integration process. It starts with an overview of the core CI process, including version control systems (VCS), merging local changes, and code review. You will learn about automated code builds, static code analysis, automated unit testing using JUNIT, and code coverage analysis. The unit also covers advanced CI processes, such as automated functional testing. Additionally, you will examine a case study on Google Canary release.			

Unit Number: 3	Title: Jenkis	No. of hours: 8
Content Summary: You will learn how to install and configure Jenkins, understand its dashboard and user interface, and create Jenkins jobs. The unit covers Jenkins triggers, plugins, and multi-node cluster setup. You will also explore integrating Docker with Jenkins and deploying applications to staging and production environments. The concept of a static pipeline is introduced as well.		
Unit Number: 4	**Title: Anatomy of a Continuous Delivery**	**No. of hours:** 8
Content Summary: anatomy of a continuous delivery pipeline. It covers different types of delivery pipelines, including simple delivery pipeline and continuous deployment pipeline. You will learn about releasing applications to production, achieving zero-downtime releases, and performing rolling back deployments. The unit also introduces concepts like blue-green deployments, canary releasing, and handling emergency fixes. Additionally, you will explore engineering practices for continuous delivery and continuous development/integration.		
Unit Number: 5	**Title: System Monitoring**	**No. of hours:** 8
Content Summary: The final unit introduces system monitoring in the context of DevOps. You will learn the goals of monitoring and the DevOps approach to monitoring. The unit explains the role of the Network Operations Center (NOC) in a DevOps world and explores telemetry and metrics. Different types of monitoring, including end-user, infrastructure, application, and log monitoring and analysis, are also covered.		

***SELF-LEARNING COMPONENTS:**

- Study API design, RESTful principles, and GraphQL
- Learn about service discovery tools like Consul, etcd, or Kubernetes' service discovery.
- Gain insights into securing microservices, including authentication, authorization, and API security. Learn about OAuth, JWT, and API gateways

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components.

Reference Books:

1. Rafal Leszko, "Continuous Delivery with Docker and Jenkins", Packt
2. Anuj Kumar, "Microservices with Clojure", O'Reilly
3. Ronnie Mitra, Irakli Nadareishvili, "Microservices Up and Running", O'Reilly
4. Mohamed Labouardy, "Pipeline as Code: Continuous Delivery with Jenkins, Kubernetes, and Terraform", Manning
5. Murat Karslioglu, "Kubernetes: A Complete DevOps Cookbook", Packt
6. Bob Aiello, "Hands-On DevOps for Architects", Packt

Define Course Outcomes (CO)

COs	Statements
CO1	Recall definitions of Continuous Integration (CI), Continuous Delivery (CD), and Continuous Deployment (CD), and their key differences.
CO 2	Comprehend the core concepts of version control systems (VCS), merging, code review, automated code builds, and unit testing.
CO 3	Apply Jenkins for job creation, triggers, plugins, and multi-node cluster setup.
CO 4	Analyze different delivery pipeline types, including blue-green deployments and canary releasing.
CO 5	Assess the role and importance of the Network Operations Center (NOC) in a DevOps environment.
CO6	Design engineering practices for continuous delivery and continuous development/integration.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1	A1	P1
C02	C3	A2	P2
CO3	C2	A3	P3
CO4	C4	A4	-
CO5	C5	-	P4
CO6	C6	A5	P5

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	-	-	2	-	-	-	-	1	-	3
CO2	1	3	3		1	-	-	-	-	2	-	3
CO3	1		3	2	1	-	-	-	-	1	-	3
CO4	-	-	3	1	2	-	-	-	-	-	-	3
CO5	-	2	3	-	2	-	-	-	-	1	-	3
CO6	1		3	2	1	-	-	-	-	1	-	3

CO-PSO Mapping

PO	PSO1	PSO2	POS3	PSO4	PSO5
CO1	3	3	-	-	1
CO2	1	3	1		1
CO3	2	1	-	1	1
CO4	-	1	-	1	1
CO5	1	2	-	-	-
CO6	2	1	-	1	-

Relevance of the Syllabus to various indicators

Unit I	**Introduction**
Local	Demonstrates how CI/CD benefits local teams by enabling efficient collaboration, faster feedback loops, and reduced integration issues.
Regional	Exploring the advantages of continuous deployment through the HP Laserjet case study showcases how regional businesses can achieve quicker time-to-market and competitive advantage.
National	Understanding CI/CD principles and implementation equips individuals with valuable skills sought by national tech industries, enhancing employability.
Global	Implementing CI/CD practices contributes to reducing waste and resource consumption in software development, aligning with global sustainability goals.
Employability	Knowledge of CI's benefits makes individuals more adaptable and valuable to diverse workplaces, enhancing their employability.
Entrepreneurship	The case study on HP Laserjet demonstrates how continuous deployment can foster innovation and responsiveness,

	important aspects for entrepreneurial success.
Skill Development	Learning the steps in CI/CD pipeline setup enhances practical skills in software development.
Professional Ethics	By fostering collaboration and improving communication in distributed teams, CI/CD can contribute to a more inclusive and diverse work environment, aligning with ethical considerations.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Stages of Continuous Integration and Delivery**
Local	-
Regional	-
National	-
Global	Understanding advanced CI processes and techniques like automated functional testing positions individuals and businesses competitively in the global tech market.
Employability	Acquiring skills in automated testing, code analysis, and version control enhances employability in software development roles.
Entrepreneurship	Learning about advanced CI processes like automated functional testing equips entrepreneurs with innovative approaches for software development.
Skill Development	Developing skills in automated unit testing, code analysis, and version control contributes to a well-rounded skill set for software development.
Professional Ethics	-
Gender	

Human Values	-
Environment & Sustainability	-
Unit III	**Jenkis**
Local	-
Regional	-
National	-
Global	Understanding advanced CI processes and techniques like automated functional testing positions individuals and businesses competitively in the global tech market.
Employability	Acquiring skills in automated testing, code analysis, and version control enhances employability in software development roles
Entrepreneurship	Learning about advanced CI processes like automated functional testing equips entrepreneurs with innovative approaches for software development.
Skill Development	Developing skills in automated unit testing, code analysis, and version control contributes to a well-rounded skill set for software development.
Professional Ethics	Emphasizing code review and code analysis aligns with professional ethics by ensuring high-quality software development.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Anatomy of a Continuous Delivery**
Local	-
Regional	-

National	-
Global	Knowledge of various delivery pipeline types and deployment strategies positions individuals and businesses competitively in the global tech market, fostering entrepreneurship.
Employability	Learning about different delivery pipelines and deployment strategies enhances employability by providing practical skills relevant to modern software development roles.
Entrepreneurship	Understanding concepts like blue-green deployments and canary releasing equips entrepreneurs with innovative approaches for efficient and reliable software deployment.
Skill Development	Exploring different types of delivery pipelines, rollback strategies, and deployment concepts contributes to a comprehensive skill set in software development.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**System Monitoring**
Local	-
Regional	-
National	-
Global	Knowledge of various delivery pipeline types and deployment strategies positions individuals and businesses competitively in the global tech market, fostering entrepreneurship.
Employability	Learning about different delivery pipelines and deployment strategies enhances employability by providing practical skills relevant to modern software development roles.
Entrepreneurship	Understanding concepts like blue-green deployments and canary releasing equips entrepreneurs with innovative

	approaches for efficient and reliable software deployment.
Skill Development	Exploring different types of delivery pipelines, rollback strategies, and deployment concepts contributes to a comprehensive skill set in software development.
Professional Ethics	Mastery of continuous delivery concepts aligns with professional ethics by emphasizing quality assurance and responsible deployment practices.
Gender	
Human Values	
Environment & Sustainability	
SDG	SDG 9, SDG 4, and SDG 8,
NEP 2020	-
POE/4th IR	CI/CD and Microservices align closely with the principles of the Fourth Industrial Revolution and the POE framework. They enable professionals to develop advanced skills, organizations to become more agile, and software development to be more ethical and efficient in the rapidly evolving technological landscape of the 4th IR.

CI/CD & MICROSERVICES LAB

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
CI/CD and Microservices Lab	**ENSP364**	**0-0-2**	**1**
Type of Course:	**Minor**		
Pre-requisite(s), if any: Nil			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Understand the principles of containerization and its role in microservices deployment.
CO 2	Demonstrate the ability to deploy microservices to a container orchestration platform.
CO 3	Create documentation outlining the steps for deploying and managing microservices in a production environment.
CO 4	Critically analyze the trade-offs between monolithic and microservices architectures.

Ex. No	Experiment Title	Mapped CO/COs
1	Set up a version control system (e.g., Git) repository, create branches, perform merges, and simulate a collaborative development environment.	CO1

2	Use a CI/CD tool (e.g., Jenkins, Travis CI) to automate code compilation, testing (unit, integration), and packaging.	CO1
3	Containerize an application using Docker, create Dockerfiles, and run containers locally.	CO2
4	Design a basic CI/CD pipeline with automated build, testing, and deployment stages.	CO2
5	Implement deployment strategies like blue-green deployment or canary releasing using CI/CD pipelines.	CO2
6	Use tools like Terraform or Ansible to provision and manage infrastructure as code for different environments.	CO2
7	Set up monitoring tools (e.g., Prometheus, Grafana) to monitor application performance and generate alerts.	CO2
8	Create a rollback mechanism in your CI/CD pipeline to revert changes in case of deployment issues.	CO3
9	Design a simple microservices architecture with multiple services communicating over APIs.	CO3
10	Implement different communication patterns between microservices (synchronous, asynchronous) using REST, gRPC, or message queues.	CO3
11	Deploy microservices using Kubernetes, create pods, services, and deploy updates without downtime.	CO3
12	Set up an API gateway (e.g., Kong, Apigee) to manage API access, security, and routing for your microservices.	CO4
13	Explore service discovery mechanisms (e.g., Consul, etcd) to dynamically locate and connect microservices.	CO2
14	Implement load balancing strategies to distribute traffic across instances of microservices.	CO3
15	Introduce resilience patterns like circuit breakers and retries to ensure robustness of microservices.	CO3
16	Implement logging, monitoring, and tracing for microservices to gain insights into their behavior and performance.	CO4

17	Dynamically scale individual microservices based on load using Kubernetes' autoscaling capabilities.	CO4
18	Implement security measures like JWT authentication and authorization for microservices.	CO3
19	Explore strategies for data management in microservices, including databases per service or shared databases.	CO4

COMPETITIVE CODING LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: **Competitive Programming Lab**	**Course Code**	**L-T-P**	**Credits**
	SEC036	0-0-4	2
Type of Course:	SEC		

Course Outcomes

CO1	**Demonstrate** proficiency in implementing and analyzing various algorithms and data structures commonly used in competitive programming.
CO2	**Develop** the ability to analyze problem statements, design efficient algorithms, and write optimized code to solve competitive programming problems within time and memory constraints.
CO3	**Cultivate** algorithmic thinking and problem-solving skills by identifying patterns, applying appropriate algorithms, and selecting optimal data structures for a given problem.
CO4	**Apply** strategies to optimize code and improve time and space complexity of solutions, considering factors such as algorithm selection, data structure usage, and efficient coding techniques.
CO5	**Gain** familiarity with different online competitive programming platforms, participate in coding competitions, and develop strong problem-solving and critical thinking skills in a competitive programming environment.

List of Experiments

S. No.	Experiment	COs
	Introduction to Competitive Coding • Overview of competitive coding and its importance in the field of	

1	computer science. • Understanding the significance of problem-solving skills and algorithmic thinking in competitive coding.	CO1
2	Data Structures and Algorithms • Review of fundamental data structures: arrays, linked lists, stacks, queues, trees, graphs, and hash tables. • Study of essential algorithms: searching, sorting, recursion, dynamic programming, greedy algorithms, and graph algorithms.	CO1
3	Time and Space Complexity Analysis • Understanding time and space complexity of algorithms. • Analysis of algorithm efficiency and choosing the most optimal solutions.	CO2
4	Problem Solving Techniques • Introduction to problem-solving techniques like brute force, divide and conquer, backtracking, and more. • Practice in applying different techniques to solve a variety of programming problems.	CO3
5	Advanced Data Structures • Study of advanced data structures: heaps, priority queues, segment trees, trie, and advanced graph structures. • Understanding the use of these data structures in solving complex programming problems.	CO4
6	Coding Paradigms • Introduction to different coding paradigms: procedural programming, object-oriented programming, and functional programming. • Understanding the benefits and drawbacks of each paradigm in competitive coding.	CO5
7	Online Judges and Contest Platforms • Familiarization with popular online judge platforms like Codeforces, Topcoder, and LeetCode. • Practice solving problems from online contests and participating in coding competitions. List of suggested links to coding platforms ▪ Codeforces: https://codeforces.com/ ▪ Topcoder: https://www.topcoder.com/ ▪ AtCoder: https://atcoder.jp/ ▪ LeetCode: https://leetcode.com/ ▪ HackerRank: https://www.hackerrank.com/ ▪ CodeChef: https://www.codechef.com/	CO5

	▪ HackerEarth: https://www.hackerearth.com/ ▪ Project Euler: https://projecteuler.net/ ▪ UVa Online Judge: https://onlinejudge.org/ ▪ SPOJ (Sphere Online Judge): https://www.spoj.com/ ▪ Google Code Jam: https://codingcompetitions.withgoogle.com/codejam ▪ Kick Start by Google: https://codingcompetitions.withgoogle.com/kickstart ▪ ACM ICPC Live Archive: https://icpcarchive.ecs.baylor.edu/ ▪ A2 Online Judge: https://a2oj.com/ ▪ CodeSignal: https://codesignal.com/	
8	Tips and Tricks for Competitive Coding • Learning effective coding techniques, shortcut methods, and best practices for competitive coding. • Developing strategies to optimize code, manage time, and improve problem-solving speed.	CO5
9	Mock Contests and Practice Sessions • Conducting mock contests and practice sessions to simulate real coding competitions. • Solving a wide range of problems to enhance coding skills and adaptability to different problem types.	CO5
10	**Self-Learning Component:** List of Suggested Competitive programming Courses: ▪ Competitive Programmer's Core Skills" by Coursera: This course covers fundamental algorithms and data structures used in competitive programming. Link: https://www.coursera.org/learn/competitive-programming-core-skills ▪ "Algorithms and Data Structures" by MIT OpenCourseWare: This course teaches essential algorithms and data structures for competitive programming. Link: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/ ▪ "Data Structures and Algorithms" by GeeksforGeeks: This course covers various data structures and algorithms commonly used in competitive programming. Link: https://practice.geeksforgeeks.org/courses/dsa-self-paced ▪ "Introduction to Competitive Programming" by NPTEL: This course introduces the basics of competitive programming and covers algorithms and problem-solving techniques. Link: https://onlinecourses.nptel.ac.in/noc21_cs07/ ▪ "Competitive Programming" by HackerRank: This course provides in-depth coverage of algorithms and data structures with hands-on coding exercises. Link:	CO5

	https://www.hackerrank.com/domains/tutorials/10-days-of-statistics ▪ "Advanced Data Structures and Algorithms" by Udemy: This course dives deeper into advanced data structures and algorithms for competitive programming. Link: https://www.udemy.com/course/advanced-data-structures-and-algorithms-in-java/ ▪ "Mastering Data Structures and Algorithms using C and C++" by Udemy: This course covers data structures and algorithms with a focus on problem-solving for coding interviews and competitive programming. Link: https://www.udemy.com/course/datastructurescncpp/ ▪ "Competitive Programming" by Coding Ninjas: This course provides comprehensive training in competitive programming, covering algorithms, data structures, and problem-solving techniques. Link: https://www.codingninjas.com/courses/online-competitive-programming-course ▪ "Algorithmic Toolbox" by Coursera: This course from the University of California San Diego covers algorithmic techniques and data structures for competitive programming. Link: https://www.coursera.org/learn/algorithmic-toolbox ▪ "Competitive Programming - From Beginner to Expert" by Udemy: This course offers a complete guide to competitive programming, starting from the basics and progressing to advanced topics. Link: https://www.udemy.com/course/competitive-programming-from-beginner-to-expert/ ▪ Competitive Programming Essentials, Master Algorithms 2022 (Udemy) https://www.udemy.com/course/competitive-programming-algorithms-coding-minutes/ ▪ The Bible of Competitive Programming & Coding Interviews **All students must complete one online course from the suggested programs*	

List of popular Competitive Programming Competitions:

1. ACM International Collegiate Programming Contest (ICPC): This is one of the most prestigious programming competitions for college students. Teams compete in solving a set of challenging algorithmic problems within a time limit. Website
2. Google Code Jam: Organized by Google, this annual coding competition challenges participants to solve algorithmic problems. It consists of multiple

online rounds leading to a final onsite competition. Website
3. Facebook Hacker Cup: This annual coding competition by Facebook features multiple online rounds and an onsite final round. Participants solve algorithmic problems for a chance to win prizes. Website
4. Topcoder Open: Topcoder hosts this annual programming competition featuring algorithmic and design challenges. Participants compete for cash prizes and a chance to be recognized by industry experts. Website
5. International Olympiad in Informatics (IOI): IOI is an annual international programming competition for high school students. Participants solve algorithmic problems in a contest format. Website
6. AtCoder Grand Contest: AtCoder hosts this regular contest series featuring algorithmic programming challenges. Participants can compete individually or as a team. Website
7. Codeforces: Codeforces is a popular competitive programming platform that hosts regular contests. Participants compete in solving algorithmic problems and earn ratings based on their performance. Website
8. LeetCode Weekly Contests: LeetCode organizes weekly contests where participants can solve algorithmic problems and compete for rankings. Website
9. HackerRank Contests: HackerRank hosts various contests and challenges covering a wide range of programming topics. Participants can compete individually or as part of a team. Website
10. Kaggle Competitions: Kaggle is a platform for data science competitions, where participants solve real-world problems using machine learning and data analysis techniques. Website

All students must participate in some competitions

Suggested Books

1. "Competitive Programming 3" by Steven Halim and Felix Halim: This book is a comprehensive guide to competitive programming, covering algorithms, data structures, problem-solving techniques, and contest strategies. It includes numerous examples, explanations, and practice problems. Book Link
2. "Algorithms" by Robert Sedgewick and Kevin Wayne: This book provides a thorough introduction to algorithms, including sorting, searching, graph algorithms, and dynamic programming. It includes detailed explanations, visualizations, and implementation examples. Book Link

3. "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein: Known as "CLRS," this book is a classic reference for algorithms. It covers a wide range of algorithms, data structures, and algorithm design techniques. Book Link
4. "Programming Challenges" by Steven S. Skiena and Miguel A. Revilla: This book presents a collection of programming problems from various competitions and online judges. It provides problem-solving techniques, algorithmic approaches, and example solutions. Book Link
5. "The Art of Computer Programming" by Donald E. Knuth: This multi-volume series is considered a classic in computer science. It covers various algorithms, data structures, and mathematical techniques in great detail. Book Link
6. "Cracking the Coding Interview" by Gayle Laakmann McDowell: Although not specifically focused on competitive programming, this book is a popular resource for coding interview preparation. It covers essential data structures, algorithms, and problem-solving techniques. Book Link
7. "Programming Pearls" by Jon Bentley: This book presents a collection of programming challenges and discusses techniques for solving them efficiently. It emphasizes problem-solving skills and algorithmic thinking. Book Link

Web References

- https://www.geeksforgeeks.org/competitive-programming-a-complete-guide/
- https://www.geeksforgeeks.org/must-do-coding-questions-for-companies-like-amazon-microsoft-adobe/
- https://www.udemy.com/course/competitive-programming
- https://github.com/smv1999/CompetitiveProgrammingQuestionBank
- https://github.com/parikshit223933/Coding-Ninjas-Competitive-Programming
- https://www.hackerearth.com/getstarted-competitive-programming/
- https://www.csestack.org/competitive-coding-questions/

List of Suggested Experiments in Lab Sessions

Questions on Arrays
1. Maximum Subarray Sum: Given an array of integers, find the contiguous subarray with the largest sum.
2. Two Sum: Given an array of integers and a target value, find two numbers in the array that add up to the target.
3. Rotate Array: Rotate an array of n elements to the right by k steps.
4. Merge Intervals: Given a collection of intervals, merge overlapping intervals.
5. Majority Element: Find the majority element in an array. The majority element

appears more than n/2 times, where n is the size of the array.
6. Trapping Rain Water: Given an array representing the heights of bars, calculate the amount of water that can be trapped between the bars.
7. Next Permutation: Implement the next permutation algorithm to find the lexicographically next greater permutation of an array of integers.
8. Subarray with Given Sum: Given an unsorted array of non-negative integers and a target sum, find a subarray that adds up to the target sum.
9. Product of Array Except Self: Given an array of n integers, return an array output such that each element at index i of the output array is the product of all the elements in the original array except the one at i.
10. Minimum Size Subarray Sum: Given an array of positive integers and a target sum, find the minimum length of a contiguous subarray whose sum is greater than or equal to the target sum.
Questions on Recursion
1. Factorial: Write a recursive function to calculate the factorial of a given number.
2. Fibonacci Series: Write a recursive function to generate the nth term of the Fibonacci series.
3. Power of a Number: Write a recursive function to calculate the power of a given number.
4. Sum of Digits: Write a recursive function to find the sum of digits of a given number.
5. Palindrome Check: Write a recursive function to check whether a given string is a palindrome or not.
6. Tower of Hanoi: Solve the Tower of Hanoi problem using recursion.
7. Binary Search: Implement a recursive binary search algorithm to find an element in a sorted array.
8. Permutations: Write a recursive function to generate all permutations of a given string.
9. Subset Sum: Given an array of integers and a target sum, write a recursive function to check if there exists a subset that sums up to the target.
10. Combination Sum: Given an array of integers and a target sum, write a

recursive function to find all possible combinations that sum up to the target.
Questions on Stacks & Queues:
1. Balanced Parentheses: Given a string of parentheses, write a function to determine if the parentheses are balanced using a stack.
2. Reverse a String: Write a function to reverse a string using a stack.
3. Evaluate Postfix Expression: Given a postfix expression, write a function to evaluate it using a stack.
4. Next Greater Element: Given an array, find the next greater element for each element in the array using a stack.
5. Largest Rectangle in Histogram: Given a histogram represented by an array of bar heights, find the largest rectangle that can be formed in the histogram using a stack.
6. Implement Stack using Queues: Implement a stack data structure using queues.
7. Implement Queue using Stacks: Implement a queue data structure using stacks.
8. Sliding Window Maximum: Given an array and an integer k, find the maximum element in each sliding window of size k using a queue.
9. Print Binary Tree in Level Order: Given a binary tree, print its elements in level order using a queue.
10. Implement Recent Counter: Design a data structure that counts the number of recent requests within a certain time range using a queue.
Questions on Linked Lists
1. Reverse a Linked List: Write a function to reverse a singly linked list.
2. Detect Cycle in a Linked List: Write a function to detect if a linked list contains a cycle.
3. Find the Middle of a Linked List: Write a function to find the middle node of a linked list.
4. Merge Two Sorted Lists: Given two sorted linked lists, write a function to merge them into a single sorted linked list.

5. Remove Nth Node from End of List: Given a linked list, remove the nth node from the end of the list and return its head.
6. Intersection of Two Linked Lists: Given two linked lists, write a function to find the intersection point if it exists.
7. Palindrome Linked List: Given a singly linked list, determine if it is a palindrome.
8. Remove Duplicates from Sorted List: Given a sorted linked list, remove duplicates from it.
9. Add Two Numbers as Linked Lists: Given two linked lists representing two numbers, write a function to add them and return the resulting linked list.
10. Flatten a Multilevel Linked List: Given a linked list with a special structure, flatten it into a single-level linked list.
Questions on Trees
1. Binary Tree Traversals: Implement different tree traversal algorithms such as in-order, pre-order, and post-order traversal.
2. Maximum Depth of Binary Tree: Find the maximum depth or height of a binary tree.
3. Validate Binary Search Tree: Given a binary tree, check if it is a valid binary search tree.
4. Lowest Common Ancestor of Two Nodes: Find the lowest common ancestor of two nodes in a binary tree.
5. Diameter of Binary Tree: Find the diameter of a binary tree, which is the longest path between any two nodes.
6. Binary Tree Level Order Traversal: Traverse a binary tree in level order and return the nodes in each level.
7. Symmetric Tree: Check if a binary tree is symmetric, meaning it is a mirror image of itself.
8. Serialize and Deserialize Binary Tree: Design algorithms to serialize and deserialize a binary tree.
9. Count Complete Tree Nodes: Count the number of nodes in a complete binary tree.
10. Construct Binary Tree from Preorder and Inorder Traversal: Given the

preorder and inorder traversal of a binary tree, construct the tree.
Questions on Graphs
1. Shortest path: Find the shortest path between two vertices in a graph. This can be solved using Dijkstra's algorithm or Bellman-Ford's algorithm.
2. Maximum flow: Find the maximum flow from one vertex to another in a graph. This can be solved using the Ford-Fulkerson algorithm or the Dinic algorithm.
3. Minimum spanning tree: Find the minimum spanning tree of a graph. This can be solved using Prim's algorithm or Kruskal's algorithm.
4. Topological sorting: Find a topological ordering of a graph. This can be solved using Kahn's algorithm.
5. Strongly connected components: Find the strongly connected components of a graph. This can be solved using Tarjan's algorithm.
6. Bipartite matching: Find a maximum bipartite matching in a graph. This can be solved using the Hungarian algorithm.
7. Traveling salesman problem: Find the shortest tour that visits all the vertices in a graph. This is an NP-hard problem, but there are approximation algorithms that can be used to find a good solution.
Time & Space Complexity
1. Time Complexity Analysis: Analyze the time complexity of a given algorithm or piece of code.
2. Space Complexity Analysis: Analyze the space complexity of a given algorithm or piece of code.
3. Big O Notation: Given a function or algorithm, determine its big O notation in terms of time or space complexity.
4. Best/Worst/Average Case Complexity: Analyze the best, worst, and average-case time or space complexity of an algorithm.
5. Sorting Algorithms: Implement and analyze the time complexity of various sorting algorithms such as Bubble Sort, Insertion Sort, Merge Sort, Quick Sort, and Heap Sort.
6. Searching Algorithms: Implement and analyze the time complexity of various searching algorithms such as Linear Search, Binary Search, and Hashing.
7. Dynamic Programming: Solve dynamic programming problems and analyze

their time and space complexity.
8. Recursion vs. Iteration: Compare and analyze the time and space complexity of recursive and iterative solutions for a given problem.
9. Complexity Trade-offs: Analyze and compare the time and space complexity trade-offs of different algorithms for the same problem.
10. Space-Optimized Data Structures: Implement and analyze space-optimized data structures such as Bit Arrays, Bloom Filters, or Space-Efficient Hash Tables.
Questions on Divide & Conquer Strategy
1. Binary Search: Implement a recursive binary search algorithm to find an element in a sorted array.
2. Merge Sort: Implement the Merge Sort algorithm to sort an array of integers.
3. Quick Sort: Implement the Quick Sort algorithm to sort an array of integers.
4. Count Inversions: Given an array of integers, find the number of inversions present using the Divide and Conquer approach.
5. Closest Pair of Points: Given a set of points in a 2D plane, find the pair of points with the smallest distance between them using the Divide and Conquer technique.
6. Maximum Subarray Sum: Given an array of integers, find the maximum sum of a subarray using the Divide and Conquer approach.
7. Matrix Multiplication: Implement a Divide and Conquer algorithm to multiply two matrices efficiently.
8. Finding Majority Element: Given an array of integers, find the majority element (appearing more than n/2 times) using the Divide and Conquer technique.
9. Finding Kth Smallest Element: Given an array of integers, find the kth smallest element using the Divide and Conquer approach.
10. Closest Pair Sum: Given two sorted arrays and a target value, find the pair of elements (one from each array) with the closest sum to the target using the Divide and Conquer technique.

Questions on Dynamic Programming
1. Fibonacci Series: Implement the Fibonacci series using dynamic programming to efficiently calculate the nth term.
2. Longest Common Subsequence: Given two strings, find the length of the longest common subsequence using dynamic programming.
3. Knapsack Problem: Given a set of items with weights and values, determine the maximum value that can be obtained by selecting a subset of items within a weight limit using dynamic programming.
4. Coin Change Problem: Given a set of coin denominations and a target value, find the minimum number of coins needed to make the target value using dynamic programming.
5. Rod Cutting Problem: Given a rod of a certain length and a price list for different rod lengths, find the maximum value that can be obtained by cutting and selling the rod using dynamic programming.
6. Edit Distance: Given two strings, find the minimum number of operations (insertion, deletion, and substitution) required to convert one string into another using dynamic programming.
7. Maximum Subarray Sum: Given an array of integers, find the maximum sum of a subarray using dynamic programming.
8. Longest Increasing Subsequence: Given an array of integers, find the length of the longest increasing subsequence using dynamic programming.
9. Matrix Chain Multiplication: Given a sequence of matrices, find the minimum number of scalar multiplications needed to multiply them using dynamic programming.
10. Subset Sum Problem: Given a set of integers and a target sum, determine if there exists a subset that sums up to the target using dynamic programming.
Questions on Greedy Programming
1. Fractional Knapsack Problem: Given a set of items with weights and values, determine the maximum value that can be obtained by selecting fractions of items within a weight limit using a greedy algorithm.
2. Activity Selection Problem: Given a set of activities with start and finish times, select the maximum number of activities that can be performed without overlapping using a greedy algorithm.

3. Minimum Spanning Tree: Given a weighted graph, find the minimum spanning tree using Kruskal's or Prim's algorithm, which are both based on greedy approaches.
4. Huffman Coding: Given a set of characters and their frequencies, construct a binary code that minimizes the total encoded length using a greedy algorithm.
5. Coin Change Problem: Given a set of coin denominations and a target value, find the minimum number of coins needed to make the target value using a greedy algorithm.
6. Job Scheduling Problem: Given a set of jobs with their deadlines and profits, schedule the jobs to maximize the total profit using a greedy algorithm.
7. Interval Scheduling Problem: Given a set of intervals, select the maximum number of non-overlapping intervals using a greedy algorithm.
8. Dijkstra's Algorithm: Given a weighted graph, find the shortest path from a source vertex to all other vertices using Dijkstra's algorithm, which is based on a greedy approach.
9. Egyptian Fraction: Given a fraction, represent it as a sum of unique unit fractions using a greedy algorithm.
10. Car Fueling Problem: Given the total distance to be covered, the capacity of the fuel tank, and a list of distances between fuel stations, determine the minimum number of refuelings needed to reach the destination using a greedy algorithm.
Questions on String Matching
1. Naive String Matching: Implement the naive string matching algorithm to find all occurrences of a pattern in a text.
2. Knuth-Morris-Pratt (KMP) Algorithm: Implement the KMP algorithm to efficiently find all occurrences of a pattern in a text.
3. Rabin-Karp Algorithm: Implement the Rabin-Karp algorithm to efficiently find all occurrences of a pattern in a text using hashing.
4. Longest Common Substring: Given two strings, find the longest common substring using dynamic programming or other efficient algorithms.
5. Longest Common Prefix: Given an array of strings, find the longest common prefix using a suitable algorithm.
6. Regular Expression Matching: Implement a regular expression matching

algorithm to determine if a string matches a given pattern.
7. Anagrams: Given a list of strings, find all pairs of strings that are anagrams of each other.
8. Palindromic Substrings: Given a string, find all palindromic substrings using a suitable algorithm.
9. Boyer-Moore Algorithm: Implement the Boyer-Moore algorithm to efficiently find all occurrences of a pattern in a text.
10. Subsequence Matching: Given two strings, determine if one string is a subsequence of the other.
Questions on Advanced Data Structures
1. Trie: Implement a Trie data structure and solve problems such as word search, autocomplete, or finding the longest common prefix.
2. Segment Tree: Implement a Segment Tree data structure and solve problems such as range sum queries, range minimum/maximum queries, or range updates.
3. Fenwick Tree (Binary Indexed Tree): Implement a Fenwick Tree data structure and solve problems such as prefix sum queries or range updates.
4. Disjoint Set Union (DSU) / Union-Find: Implement a DSU data structure and solve problems such as connected components, cycle detection, or Kruskal's algorithm for finding the minimum spanning tree.
5. Treap: Implement a Treap (a balanced binary search tree with randomized priorities) and solve problems such as maintaining the median of a dynamic set of numbers or solving range queries on a set of intervals.
6. Suffix Array: Implement a Suffix Array data structure and solve problems such as finding the longest common substring, finding the lexicographically smallest substring, or pattern matching.
7. LCA (Lowest Common Ancestor): Implement an LCA data structure and solve problems such as finding the lowest common ancestor of two nodes in a tree or solving distance-related queries on a tree.
8. K-D Tree: Implement a K-D Tree data structure and solve problems such as nearest neighbor search or range search in a multi-dimensional space.
9. AVL Tree or Red-Black Tree: Implement a balanced binary search tree (either AVL Tree or Red-Black Tree) and solve problems such as maintaining a sorted

dynamic set or solving range queries.
10. B+ Tree: Implement a B+ Tree data structure and solve problems such as indexing or range queries on a large dataset.

References to Interview Questions

- https://www.simplilearn.com/coding-interview-questions-article
- https://www.csestack.org/competitive-coding-questions/
- https://www.geeksforgeeks.org/a-competitive-programmers-interview/
- https://www.geeksforgeeks.org/must-do-coding-questions-for-companies-like-amazon-microsoft-adobe/
- https://unstop.com/blog/competitive-coding-questions-with-solutions

https://unstop.com/blog/competitive-coding-questions-with-solutions

MINOR PROJECT-II

Department:	**Department of Computer Science and Engineering**		
Course Name: Minor Project-II	**Course Code**	**L-T-P**	**Credits**
	ENSI352	---	2
Type of Course:	Project		
Pre-requisite(s), if any: NA			

- Students expected to develop a basic project that demonstrates the application of learnings from studied subjects.
- Students are required to submit a hard copy of project file as per the template (Provided at the end of Handbook). File needs to be submitted in spiral bind.
- Project will be evaluated on the scale of 100 with following evaluation criteria.
    - Project idea & features (10)
    - Literature review (10)
    - Tools & Techniques employed (10)
    - Methodology (10)
    - Presentation of Results and its usefulness (20)
    - Implementation and its understandability (10)
    - Meetings & comments by guide (20)
    - Research paper (10)

File format for Minor project

1.	Abstract	Page No.
2.	Introduction (description of broad topic)	
3.	Motivation	
4.	Literature Review	
5.	Gap Analysis	
6.	Problem Statement	
7.	Objectives	

8.	Tools/platform used	
9.	Methodology	
10.	Experimental Setup	
11	Evaluation Metrics	
12	Results And Discussion	
13	Conclusion & Future Work	
14	References	
15	Annexure I: Responsibility Chart	
16	Annexure II: Screenshots of all the MS-Team Meetings with links (online)/ handwritten comments(offline) from guide	
17	Annexure III Complete implementation code	
18	Annexure IV Research Paper (Published/Submitted)	

Semester: 7

(DEPARTMENT ELECTIVE-I)

SECURE CODING AND VULNERABILITIES

Department:	**Department of Computer Science and Engineering**		
Course Name: **Secure Coding & Vulnerabilities**	**Course Code** **ENSP301**	**L-T-P** 4-0-0	**Credits** 4
Type of Course:	Minor		
Pre-requisite(s), if any:			

Brief Syllabus:

Secure Coding and Vulnerabilities is a comprehensive course that focuses on understanding and mitigating application security threats and attacks. The course covers various aspects such as security requirements gathering, secure application design and architecture, and secure coding practices for input validation, authentication, cryptography, session management, and error handling. Students will learn about common application vulnerabilities and the potential consequences of security breaches. They will also gain knowledge and skills in static and dynamic application security testing methods, as well as secure deployment and maintenance practices. The course aims to equip students with the necessary tools and techniques to develop robust and secure applications while adhering to best practices in the field of application security.

UNIT WISE DETAILS

Unit Number: 1	**Title: Introduction to coding and Security**	**No. of hours: 12**

Content Summary:

Introduction-security concepts-CIA Triad, Viruses, Trojans, and Worms, threat, vulnerability, risk, attack. Coding Standards: Dirty Code and Dirty Compiler, Dynamic Memory Management functions, Common memory management Errors (Initialization Errors, Forget to Check Return Values, accessing already freed memory, Freeing the same memory multiple times, Forget to free the allocated memory), Integer Security

–Introduction to integer types: Integer Data Types, data type conversions, Integer vulnerabilities and mitigation strategies

Unit Number: 2	Title: Secure Application Design and Architecture	No. of hours: 8

Content Summary:

Security requirements gathering and analysis, Secure software development life cycle (SSDLC) , Security issues while writing SRS, Design phase security, Development Phase, Test Phase, Maintenance Phase, Writing Secure Code – Best Practices SD3 (Secure by design, default and deployment), Security principles and Secure Product Development Timeline.

Unit Number: 3	Title: Secure Coding Practices and Vulnerabilities	No. of hours: 8

Content Summary:

Input validation Techniques-whitelist validation, regular expressions, authentication and authorization, Cryptography, buffer overflows, Session management and protection against session-related attacks, Secure error handling and logging practices, SQL Injection Techniques and Remedies, Race conditions

Unit Number: 4	Title: Application Security Testing and Deployment	No. of hours: 12

Content Summary:

Security code overview, Secure software installation. The Role of the Security Tester, Building the Security Test Plan. Testing HTTP-Based Applications, Testing File-Based Applications, Testing Clients with Rogue Servers, Static and Dynamic Application Security Testing (SAST & DAST), Secure Deployment and Maintenance, Patch management and software updates, Vulnerability scanning and penetration testing.

***Self-Learning Components**: **mention 4-5 topics for students in bullet points**

Please Note:

1) Code Review Tools: Students can explore open-source code review tools such as SonarQube, ESLint, or FindBugs to understand how these tools can help identify security vulnerabilities in code.
2) Security Frameworks such as OWASP (Open Web Application Security Project) and their associated resources.
3) Secure Development Tools: Students can explore tools like Burp Suite, ZAP (Zed Attack Proxy), or WebInspect to understand how these tools can be used

for dynamic application security testing (DAST) and penetration testing.

4) Secure Coding in Web Applications: Students can dive deeper into web application security topics, such as Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), or security measures like Content Security Policy (CSP) and HTTP security headers.

Reference Books:

- Writing Secure Code, Michael Howard and David LeBlanc,Microsoft Press, 2nd Edition, 2004
- Buffer Overflow Attacks: Detect, Exploit, Prevent by Jason Deckard ,Syngress,1st Edition, 2005
- Threat Modeling, Frank Swiderski and Window Snyder,Microsoft Professional, 1st Edition ,2004
- Secure Coding: Principles and Practices by Mark G. Graff, Kenneth R. van Wyk, Publisher(s): O'Reilly Media, Inc., 2003
- The Software Vulnerability Guide (Programming Series) by H. Thompson (Author), Scott G. Chase, 2005

Reference Links:

- **"Secure Coding Practices" on Udemy - Offered by The App Brewery. Link: https://www.udemy.com/course/secure-coding-practices/**
- "**Secure Coding: Preventing Software Vulnerabilities" on Pluralsight - Offered by Pluralsight**. Link: Secure Coding: Preventing Software Vulnerabilities
- "**Software Security" on edX - Offered by University of Maryland, College Park**. Link: Software Security
- Identifying Security Vulnerabilities in C/C++Programming | Coursera
- Principles of Secure Coding | Coursera
- Identifying Security Vulnerabilities | Coursera

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** different types of application security threats and their potential impact.
CO2	**Apply** secure design principles and architectures to develop robust and secure applications.

CO3	**Implement** secure coding practices for input validation, authentication, cryptography, session management, and error handling.
CO4	**Conduct** static and dynamic application security testing to identify vulnerabilities and implement secure deployment and maintenance practices.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A2	P2
CO2	C3	A3	P3
CO3	C3	A3	P3
CO4	C4	A4	P4

CO-PO Mapping

CO-PO	PO 1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2	1	1	2	2	1	2	2	2	2	3
CO2	3	2	3	2	2	2	2	3	3	3	3	3
CO3	3	3	3	2	2	2	2	3	3	3	3	3
CO4	2	2	1	3	2	2	2	2	3	2	3	3

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

CO-PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	2	1	2	1
CO2	3	2	2	2	2
CO3	2	3	3	1	1
CO4	2	3	2	3	2

Relevance of the Syllabus to various indicators

Unit I	**Introduction to coding and Security**
Local	-
Regional	-
National	-
Global	Enhancing cybersecurity capabilities and promoting secure software development.
Employability	-
Entrepreneurship	-
Skill Development	Developing coding skills with a focus on security.
Professional Ethics	Develop applications that prioritize data security and user privacy.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Secure Application Design and Architecture**

Local	Applying security requirements and following secure software development life cycle (SSDLC) practices aligns with local needs to protect sensitive data and ensure secure application design.
Regional	-
National	-
Global	Promotes global standards in application security, fostering a global culture of secure software development.
Employability	Proficiency in secure application design and adherence to security principles enhances students' employability in organizations seeking professionals with secure development skills
Entrepreneurship	-
Skill Development	Enhances students' skills in secure software design, architecture, and implementation, preparing them to tackle application security challenges.
Professional Ethics	Ethical considerations of data privacy and security, fostering responsible development practices.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Secure Coding Practices and Vulnerabilities**
Local	Understanding input validation techniques, cryptography, and secure error handling directly addresses local needs for developing secure applications and mitigating common vulnerabilities.
Regional	-
National	Addressing vulnerabilities such as SQL injection and race conditions through secure coding practices aligns with national objectives of securing critical applications and preventing

	cyber-attacks.
Global	Knowledge of secure coding practices and vulnerability mitigation strategies helps establish global standards for secure software development and promotes a secure digital environment worldwide.
Employability	-
Entrepreneurship	-
Skill Development	Hones students' skills in secure coding, vulnerability identification, and remediation, enhancing their technical capabilities in application security.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Application Security Testing and Deployment**
Local	-
Regional	-
National	-
Global	Understanding static and dynamic application security testing methods and secure deployment practices promotes global standards for secure software deployment and mitigating application vulnerabilities.
Employability	Expertise in application security testing and secure deployment enhances students' employability in roles focused on ensuring application security and secure software deployment.
Entrepreneurship	-
Skill Development	Develops students' skills in application security testing, penetration testing, and secure deployment, equipping them

	with practical expertise in securing applications.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG-4,9,16
NEP 2020	Skill development, employability, and entrepreneurship
POE/4th IR	Emphasizes the importance of cybersecurity in the digital era.

SECURE CODING AND VULNERABILITIES LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: **Secure Coding & Vulnerabilities Lab**	**Course Code**	**L-T-P**	**Credits**
	ENSP351	0-0-2	1
Type of Course:	Minor		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	**Apply** Secure Coding Techniques for enhancing application security.
CO 2	**Analyze** and Evaluate Security Vulnerabilities
CO 3	**Evaluate** and Communicate Importance of Secure Coding by analyzing potential vulnerabilities
CO 4	**Design** and Implement Secure Applications using secure cryptographic libraries.

Ex. No	Experiment Title	Mapped CO/COs
1	Write code to convert between different data types	CO1
2	Implement dynamic memory allocation and deallocation operations and analyze potential errors and vulnerabilities.	CO2

3	Write code snippets with initialization errors, memory leaks, and double free issues, and use tools like Valgrind to detect and fix these errors.	CO2
4	Analyze a given code snippet with dirty code practices	CO2
5	Perform static code analysis on a sample codebase using a secure coding tool SonarQube	CO3
6	Conduct dynamic code analysis on a web application using OWASP	CO2
7	Configure the tool to intercept and analyze HTTP requests and responses.	CO2
8	Implement encryption algorithms (e.g., AES) using secure cryptographic libraries or frameworks.	CO4
9	Implement and test whitelist validation techniques to ensure secure input handling.	CO1
10	Develop a simple web application that requires user authentication.	CO4
11	Implement cryptographic functions for secure data protection	CO4
12	Implement input sanitization and validation techniques to prevent SQL injection attacks.	CO1
13	Conduct a security audit and penetration testing on a provided application to identify SQL injection vulnerabilities.	CO2
14	Identify common memory management errors such as forgetting to check return values or accessing already freed memory.	CO2
15	Write a sample code that requires input validation, such as user input or data from external sources.	CO1
16	Write a sample code that involves cryptographic operations, such as encryption or hashing.	CO4
17	Conduct security testing on an HTTP-based application to identify vulnerabilities and security weaknesses.	CO2

18	Set up a local or web-based application that operates over HTTP. Perform security testing using appropriate tools and techniques, such as vulnerability scanners and penetration testing	CO2
19	Perform security testing on a file-based application to assess its security posture and identify potential vulnerabilities.	CO2
20	Utilize appropriate tools and techniques to conduct static analysis on the application's source code to identify potential vulnerabilities	CO3
21	Identify and configure important HTTP security headers, such as Strict-Transport-Security (HSTS), X-Frame-Options, X-XSS-Protection, and X-Content-Type-Options.	CO1
22	Develop a sample web application that includes error handling and logging functionality.	CO4
23	Implement secure error handling techniques, such as displaying generic error messages to users and logging detailed errors only to authorized personnel.	CO4
24	Apply secure coding best practices, such as input validation, output encoding, proper error handling, and secure use of APIs and libraries.	CO1
25	Test the code for vulnerabilities and discuss the importance of writing secure code to prevent potential exploitation.	CO3

Projects

- Implement a secure software development lifecycle
  http://www.owasp.org/index.php/Category:OWASP_CLASP_Project
- Establish secure coding standards
  http://www.owasp.org/index.php/Category:OWASP_Guide_Project
- Build a re-usable object library
  http://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API
- Verify the effectiveness of security controls

http://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project

- Establish secure outsourced development practices including defining security requirements and verification methodologies in both the request for proposal (RFP) and contract.
  http://www.owasp.org/index.php/Category:OWASP_Legal_Project

CYBER CRIME INVESTIGATION & DIGITAL FORENSICS

Department:	**Department of Computer Science and Engineering**		
Course Name: **Cyber Crime Investigation & Digital Forensics**	**Course Code**	**L-T-P**	**Credits**
	ENSP303	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			
Brief Syllabus: Introduces the principles and practices of digital forensics including digital investigations, data and file recovery methods, and digital forensics analysis and invalidation. Topics include data acquisition, digital forensics tools, virtual machines, network, mobile devices and cloud forensics.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Title: Introduction**	**No. of hours: 8**	
Content Summary: Introduction to Digital Forensics, Definition and types of cybercrimes, electronic evidence and handling, electronic media, collection, searching and storage of electronic media, introduction to internet crimes, hacking and cracking, credit card and ATM frauds, web technology, cryptography, emerging digital crimes and modules.			
Unit Number: 2	**Title: Types of Cyber Crimes**	**No. of hours: 10**	
Content Summary: Crimes targeting Computers: Unauthorized Access Packet Sniffing Malicious Codes including Trojans, Viruses, Logic Bombs, etc. Online based Cyber Crimes: Phishing and its variants Web Spoofing and E-mail Spoofing Cyber Stalking			

Web defacement Financial crimes, ATM and Card Crimes etc Spamming Commercial espionage and Commercial Extortion online Software and Hardware Piracy Money Laundering Fraud& Cheating Other Cyber Crimes.

Unit Number: 3	Title: Investigation of Cyber Crimes	No. of hours: 12

Content Summary:

Investigation of malicious applications Agencies for investigation in India, their powers and their constitution as per Indian Laws Procedures followed by First Responders; Evidence Collection and Seizure Procedures of Digital mediums Securing the Scene, Documenting the Scene, Evidence Collection and Transportation Data Acquisition Data Analysis Reporting

Unit Number: 4	Title : Forensic Tools and Processing of Electronic Evidence	No. of hours: 10

Content Summary:

Introduction to Forensic Tools, Usage of Slack space, tools for Disk Imaging, Data Recovery, Vulnerability Assessment Tools, Encase and FTK tools, Anti Forensics and probable counters, retrieving information, process of computer forensics and digital investigations, processing of digital evidence, digital images, damaged SIM and data recovery, multimedia evidence, retrieving deleted data: desktops, laptops and mobiles, retrieving data from slack space, renamed file, ghosting, compressed files.

***Self-Learning Components:**

1. **Open-Source Digital Forensics Tools: Introduction to popular open-source digital forensics tools such as Autopsy, Sleuth Kit, and Volatility.**
2. **Exploring open-source threat intelligence platforms like MISP and AlienVault OTX.**
3. **Discussing the benefits of information sharing and collaborative efforts in combating cyber threats.**
4. **Digital Forensics and Cyber-Crime Investigation, https://www.udemy.com/course/digital-forensics-and-cyber-crime-investigation/**

Reference Books:

- Moore, Robert, (2011). Cybercrime, investigating high-technology computer crime(2nd Ed.). Elsevie
- C. Altheide& H. Carvey Digital Forensics with Open Source Tools, Syngress, 2011.
- Majid Yar, "Cybercrime and Society", SAGE Publications Ltd, Hardcover, 2nd Edition, 2013.
- Robert M Slade, "Software Forensics: Collecting Evidence from the Scene of a Digital Crime", Tata McGraw Hill, Paperback, 1st Edition, 2004.

Web references:

- https://www.coursera.org/learn/digital-forensics-concepts
- https://www.udemy.com/course/computer-forensics-and-digital-forensics-for-everyone/

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** the nature and classification of conventional and cyber-crimes.
CO2	**Analyze** and **identify** various types of cyber-crimes and their modes of operation.
CO3	**Evaluate** the impact of cyber-crimes on individuals, organizations, and society**.**
CO4	**Develop** an understanding of digital forensics and the investigative procedures used in cyber-crime cases.
CO5	**Apply** forensic tools and techniques to retrieve and analyze digital evidence.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A1	-
C02	C3	-	-
CO3	C4	A2	-
CO4	C5	-	P5
CO5	C6	-	P2

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1	1	-	2	2	-	2	2	1	-	1
CO2	-	3	3	-	3	2	-	2	3	-	-	1
CO3	-	1	3	-	3	2	2	3	3	-	-	2
CO4	2	2	2	3	3	3	2	2	2	-	-	1
CO5	-	2	2	3	3	3	-	2	2	-	-	2

Justification for mapping must be relevant

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PSO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	1	1	1
CO2	3	1	2	1	1
CO3	2	1	3	2	1
CO4	2	2	2	3	-
CO5	3	1	2	2	-

Relevance of the Syllabus to various indicators

Unit I	**Introduction**
Local	The knowledge and understanding of cybercrime and computer crime can help local communities and law enforcement agencies address and prevent such crimes in their area.
Regional	Cybercrime is a regional concern, and understanding its types and emerging trends can help in regional collaboration for combating cyber threats.
National	Cybercrime is a significant concern at the national level. Developing expertise in digital forensics and cybercrime investigation can enhance national security and protect critical infrastructure.
Global	Cybercrime has a global impact, and knowledge in this area can contribute to international efforts in combating cyber threats and promoting cybersecurity.
Employability	The skills and knowledge gained in this unit can enhance employability in the field of cybersecurity, law enforcement, digital forensics, and related industries.
Entrepreneurship	-
Skill	Developing skills in digital forensics, evidence handling, and

Development	understanding emerging digital crimes, contributing to skill development in the field.
Professional Ethics	Studying cybercrime and computer crime can raise awareness of ethical issues related to information security, privacy, and responsible use of technology.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Types of Cyber Crimes**
Local	-
Regional	-
National	Cyber crimes pose significant challenges to national security and the economy. Understanding different types of cyber crimes allows governments and law enforcement agencies to develop robust policies, laws, and strategies to address cyber threats at the national level.
Global	Cyber crimes have a global reach and impact. By studying the types of cyber crimes, individuals and organizations can contribute to global efforts in promoting cybersecurity, sharing threat intelligence, and developing international frameworks to combat cyber threats.
Employability	Acquiring knowledge about various types of cybercrimes enhances employability in the field of cybersecurity.
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in internet technologies and network protocols
Professional Ethics	Awareness of different types of cybercrimes raises ethical considerations surrounding privacy, data protection, and responsible use of technology.
Gender	-

Human Values	-
Environment & Sustainability	-
Unit III	**Investigation of Cyber Crimes**
Local	-
Regional	Collaboration among regional investigation agencies can be improved through the knowledge of investigation procedures and digital evidence handling.
National	Investigating cyber crimes is a critical aspect of national security, and this unit's content can enhance the investigation capabilities of agencies at the national level.
Global	Aligns with global Cooperation and sharing of best practices in cybercrime investigation.
Employability	Proficiency in cybercrime investigation and evidence handling is in high demand, offering employment opportunities in the field of digital forensics and cybersecurity.
Entrepreneurship	Knowledge in cybercrime investigation can inspire entrepreneurs to develop innovative tools and services for digital forensics and incident response.
Skill Development	Developing skills in evidence collection, data analysis, and reporting, contributing to skill development in the field of cybercrime investigation.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Forensic Tools and Processing of Electronic Evidence**
Local	The knowledge and skills gained in this unit are relevant at the local level as local law enforcement agencies and forensic

	professionals need to be equipped with the tools and techniques to effectively process electronic evidence in cybercrime investigations within their jurisdiction
Regional	-
National	Protecting national security and upholding the rule of law in the digital realm requires a strong capability in digital forensics. The knowledge and proficiency in forensic tools and processing of electronic evidence contribute to national efforts in preventing and investigating cybercrimes.
Global	Cybercrimes are a global concern, and international cooperation is vital in addressing them.
Employability	Proficiency in forensic tools and processing of electronic evidence enhances employability in the field of digital forensics and cybersecurity.
Entrepreneurship	Knowledge of forensic tools and techniques can inspire entrepreneurs to develop innovative solutions, tools, and services in the field of digital forensics.
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4,9,16
NEP 2020	Its aligns with the policy's objective of developing skills relevant to the current and future job market, particularly in the field of cyber security.
POE/4th IR	The Fourth Industrial Revolution by providing knowledge and skills necessary to combat cyber threats and protect digital assets in an increasingly interconnected and digital world

CYBER CRIME INVESTIGATION & DIGITAL FORENSICS LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: **Cyber Crime Investigation & Digital Forensics Lab**	**Course Code**	**L-T-P**	**Credits**
	ENSP353	0-0-2	1
Type of Course:	Minor		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	**Understand** the fundamental concepts and principles of digital forensics and cybercrimes.
CO 2	**Apply** the knowledge of digital forensics techniques and procedures to collect, analyse, and preserve electronic evidence in various types of cybercrimes.
CO 3	**Evaluate** and utilize forensic tools and technologies for data acquisition, analysis, and recovery in the investigation of cybercrimes.
CO 4	**Analyse** and interpret digital evidence obtained from different sources, such as electronic media, internet crimes, malicious applications, and various forms of cybercrimes.

Ex. No	Experiment Title	Mapped CO/COs
1	Experiment on collecting and preserving electronic media for forensic analysis.	CO 2
2	Experiment on searching and retrieving digital evidence from various storage devices.	CO 2
3	Experiment on handling and analyzing malicious codes, such as Trojans, viruses, and logic bombs.	CO 2
4	Experiment on investigating unauthorized access to computer systems.	CO 2
5	Experiment on packet sniffing and analyzing network traffic for evidence.	CO 2
6	Experiment on identifying and investigating phishing attacks and their variants.	CO 2
7	Experiment on detecting and investigating web spoofing and email spoofing incidents.	CO 2
8	Experiment on cyber stalking investigation techniques.	CO 2
9	Experiment on investigating web defacement incidents and identifying the perpetrators.	CO 2
10	Experiment on investigating financial crimes, including ATM and credit card frauds.	CO 2
11	Experiment on tracing and investigating spamming activities.	CO 2
12	Experiment on investigating cases related to software and hardware piracy.	CO 2
13	Experiment on tracing and investigating money laundering activities.	CO 2
14	Experiment on investigating fraud and cheating cases in the digital realm.	CO 2
15	Experiment on analyzing malicious applications and their impact on digital devices.	CO 4

16	Experiment on understanding the role and capabilities of investigation agencies in India.	CO 1
17	Experiment on following proper evidence collection and seizure procedures in digital investigations.	CO 2
18	Experiment on securing and documenting the crime scene in digital forensics.	CO 2
19	Experiment on acquiring and analyzing data from digital devices.	CO 3
20	Experiment on using forensic tools like EnCase and FTK for digital investigations.	CO 3
21	Experiment on countering anti-forensics techniques and retrieving hidden information.	CO 3
22	Experiment on recovering data from damaged SIM cards and other multimedia evidence.	CO 2
23	Experiment on recovering deleted data from desktops, laptops, and mobile devices.	CO 2
24	Experiment on analyzing data from slack space and renamed files.	CO 4
25	Experiment on forensic imaging, including ghosting and analysis of compressed files.	CO 3

Description of experiments:

Session 1:
Topic: Experiment on collecting and preserving electronic media for forensic analysis
• Introduction to electronic media collection and preservation in digital forensics
• Techniques for ensuring the integrity and authenticity of collected data
• Chain of custody and documentation procedures
Exercise: Practice collecting electronic media and preserving it for forensic

analysis.
Project: Create a comprehensive report on the collection and preservation of electronic media: Document the process of collecting electronic media, maintain a chain of custody, and ensure the integrity of the collected data.
Session 2:
Topic: Experiment on searching and retrieving digital evidence from various storage devices
• Different types of storage devices and their characteristics
• Techniques for searching and retrieving digital evidence from storage devices
• File systems analysis and data carving
Exercise: Search for and retrieve digital evidence from different storage devices.
Project: Analyze and document the process of searching and retrieving digital evidence: Perform data recovery and analysis on different storage devices, document the findings, and present a comprehensive report.
Session 3:
Topic: Experiment on handling and analyzing malicious codes, such as Trojans, viruses, and logic bombs
• Introduction to different types of malicious codes
• Techniques for analyzing and understanding malicious code behavior
• Anti-malware tools and techniques
Exercise: Analyze and dissect different types of malicious codes to understand their behavior.
Project: Develop a comprehensive report on the analysis of malicious codes: Analyze and document the behavior of various malicious codes, identify their impact, and propose countermeasures.
Session 4:
Topic: Experiment on investigating unauthorized access to computer systems

• Understanding the concept of unauthorized access
• Techniques for investigating unauthorized access incidents
• Log analysis and intrusion detection systems
Exercise: Investigate and analyze unauthorized access incidents in computer systems.
Project: Create a detailed investigation report on unauthorized access incidents: Analyze log files, identify the extent of unauthorized access, determine the entry points, and propose preventive measures.
Session 5:
Topic: Experiment on packet sniffing and analyzing network traffic for evidence
• Introduction to packet sniffing and network traffic analysis
• Tools and techniques for capturing and analyzing network packets
• Identifying and extracting relevant evidence from network traffic
Exercise: Capture and analyze network packets to extract evidence.
Project: Prepare a comprehensive report on network traffic analysis for a given scenario: Analyze captured network packets, extract relevant evidence, and present the findings in a structured report.
Session 6:
Topic: Experiment on identifying and investigating phishing attacks and their variants
• Understanding phishing attacks and their impact
• Techniques for identifying and investigating phishing incidents
• Analyzing phishing emails and websites
Exercise: Identify and investigate phishing attacks by analyzing phishing emails and websites.
Project: Perform a comprehensive analysis of a phishing attack: Analyze phishing emails and websites, identify the modus operandi, and propose countermeasures to prevent future attacks.

Session 7:
Topic: Experiment on detecting and investigating web spoofing and email spoofing incidents
• Understanding web spoofing and email spoofing techniques
• Techniques for detecting and investigating web and email spoofing incidents
• Analyzing spoofed web pages and email headers
Exercise: Detect and investigate web spoofing and email spoofing incidents by analyzing spoofed web pages and email headers.
Project: Prepare a detailed investigation report on web and email spoofing incidents: Analyze spoofed web pages and email headers, identify the perpetrators, and suggest preventive measures.
Session 8:
Topic: Experiment on cyber stalking investigation techniques
• Understanding cyber stalking and its implications
• Techniques for investigating cyber stalking incidents
• Gathering digital evidence and documenting the case
Exercise: Investigate and gather digital evidence for a cyber stalking case.
Project: Create a comprehensive investigation report on a cyber stalking incident: Analyze the digital evidence, document the case details, and propose measures to protect the victim.
Session 9:
Topic: Experiment on investigating web defacement incidents and identifying the perpetrators
• Understanding web defacement and its impact
• Techniques for investigating web defacement incidents
• Analyzing web defaced pages and server logs
Exercise: Investigate web defacement incidents and analyze defaced web pages

and server logs.
Project: Prepare a detailed investigation report on web defacement incidents: Analyze defaced web pages and server logs, identify the perpetrators, and suggest measures to enhance website security.
Session 10:
Topic: Experiment on investigating financial crimes, including ATM and credit card frauds
• Understanding financial crimes in the digital realm
• Techniques for investigating ATM and credit card frauds
• Analyzing financial transaction records and digital evidence
Exercise: Investigate financial crimes related to ATM and credit card frauds by analyzing financial transaction records and digital evidence.
Project: Create a comprehensive report on the investigation of financial crimes: Analyze financial transaction records, identify fraudulent activities, and propose preventive measures.
Session 11:
Topic: Experiment on tracing and investigating spamming activities
• Understanding spamming activities and their impact
• Techniques for tracing and investigating spamming incidents
• Analyzing spam emails and tracking email senders
Exercise: Trace and investigate spamming activities by analyzing spam emails and tracking email senders.
Project: Prepare a detailed investigation report on spamming activities: Analyze spam emails, trace email senders, identify the source of spamming, and propose measures to mitigate spamming incidents.
Session 12:
Topic: Experiment on investigating cases related to software and hardware piracy
• Understanding software and hardware piracy and its consequences

• Techniques for investigating piracy cases
• Analyzing pirated software and counterfeit hardware
Exercise: Investigate cases related to software and hardware piracy by analyzing pirated software and counterfeit hardware.
Project: Develop a comprehensive report on software and hardware piracy investigations: Analyze pirated software, identify counterfeit hardware, determine the extent of piracy, and propose measures to combat piracy.
Session 13:
Topic: Experiment on tracing and investigating money laundering activities
• Understanding money laundering in the digital realm
• Techniques for tracing and investigating money laundering incidents
• Analyzing financial transaction records and blockchain data
Exercise: Trace and investigate money laundering activities by analyzing financial transaction records and blockchain data.
Project: Prepare a detailed investigation report on money laundering activities: Analyze financial transaction records, track money flow, identify money laundering techniques, and propose measures to prevent money laundering.
Session 14:
Topic: Experiment on investigating fraud and cheating cases in the digital realm
• Understanding fraud and cheating in the digital realm
• Techniques for investigating fraud and cheating cases
• Analyzing digital evidence and transaction records
Exercise: Investigate fraud and cheating cases in the digital realm by analyzing digital evidence and transaction records.
Project: Create a comprehensive investigation report on fraud and cheating cases: Analyze digital evidence, identify fraudulent activities, document the case details, and propose preventive measures.
Session 15:

Topic: Experiment on analyzing malicious applications and their impact on digital devices
• Understanding malicious applications and their impact
• Techniques for analyzing and identifying malicious applications
• Analyzing malware behavior and reverse engineering techniques
Exercise: Analyze and identify malicious applications and study their impact on digital devices.
Project: Prepare a detailed analysis report on malicious applications: Analyze the behavior of different types of malicious applications, identify their impact on digital devices, and propose measures to prevent malware infections.
Session 16:
Topic: Experiment on understanding the role and capabilities of investigation agencies in India
• Introduction to investigation agencies in India
• Understanding the roles and responsibilities of investigation agencies
• Case studies and examples of investigations conducted by Indian agencies
Exercise: Study and understand the roles and capabilities of investigation agencies in India through case studies and examples.
Project: Prepare a report highlighting the role and capabilities of investigation agencies in India: Discuss the functions, powers, and responsibilities of key investigation agencies, and analyze their notable investigations.
Session 17:
Topic: Experiment on following proper evidence collection and seizure procedures in digital investigations
• Understanding the importance of proper evidence collection and seizure
• Techniques and procedures for collecting and preserving digital evidence
• Documentation and chain of custody requirements
Exercise: Practice following proper evidence collection and seizure procedures in

digital investigations.
Project: Create a comprehensive report on evidence collection and seizure procedures: Document the process of evidence collection, maintain the chain of custody, and ensure compliance with legal and procedural requirements.
Session 18:
Topic: Experiment on securing and documenting the crime scene in digital forensics
• Importance of securing the crime scene in digital forensics
• Techniques for securing and documenting the crime scene
• Best practices for maintaining the integrity of digital evidence
Exercise: Secure and document the crime scene in a simulated digital forensics case.
Project: Prepare a detailed report on securing and documenting the crime scene: Describe the steps taken to secure the crime scene, document the process, and provide recommendations for improving crime scene management.
Session 19:
Topic: Experiment on acquiring and analyzing data from digital devices
• Techniques for acquiring data from digital devices
• Best practices for preserving the integrity of acquired data
• Analyzing acquired data using forensic tools and techniques
Exercise: Acquire and analyze data from different digital devices using forensic tools and techniques.
Project: Analyze and document the process of acquiring and analyzing data from digital devices: Perform data acquisition, analyze the acquired data, and present the findings in a structured report.
Session 20:
Topic: Experiment on using forensic tools like EnCase and FTK for digital investigations

• Introduction to popular forensic tools like EnCase and FTK
• Familiarization with the features and capabilities of forensic tools
• Hands-on practice with forensic tool usage in digital investigations
Exercise: Use forensic tools like EnCase and FTK to conduct digital investigations on simulated cases.
Project: Prepare a comprehensive report on the usage of forensic tools in digital investigations: Describe the features and capabilities of EnCase and FTK, document the usage in specific investigations, and evaluate their effectiveness.
Session 21:
Topic: Experiment on countering anti-forensics techniques and retrieving hidden information
• Understanding anti-forensics techniques used to hide digital evidence
• Techniques for countering anti-forensics and retrieving hidden information
• Analysis of steganography, encryption, and file obfuscation methods
Exercise: Counter anti-forensics techniques and retrieve hidden information from digital evidence.
Project: Develop a comprehensive report on countering anti-forensics techniques: Analyze different anti-forensics methods, propose countermeasures, and demonstrate the retrieval of hidden information.
Session 22:
Topic: Experiment on recovering data from damaged SIM cards and other multimedia evidence
• Techniques for recovering data from damaged SIM cards
• Recovering data from damaged multimedia evidence like CCTV footage and audio recordings
• Best practices for data recovery from different types of damaged media
Exercise: Recover data from damaged SIM cards and analyze multimedia evidence from various sources.
Project: Prepare a detailed report on data recovery from damaged media:

Document the process of recovering data from damaged SIM cards and analyze recovered multimedia evidence.
Session 23:
Topic: Experiment on recovering deleted data from desktops, laptops, and mobile devices
• Techniques for recovering deleted data from different devices
• Understanding file systems and data storage mechanisms
• Analyzing recovered deleted data for evidence
Exercise: Recover deleted data from desktops, laptops, and mobile devices and analyze the recovered data for evidence.
Project: Analyze and document the process of recovering deleted data: Recover deleted data from different devices, analyze the recovered data, and present the findings in a comprehensive report.
Session 24:
Topic: Experiment on analyzing data from slack space and renamed files
• Understanding slack space and its significance in digital forensics
• Techniques for analyzing data from slack space and renamed files
• Extracting hidden information and evidence from slack space and renamed files
Exercise: Analyze data from slack space and renamed files to extract hidden information and evidence.
Project: Prepare a detailed report on the analysis of data from slack space and renamed files: Analyze the data, extract hidden information, and present the findings in a structured report.
Session 25:
Topic: Experiment on forensic imaging, including ghosting and analysis of compressed files
• Understanding forensic imaging and its importance in digital forensics

• Techniques for creating forensic images and conducting analysis
• Analyzing ghost images and compressed files for evidence
Exercise: Create forensic images, analyze ghost images, and conduct analysis on compressed files.
Project: Develop a comprehensive report on forensic imaging and analysis: Describe the process of creating forensic images, analyze ghost images, and analyze compressed files for evidence. Present the findings in a structured report.

AI IN CYBER SECURITY

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
AI in Cyber Security	**ENSP305**	4-0-0	4
Type of Course:	Programme Core / Programme Elective /Open Elective		

Pre-requisite(s), if any: basic understanding of web development technologies such as HTML, CSS, and JavaScript. Additionally, students should have some familiarity with networking concepts, operating systems, and databases.

Brief Syllabus:

This syllabus covers essential topics in web application security, including injection attacks, authentication and access control, cryptography, testing, security standards, best practices, and risk management. It is divided into four units and may be completed in a semester-long course. Students will gain an understanding of common web application vulnerabilities and how to prevent and mitigate them. They will also learn about authentication and access control mechanisms, cryptography techniques, and web application security testing. Finally, students will explore best practices for secure web application development and incident response and disaster recovery planning.

UNIT WISE DETAILS

Unit Number: 1	**Title: Introduction to AI and Cyber Security**	**No. of hours: 4**

Content Summary:

Overview of Artificial Intelligence and its applications in Cyber Security History and evolution of AI in cyber security, Understanding of the Cyber Security threats landscape, Familiarization with the latest trends and techniques of AI in Cyber Security, Basic principles of Machine Learning and Deep Learning in Cyber Security, Ethical considerations and challenges of using AI in cyber security.

Unit Number: 2	**Title: Machine Learning Techniques for Cyber Security**	**No. of hours: 8**

Content Summary:

An introduction to Machine Learning techniques, Supervised and unsupervised Machine Learning models in Cyber Security, Feature engineering and data preparation for Machine Learning models, Case studies demonstrating the application of Machine Learning to Cyber Security problems.

Unit Number: 3	Title: Deep Learning Techniques for Cyber	No. of hours: 8

Content Summary:

Introduction to Deep Learning techniques ,Convolutional Neural Networks (CNNs) and their application in Cyber Security ,Recurrent Neural Networks (RNNs) and their application in Cyber Security ,GANs and their application in Cyber Security ,Case studies demonstrating the application of Deep Learning to Cyber Security problems.

Unit Number: 4	Title: AI for Cyber Security: Threat Detection and Prevention	No. of hours: 8

Content Summary:

Introduction to AI and its applications in threat detection and prevention ,Overview of different types of threats in cyber security and their characteristics ,Understanding the limitations of traditional threat detection and prevention methods ,Fundamentals of machine learning and deep learning for threat detection and prevention ,Supervised machine learning algorithms for threat detection, such as decision trees, support vector machines, and random forests ,Unsupervised machine learning algorithms for anomaly detection, such as clustering and outlier detection ,Deep learning techniques for threat detection, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) ,Feature selection and feature engineering for machine learning in threat detection, Emerging trends and challenges in AI for threat detection and prevention, including adversarial machine learning, explainable AI, and privacy concerns.

***Self-Learning Components:**

1) Anomaly Detection
2) Malware Detection
3) Adaptive Access Control
4) Network Traffic Analysis

Reference Books:

1. Artificial Intelligence for Cybersecurity" by Bhaskar Sinha (Auerbach Publications)
2. Machine Learning and Security: Protecting Systems with Data and Algorithms" by Clarence Chio and David Freeman (O'Reilly Media)

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** Understand the concepts and applications of AI in the field of cyber security.
CO2	**Express** the ethical and legal considerations associated with the use of AI in cyber security.
CO3	**Determine** emerging trends and technologies in AI for cyber security, and their potential impact on the field.
CO4	**Identify** strategies for integrating AI-driven solutions into existing cyber security frameworks, policies, and practices.
CO5	**Articulate** critical thinking and problem-solving skills to address real-world cyber security challenges using AI techniques.
CO6	**Design** machine learning techniques for threat detection and prevention in cyber security, including supervised and unsupervised algorithms.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A1	P1
C02	C3	A3	P2
CO3	C3	A3	P3
CO4	C1	A2	-
CO5	C2	A3	P5
CO6	C6	A4	-

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	3	2	2	2	-	-	1	1	1	1
CO2	3	2	3	2	2	1	1	1	1	2	1	2
CO3	3	3	3	3	3	2	2	1	2	1	2	1
CO4	2	2	3	3	3	1	-	-	1	2	1	1
CO5	3	2	3	3	3	2	-	1	2	1	1	1
CO6	3	3	2	3	3	2	1	1	2	1	1	1

Justification for mapping must be relevant

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	2	3	2	1
CO2	3	3	3	2	1
CO3	3	3	3	3	-
CO4	3	3	3	2	-
CO5	3	3	3	3	1
CO6	3	3	3	2	1

Relevance of the Syllabus to various indicators

Unit I	**Introduction to AI and Cyber Security**
Local	Addresses local understanding of the Cyber Security and its impact on society
Regional	Addresses regional Cyber Security infrastructure requirements.
National	Contributes to national Cyber Security literacy and its impact to the nation.
Global	Aligns with global trends in Cyber Security technologies and network protocols
Employability	Develops skills in using Cyber Security and its tools for network protocols
Entrepreneurship	Build entrepreneurship
Skill Development	Develops basic knowledge and skills in Cyber Security technologies and network protocols
Professional Ethics	-

Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Machine Learning Techniques for Cyber Security**
Local	Addresses local understanding of the Internet and its impact on society
Regional	-
National	Contributes to national digital literacy and internet connectivity strategies
Global	Aligns with global trends in internet technologies and network protocols
Employability	Develops skills in using Machine learning techniques and understanding network protocols
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in Machine learning techniques technologies and network protocols
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Deep Learning Techniques for Cyber**
Local	Addresses local network security needs and practices
Regional	-
National	Contributes to national network security strategies and protocols

Global	Aligns with global trends in network security techniques and protocols
Employability	Develops skills in Deep learning techniques and network security techniques
Entrepreneurship	-
Skill Development	Develops knowledge and skills in Deep learning techniques and network security
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**AI for Cyber Security: Threat Detection and Prevention**
Local	Addresses local understanding of Threat Detection and Prevention. and implementation of internet-based services
Regional	-
National	Contributes to national digital communication strategies and multimedia applications
Global	Aligns with global trends in internet telephony, multimedia applications, and SEO
Employability	Develops skills in Threat Detection and Prevention.
Entrepreneurship	-
Skill Development	Develops knowledge and skills in Threat Detection and Prevention
Professional Ethics	-
Gender	-

Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	Aligns with the concepts of internet telephony, multimedia applications, and SEO

AI IN CYBER SECURITY LAB

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
AI in Cyber Security Lab	**ENSP355**	0-0-2	1
Type of Course:	Programme Core / Programme Elective /Open Elective		
Pre-requisite(s), if any: basic understanding of web development technologies such as HTML, CSS, and JavaScript. Additionally, students should have some familiarity with networking concepts, operating systems, and databases.			

Proposed Lab Experiments

Defined Course Outcomes

COs	Comprehensive Understanding of AI in Cyber Security:
CO 1	**Practical** Experience with AI Tools and Techniques
CO 2	**Enhanced** Malware Detection and Classification Skills
CO 3	**Critical** Thinking and Problem-Solving Abilities
CO 4	**Research** and Innovation in AI Cyber Security

Ex. No	Experiment Title	Mapped CO/COs
1	Malware detection: Develop an AI model to detect and classify different types of malwares.	CO 2
2	Intrusion detection: Build an AI system to identify and alert on network intrusions and suspicious activities.	CO 1

3	Phishing detection: Train an AI algorithm to recognize and flag phishing emails or websites.	CO 1, CO 3
4	Vulnerability assessment: Use AI techniques to identify potential vulnerabilities in software or systems.	CO 1, CO 3
5	Botnet detection: Develop an AI model to detect and track botnet activities on a network.	CO 1
6	Password cracking: Build an AI system to analyze and crack weak passwords.	CO 1, CO 3
7	Network traffic analysis: Use AI algorithms to analyze network traffic and identify patterns or anomalies.	CO 1, CO 3
8	Behavioral authentication: Develop an AI model to authenticate users based on their behavioral patterns.	CO 1, CO 3
9	Anomaly detection: Train an AI system to detect anomalous behavior in user activities or system logs.	CO 1
10	Zero-day vulnerability detection: Use AI techniques to identify unknown or previously undiscovered vulnerabilities.	CO 1, CO 3
11	Social engineering detection: Build an AI system to recognize and alert on social engineering attempts.	CO 1, CO 3
12	Web application security: Develop an AI model to identify and mitigate web application vulnerabilities.	CO 3, CO 4
13	Data exfiltration detection: Train an AI algorithm to detect and prevent unauthorized data exfiltration attempts.	CO 3, CO 4
14	Ransomware detection: Use AI techniques to identify and block ransomware attacks in real-time.	CO 3, CO 4
15	Firewall optimization: Employ AI algorithms to optimize firewall rules and configurations for better security.	CO 3, CO 4
16	Network anomaly prediction: Build an AI system to predict network anomalies before they occur.	CO 1, CO 3
17	Security log analysis: Use AI techniques to automatically analyze and correlate security logs for identifying threats.	CO 3, CO 4

18	DDoS attack detection: Develop an AI model to detect and mitigate Distributed Denial of Service (DDoS) attacks.	CO 3, CO 4
19	Mobile application security: Train an AI algorithm to identify security vulnerabilities in mobile applications.	CO 3, CO 4
20	Network segmentation optimization: Employ AI techniques to optimize network segmentation for enhanced security.	CO 1, CO 3
21	Threat intelligence analysis: Use AI algorithms to analyze and extract insights from threat intelligence feeds.	CO 1, CO 3
22	Security incident response automation: Develop an AI system to automate and streamline security incident response processes.	CO 3, CO 4
23	Deepfake detection: Train an AI model to identify and flag manipulated or forged media content.	CO 1, CO 3
24	Network forensics: Use AI techniques to analyze network traffic and digital artifacts for forensic investigations.	CO 3, CO 4
25	Security policy compliance: Develop an AI system to assess and ensure compliance with security policies and regulations.	CO 3, CO 4

SOCIAL MEDIA SECURITY

Department:	**Department of Computer Science and Engineering**		
Course Name: **Social Media Security**	**Course Code**	**L-T-P**	**Credits**
	ENSP307	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			

Brief Syllabus:

Social media has become an integral part of our lives, shaping our online behaviors and interactions in numerous ways. People join social media platforms to share information, connect with friends, and engage in online communities. While social media offers these advantages, it also brings forth concerns regarding privacy and security. The constant flow of personal information shared on these platforms makes individuals vulnerable to various risks. Therefore, it is crucial for all of us to understand and address the issues surrounding privacy and security in the realm of social media. By acquiring knowledge about these challenges, we can adopt safer practices and protect ourselves from potential threats while enjoying the benefits of social media platforms. Being aware and proactive about social media security empowers us to navigate the digital landscape responsibly and ensure our online safety.

UNIT WISE DETAILS

Unit Number: 1	**Title: Social Media Overview**	**No. of hours: 10**

Content Summary:

Introduction to Social media. Types of Social media, Social media platforms, Social media monitoring, Hashtag, Viral content, Social media marketing, challenges, opportunities, and pitfalls in online social networks, APIs, Collecting data from Online Social Media, Social Media Content Analysis - BoW Model, TF-IDF; Network Analysis - Node Centrality Measures, Degree Distribution, Average

Path Length, Clustering Coefficient, Power Law; Synthetic Networks - Random Graphs, Preferential Attachment Model.

Unit Number: 2	Title: Security Issues in Social Media	No. of hours: 10

Content Summary:

Overview, Review of Machine Learning, The evolution of privacy and security concerns with networked technologies, Contextual influences on privacy attitudes and behaviours, Anonymity in a networked world, Identity Theft - Profile Cloning, Social Phishing, Fake, Compromised, Sybil accounts and their behaviour, Spamming, Rumour or Misinformation, Cyberbullying, Collective Misbehaviours, Flagging and reporting of inappropriate content.

Unit Number: 3	Title: Privacy Issues in Social Media	No. of hours: 10

Content Summary:

Overview, Privacy Settings, PII Leakage, Identity vs Attribute Disclosure Attacks, Inference Attacks, De-anonymization Attacks, Privacy Metrics - k-anonymity, l-diversity, Personalization vs Privacy, Differential Privacy, Social Media and User Trust.

Unit Number: 4	Title: Social Media Security: Laws, Best Practices, and Case Studies	No. of hours: 10

Content Summary:

Laws regarding posting of inappropriate content, Best practices for the use of Social media, Content Moderation and Removal Policies, User Authentication and Access Control, Security Awareness and Education, Social media Case studies- Facebook, Twitter, Instagram, YouTube, LinkedIn, StackOverflow, GitHub, Quora, SnapChat, Reddit, FourSquare, Yelp.

***Self-Learning Components:**

1. Social Media Security 101 - Stop The Hackers!
2. Privacy and Security in Online Social Media
3. CompTIA Social Media Security

References:

1. https://www.udemy.com/course/social-media-security-101-stop-the-hackers/

2. https://onlinecourses.nptel.ac.in/noc20_cs31/preview
3. https://niccs.cisa.gov/education-training/catalog/certfirst/comptia-social-media-security

Reference Books:

14. Mastering Social Media Mining, Bonzanini Marco, Packt Publishing Limited
15. Mining the Social Web, Mikhail Klassen and Matthew A. Russell, O'Reilly Media, Inc
16. Social media mining: an introduction, Zafarani, Reza, Mohammad Ali Abbasi, and Huan Liu, Cambridge University Press
17. Social Media Security: Leveraging Social Networking While Mitigating Risk, Michael Cross, Syngress
18. Social Media and the Law: A Guidebook for Communication Students and Professionals, Daxton R. Stewart, Taylor & Francis Ltd
19. Security in the Digital Age: Social Media Security Threats and Vulnerabilities by Henry A. Oliver, Create Space Independent Publishing Platform.

Online References:

1. https://media.defense.gov/2021/Sep/16/2002855950/-1/-1/0/CSI_KEEPING_SAFE_ON_SOCIAL_MEDIA_20210806.PDF
2. https://www.technology.pitt.edu/security/best-practices-safe-social-networking
3. https://www.mdpi.com/1999-5903/10/12/114

Course Outcomes (CO)

COs	Statements
CO1	**Demonstrate an understanding** of the different types of social media platforms, their features, and their impact on communication, marketing, and society.
CO2	**Acquire knowledge and skills** in social media monitoring techniques, including data collection, analysis, and the use of relevant tools and technologies.
CO3	**Develop** the ability to analyze and evaluate viral content on social media, understand the factors contributing to its spread, and recognize its implications for marketing and online engagement.
CO4	**Identify and analyze** the challenges, opportunities, and pitfalls associated with social media marketing, and formulate strategies for effective audience targeting, engagement, and brand promotion.
CO5	**Develop** strategies to safeguard personal information, foster user trust, and mitigate associated risks.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels(C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A3	P1
C02	C2	A2	P2
CO3	C4	A5	-
CO4	C4	A3	P4
CO5	C6	A4	P5

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	3	-	1	-	3	-	-	-	-	-	-
CO2	2	3	-	1	-	-	-	2	-	-	-	-
CO3	-	3	-	2	-	-	1	-	3	-	-	-
CO4	-	3	3	2	-	-	-	2	-	-	-	2
CO5	-	-	-	1	2	2	3	-	-	-	-	2

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	1	-	-	-
CO2	-	2	1	-	-
CO3	-		2	1	-
CO4	-	-	-	2	1
CO5	-	-	-	-	3

Relevance of the Syllabus to various indicators

Unit I	**Social Media Overview**
Local	-
Regional	-
National	Provides essential knowledge and skills related to social media platforms, social media marketing, and data collection from online social media.
Global	Covers key aspects of social media platforms, social media marketing, and data analysis techniques that have global

	applicability
Employability	Highly valued in the job market.
Entrepreneurship	Explore entrepreneurial opportunities in the digital marketing and social media industry.
Skill Development	Enhances students' technical skills in understanding and utilizing social media effectively.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Security Issues in Social Media**
Local	Addresses local indicators by covering security issues in social media.
Regional	-
National	Provides essential knowledge and skills related to security issues in social media, which are relevant at the national level for ensuring online safety.
Global	Covers key aspects of security issues in social media that have global relevance.
Employability	Highly valued in the job market, particularly in roles related to cybersecurity, digital risk management, and social media governance.
Entrepreneurship	Equips with knowledge of security issues in social media, allowing to identify entrepreneurial opportunities in developing innovative solutions for securing social media platforms.
Skill Development	Enhances technical skills in identifying, analyzing, and addressing security threats.
Professional	Indirectly promotes professional ethics by emphasizing the importance of protecting user privacy, preventing

Ethics	cyberbullying, and addressing collective misbehaviors in social media platforms.
Gender	-
Human Values	Indirectly supports human values by fostering a safe and inclusive online environment.
Environment & Sustainability	-
Unit III	**Privacy Issues in Social Media**
Local	Addresses local indicators by covering privacy issues in social media that are relevant to the local context.
Regional	-
National	Provides essential knowledge and skills related to privacy issues in social media, which are relevant at the national level for ensuring data protection and privacy rights.
Global	Covers key aspects of privacy issues in social media that have global relevance, such as identity and attribute disclosure attacks.
Employability	Highly valued in the job market, particularly in roles related to data privacy, information security, and compliance with privacy regulations.
Entrepreneurship	Equips with knowledge of privacy issues in social media, allowing them to identify entrepreneurial opportunities in developing privacy-enhancing solutions for social media platforms.
Skill Development	Enhances technical skills in privacy settings, data protection, and privacy metrics.
Professional Ethics	Indirectly promotes professional ethics by emphasizing the importance of respecting user privacy, protecting personal information, and ensuring transparency.
Gender	-
Human Values	Indirectly supports human values by fostering a culture of

	privacy and user trust in social media platforms. .
Environment & Sustainability	-
Unit IV	**Social Media Security: Laws, Best Practices, and Case Studies**
Local	Addresses local indicators by covering laws regarding posting of inappropriate content that are relevant to local jurisdictions and regulations.
Regional	Provides regional relevance by including case studies of popular social media platforms that are widely used in the regional context, such as Facebook, Twitter, Instagram, and LinkedIn.
National	Covering laws related to social media and best practices for the use of social media platforms
Global	Includes case studies of various global social media platforms.
Employability	Highly valued in roles related to social media management, digital marketing, content moderation, and information security..
Entrepreneurship	Equips with knowledge of social media security laws, best practices, and case studies, allowing them to identify entrepreneurial opportunities in providing social media security services
Skill Development	Enhances students' skills in content moderation, user authentication, access control, security awareness, and education.
Professional Ethics	Emphasizing the importance of adhering to social media laws.
Gender	-
Human Values	Indirectly supports human values by promoting responsible use of social media, ensuring user privacy and safety, and addressing ethical considerations.
Environment &	-

Sustainability	
SDG	-
NEP 2020	Digital literacy, Critical thinking, Ethical use of technology
POE/4th IR	Technological advancements, innovation, adaptability, digital fluency, problem-solving, collaboration, and lifelong learning.

SOCIAL MEDIA SECURITY LAB

Department:	**Department of Computer Science and Engineering**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
Social Media Security Lab	**ENSP357**	0-0-2	1
Type of Course:	Minor		
Pre-requisite(s), if any:			

Course Outcomes (CO)

COs	Statements
CO1	**Understand** the risks and vulnerabilities associated with social media platforms.
CO2	**Understand** the social and ethical implications of social media security.
CO3	**Develop** practical skills to secure social media accounts and data.
CO4	**Analyze** and respond to social media security incidents.
CO5	**Evaluate** the effectiveness of social media security controls.

Proposed Lab Experiments

Ex. No	Experiment Title	Mapped CO/COs
1	Exploring Different Social Media Platforms a. Research and analyze various social media platforms. b. Identify their key features, target audiences, and unique characteristics. c. Compare and contrast their usage, advantages, and challenges.	CO1, CO2, CO5
2	Monitoring Social Media Trends a. Use social media monitoring tools to track popular hashtags and viral content.	CO1, CO5

	b. Analyze the patterns and trends in social media conversations. c. Identify the factors contributing to the popularity of certain content.	
3	Social Media Marketing Analysis a. Study real-world social media marketing campaigns. b. Analyze their strategies, target audience engagement, and impact. c. Evaluate the challenges and opportunities in social media marketing.	CO2, CO3
4	Collecting and Analyzing Social Media Data a. Utilize APIs to collect data from online social media platforms. b. Perform content analysis using techniques like Bag-of-Words (BoW) model and TF-IDF. c. Extract insights and patterns from the collected data.	CO3
5	Social Network Analysis a. Perform network analysis on social media data. b. Calculate node centrality measures, degree distribution, average path length, and clustering coefficient. c. Identify key influencers and community structures within the social network.	CO3, CO5
6	Creating Synthetic Networks a. Generate random graphs and preferential attachment models to simulate social networks. b. Analyze the characteristics of the synthetic networks. c. Compare and contrast them with real-world social networks.	CO1, CO5
7	Profile Cloning and Identity Theft a. Study different types of identity theft in social media. b. Analyze profile cloning, social phishing, and compromised accounts. c. Understand the behavioral patterns and impacts of these attacks.	CO1, CO2
8	Dealing with Spam and Misinformation a. Analyze the spread of spam and misinformation in social media. b. Identify techniques to detect and mitigate spamming activities. c. Evaluate the effectiveness of flagging and reporting mechanisms.	CO4
9	Privacy Settings Evaluation a. Evaluate the privacy settings of popular social media platforms. b. Assess the level of protection they provide for Personally Identifiable Information (PII).	CO2, CO5

	c. Propose recommendations for enhancing user privacy.	
10	Privacy Attacks and Anonymity a. Study different privacy attacks in social media, such as inference attacks and de-anonymization attacks. b. Analyze the impact of identity disclosure and attribute disclosure attacks. c. Explore techniques like differential privacy for preserving user privacy.	CO2
11	Privacy Metrics Analysis a. Investigate privacy metrics like k-anonymity and l-diversity. b. Apply these metrics to analyze the privacy risks in social media datasets. c. Discuss the trade-offs between personalization and privacy in social media.	CO2
12	Understanding Social Media Laws and Regulations a. Study the laws and regulations related to social media usage. b. Analyze the legal implications of posting inappropriate content. c. Explore content moderation policies and user responsibilities.	CO2
13	User Authentication and Access Control a. Evaluate user authentication mechanisms in popular social media platforms. b. Analyze access control policies and user permissions. c. Discuss best practices for ensuring secure user authentication.	CO3
14	Security Awareness and Education a. Develop security awareness campaigns for social media users. b. Design educational materials to raise awareness about social media security risks. c. Evaluate the effectiveness of these campaigns through surveys or assessments.	CO2
15	Case Study Analysis - Facebook a. Analyze the security and privacy practices of Facebook. b. Explore the challenges faced by Facebook in maintaining user data privacy. c. Discuss notable security incidents and their impact on user trust.	CO1, CO2, CO4
16	Case Study Analysis - Twitter a. Investigate the security measures implemented by Twitter. b. Analyze the response to cybersecurity incidents on the	CO1, CO2, CO4

	platform. c. Discuss the role of Twitter in addressing misinformation and cyberbullying.	
17	Case Study Analysis - Instagram a. Analyze the privacy and security features of Instagram. b. Investigate the effectiveness of content moderation policies. c. Discuss the impact of influencer marketing and brand safety on Instagram.	CO1, CO2, CO4
18	Case Study Analysis - YouTube a. Evaluate the security controls and privacy settings of YouTube. b. Analyze the challenges of content moderation and copyright infringement. c. Discuss the role of YouTube in combating hate speech and harmful content.	CO1, CO2, CO4
19	Case Study Analysis - LinkedIn a. Study the security and privacy considerations on LinkedIn. b. Analyze the protection of professional user data and connections. c. Discuss the impact of LinkedIn in job search and professional networking.	CO1, CO2
20	Case Study Analysis - StackOverflow a. Investigate the security practices implemented on StackOverflow. b. Analyze the trust and reputation systems within the community. c. Discuss the role of StackOverflow in knowledge sharing and code collaboration.	CO1, CO2
21	Case Study Analysis - GitHub a. Analyze the security measures adopted by GitHub for source code repositories. b. Investigate the role of vulnerability reporting and code review processes. c. Discuss the importance of secure coding practices in open-source projects.	CO1, CO2
22	Case Study Analysis - Quora a. Evaluate the privacy controls and content moderation on Quora. b. Analyze the impact of user-generated content and knowledge sharing. c. Discuss the challenges of maintaining a respectful and inclusive community.	CO1, CO2
23	Case Study Analysis - SnapChat a. Study the privacy and security features of SnapChat.	CO1, CO2

	b. Analyze the ephemeral messaging and privacy-by-design approach. c. Discuss the challenges of preventing data leaks and unauthorized access.	
24	Case Study Analysis - Reddit a. Analyze the security and privacy considerations on Reddit. b. Investigate the moderation policies and community-driven content curation. c. Discuss the challenges of maintaining a balance between free speech and harmful content.	CO1, CO2

K.R. MANGALAM UNIVERSITY

(DEPARTMENTAL ELECTIVE II)

COMPUTATIONAL SERVICES IN THE CLOUD

Department:	**Department of Computer Science and Engineering**		
Course Name: Computational Services In The Cloud	**Course Code**	**L-T-P**	**Credits**
	ENSP401	4-0-0	4
Type of Course:	Department Elective-II (Minor)		
Pre-requisite(s), if any:			

Brief Syllabus:

This course covers evolutionary computing paradigms from multi-processor systems to Cloud, Edge, and Fog Computing. The course design follows cloud deployment models, service models, virtualization techniques and cloud architectural solutions. It also elaborates cloud compliances and security at fine-grained level by following a shared responsibility model. It reveals a design pattern to the students, enabling them to think through the process of designing and implementing cloud infrastructure and optimal IT solutions

Unit Number: 1	**Title: : Introduction to Cloud Computing**	**No. of hours: 11**

Content Summary:

Cloud Computing, Adoption of cloud-based IT resources, Service Models: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Software-as-a-Service(SaaS), Deployment models: Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud, Cloud Computing Characteristics, Challenges of cloud computing,Virtualization concept, Types of virtualizations, Demo of virtualization, Virtualization Merits, Role of virtualization in cloud computing, Virtualization Demerits, VMPlacement, VM Migration, VM Migration Demo, VM clustering, Design Issues in VM Clustering, Need of Dockers and Containers, Docker Eco-System, Hypervisor vs Docker.

Unit	**Title: Microservices**	**No. of hours: 10**

Number: 2		

Content Summary:

Microservices, Service-Oriented Architecture, REST API, IP Addressing, Subnetting, Supernetting, Designing of Virtual Private Cloud, Demo of VPC, VPC Peering,VPC Case Study, Cloud Storage, Serverless Computing, Cloud API Gateway, Cloud Databases, Resource Provisioning, Time shared and space shared, Efficient VMConsolidation on cloud server, Task/DAG Scheduling Algorithms, Min-Min, Max-Min, MET, B-level Demo, T-level Demo, Task-VM Mapping, Auto Scaling, Load Balancing.

Unit Number: 3	Title: Case Study	No. of hours: 08

Content Summary:

Case Study: Cloud Market analysis, Security and Compliances, Shared security model in IAAS/PAAS/SAAS, Shared technology issues, Data loss or leakage, Accountor service hijacking, Implementation of cloud security, Security Groups, Network Access Control Lists, Cloud databases, Parallel Query Execution with NoSQLDatabase, Big Data, Handling Big Data on Cloud Platform, Map- Reduce framework for large clusters using Hadoop, Design of data applications based on Map Reducein Apache Hadoop.

Unit Number: 4	Title: : Comparative study/analysis of public clouds	No. of hours: 09

Content Summary:

Comparative study/analysis of public clouds, Edge Computing, Fog Computing, Data Offloading, Cloud-Based DevOps Tools, Task Partitioning, Data Partitioning, Data Synchronization, Distributed File System, Data center, Ongoing Research Topics.

Self-Learning Component: - The students are expected to choose a topic in discussion with the industry expert and implement the concepts of cloud computing. The student should present the progress at the end of the semester.

Reference Books:

1. Lizhe Wang, Rajiv Ranjan, Jinjun Chen and Boualem Benatallah, Cloud Computing (1 ed.), CRC Press, 2017. ISBN 978-1351833097.
2. Judith S. Hurwitz and Daniel Kirsch, Cloud Computing For Dummies (2 ed.), Hoboken: John Wiley & Sons, 2020. ISBN 978-1119546658.

3. Prerna Sharma, Moolchand Sharma and Mohamed Elhoseny, Applications of Cloud Computing (1 ed.), CRC Press, 2020. ISBN 9780367904128.

Define Course Outcomes (CO)

COs	
CO1	Explain the core concepts of the cloud computing paradigm: how and why this paradigm shift came about, the characteristics, advantages and challenges brought about by the various models and services in cloud computing.
CO2	Apply the fundamental concepts in datacenters to understand the tradeoffs in power, efficiency and cost.
CO3	Identify resource management fundamentals, i.e. resource abstraction, sharing and sandboxing and outline their role in managing infrastructure in cloud computing.
CO4	Analyze various cloud programming models and apply them to solve problems on the cloud.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels (c) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C3	A3	P4
CO2	C3	A4	P2

CO3	C2	A2	P1
CO4	C4	A5	P5

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	1	1				2	1	2	2	1
CO2	2	2	2	1			1	3		1	2	2
CO3	3	3	3		1	2		2	1		1	1
CO4	3	3	2	1		2		2	2		2	1

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	2	1	3	-
CO2	3	3	2	3	-
CO3	1	1	3	1	-
CO4	1	1	1	2	1

Relevance of the Syllabus to various indicators

Unit I	**Introduction to Design thinking**
Local	-
Regional	-
National	-
Global	It promotes international collaboration, data sharing, and

	connectivity, facilitating global trade, research, and innovation.
Employability	Proficiency in cloud computing technologies, service models, and deployment models enhances employability and career prospects.
Entrepreneurship	-
Skill Development	This includes skills in virtualization, cloud service models, deployment models, and containerization.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Case studies in Design thinking**
Local	-
Regional	-
National	-
Global	In the global context, microservices, service-oriented architecture, and cloud computing are essential technologies driving digital innovation and transformation. Global organizations heavily rely on these technologies to deliver scalable and resilient applications and services to a global user base.
Employability	Acquiring skills in microservices, service-oriented architecture, and cloud computing enhances employability prospects in the IT industry. Proficiency in these technologies is in high demand as organizations increasingly adopt cloud-based architectures and microservice-oriented approaches
Entrepreneurship	-

Skill Development	The course on microservices, service-oriented architecture, and cloud computing promotes skill development in areas such as cloud infrastructure design, API development, virtualization, storage management, and resource provisioning
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Design Frameworks**
Local	-
Regional	-
National	-
Global	It helps address global challenges such as data privacy, security breaches, and international data transfer regulations.
Employability	Knowledge of cloud market analysis, security, and compliance enhances employability in various roles, including cloud architects, cloud security specialists, and cloud consultants.
Entrepreneurship	-
Skill Development	Studying cloud market analysis, security, and compliance develops critical skills such as risk assessment, security implementation, and compliance management.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-

Unit IV	**Innovation & Creativity**
Local	-
Regional	-
National	-
Global	Cloud computing is a global phenomenon, and a course on comparative study/analysis of public clouds and related topics provides students with a global perspective on cloud technologies.
Employability	Proficiency in cloud computing technologies and understanding the different deployment models, such as public clouds and edge computing, enhances employability prospects in the IT industry.
Entrepreneurship	-
Skill Development	A course on comparative study/analysis of public clouds and related topics enhances various skills such as research, critical analysis, problem-solving, and technical proficiency.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4, 9
NEP 2020	Integration of Emerging Technologies and Skill Development
POE/4th IR	The course on cloud computing aligns with the Fourth Industrial Revolution (IR 4.0) in the following ways: Advanced Infrastructure, Data Management and Analytics and Digital Transformation.

COMPUTATIONAL SERVICES IN THE CLOUD LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: Computational Services In The Cloud Lab	**Course Code**	**L-T-P**	**Credits**
	ENSP451	0-0-2	1
Type of Course:	Minor		
Pre-requisite(s), if any:			

Proposed lab experiments

Defined Course Outcomes

COs	
CO 1	Apply the concept cloud computing to solve practical use cases.
CO 2	Analyzing different services in cloud computing
CO 3	Evaluate different available services provided by cloud vendors
CO 4	Design Cloud based application

Ex No	Experiment Title	Mapped CO/COs
1	Install Virtualbox / VMware Workstation with different flavours of linux or windows OS	CO1
2	Install a C compiler in the virtual machine created using virtual box and execute Simple Programs	CO1
3	Install Google App Engine. Create hello world app and other simple web applications using python/java.	CO1

4	Use GAE launcher to launch the web applications.	CO1
5	Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.	CO2
6	Find a procedure to transfer the files from one virtual machine to another virtual machine	CO2
7	Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)	CO2
8	Install Hadoop single node cluster and run simple applications like wordcount.	CO1
9	Deploy a simple web application using a Platform as a Service (PaaS) offering like AWS Elastic Beanstalk, Azure App Service, or Google App Engine	CO4
10	Create and manage databases using services like AWS RDS, Azure SQL Database, or Google Cloud SQL.	CO3
11	Containerize an application using Docker and create a container registry on a cloud platform.	CO3
12	Deploy and manage containers using container orchestration tools like AWS ECS, Azure Kubernetes Service (AKS), or Google Kubernetes Engine (GKE)	CO4
13	Develop and deploy a serverless function using AWS Lambda, Azure Functions, or Google Cloud Functions	CO4
14	Configure event triggers, access permissions, and monitoring for serverless functions	CO2
15	Utilize cloud-based data analytics tools like AWS Athena, Azure Data Lake Analytics, or Google BigQuery to query and analyze large datasets	CO2
16	Build and train machine learning models using cloud-based services like AWS SageMaker, Azure Machine Learning, or Google Cloud AutoML	CO2
17	Use infrastructure provisioning tools such as AWS CloudFormation, Azure Resource Manager, or Google Cloud Deployment Manager to define and deploy infrastructure components	CO1

18	Set up cloud monitoring and logging services like AWS CloudWatch, Azure Monitor, or Google Cloud Monitoring to track the performance and health of cloud resources	CO3
19	Implement access control policies and roles using AWS IAM, Azure Active Directory, or Google Cloud Identity and Access Management (IAM)	CO4
20	Enable encryption for data at rest and in transit using cloud security services	CO3
21	Explore cost optimization techniques like auto-scaling, spot instances, or reserved instances to optimize cloud resource usage and reduce costs	CO2

MICROSOFT AZURE CLOUD FUNDAMENTALS

Department:	**Department of Computer Science and Engineering**		
Course Name: Microsoft Azure Cloud Fundamentals	**Course Code**	**L-T-P**	**Credits**
	ENSP403	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			
Brief Syllabus: The Microsoft Azure Cloud Fundamentals course introduces the concepts of cloud computing and the Azure platform. It covers Azure services such as Virtual Machines, Storage, Networking, Identity, App Services, and Databases. The course focuses on security, monitoring, and management in Azure. Real-world case studies and hands-on labs enable practical application. The subject equips students to design trustworthy intrusion detection systems and enhances security in IoT networks using Azure.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction to Cloud Computing**	**No. of hours: 10**	
Content Summary: Introduction to Cloud Computing and its Characteristics, Benefit, and Challenges of cloud computing. Cloud service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Cloud Deployment Models, for example, Public, private, hybrid, and community clouds. shared responsibility model. Identify appropriate use cases for each cloud model. Consumption-based model. Comparison of cloud pricing models.			
Unit Number: 2	**Title: Introduction to Microsoft Azure**	**No. of hours: 10**	
Content Summary: Microsoft Azure cloud platform and its significance in the industry, Azure regions, and availability zones for understanding global data center distribution. Introduction of various Azure services and solutions available for			

different scenarios.		
Unit Number: 3	**Title: Azure Virtual Machines (VMs) and Storage**	**No. of hours: 10**
Content summary: Create and manage virtual machines using Azure. Different VM sizes and types based on performance requirements. VM scaling and load balancing for optimizing application performance. Azure storage services: Blob Storage, Table Storage, File Storage, and Disk Storage.		
Unit Number: 4	**Title: Azure Networking, Identity and Access Management**	**No. of hours: 10**
Content Summary: creation and configuration of virtual networks and subnets in Azure. Azure Load Balancer for distributing incoming network traffic, VPN Azure Active Directory (Azure AD) for managing identities and authentication. Gateway for secure communication between on-premises networks, and Azure. Azure Active Directory (Azure AD) for managing identities and authentication. Database Services, Azure Storage Account for data storage and retrieval.		
Reference Books: 1. Microsoft Azure Essentials Fundamentals of Azure by Michael S. Collier and Robin E. Shahan 2. Mastering Microsoft Azure Infrastructure Services by John Savill 3. Azure for Architects by Ritesh Modi **Text Books:** **1.** Microsoft Azure For Dummie by Timothy L. Warner and Katrina Dow, and Michael Washam		

Define Course Outcomes (CO)

COs	
CO1	**Understand** Microsoft Azure concepts
CO2	**Express** proficiency in the handling of Azure services

CO3	**Determine** methods to create and manipulate virtual machines
CO4	**Identify** commonly used models to implement cloud network
CO5	**Articulate** Azure database services

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels (C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2		P1
C02	C3		P2
CO3	C3		P4
CO4	C1		-
CO5	C1		P4

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	–	–	–	–	1	1	1	–	–	–	1
CO2	3	3	3	–	3	–	–	–	–	2	–	–
CO3	3	3	2	3	3	–	–	–	–	–	–	2

CO4	2	2	2	2	2	–	–	–	–	–	–	–
CO5	2	2	–	2	2	–	–	–	–	2	–	–

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	–	–	–	1
CO2	3	3	3	–	1
CO3	3	3	–	–	-
CO4	2	2	–	2	-
CO5	2	2	2	2	1

Relevance of the Syllabus to various indicators

Unit I	**Introduction to Cloud Computing**
Local	
Regional	
National	
Global	Develops skills in using cloud-based services
Employability	Develops skills in using cloud-based services
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills regarding Cloud concepts as well as in cloud computing, network management, and Azure services
Professional Ethics	-
Gender	-

Human Values	-
Environment & Sustainability	-
Unit II	**Introduction to Microsoft Azure**
Local	Understanding of Microsoft Azure services
Regional	-
National	-
Global	Aligns with global trends of Cloud
Employability	Develops cloud-based Knowledge for cloud architects
Entrepreneurship	-
Skill Development	Develops conceptual knowledge of Cloud computing
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Azure Virtual Machines (VMs) and Storage**
Local	
Regional	-
National	-
Global	Designing Virtual machines in Azure and storage management
Employability	Develops skills regarding the cloud security and cloud architecture
Entrepreneurship	-
Skill Development	-

Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Azure Networking, Identity, and Access Management**
Local	
Regional	-
National	-
Global	Access management of data
Employability	Develops skills in cloud security and network management
Entrepreneurship	-
Skill Development	Develops knowledge and skills in data storage and access management
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	Aligns with the concepts of technology and coding

MICROSOFT AZURE CLOUD FUNDAMENTALS LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: Microsoft Azure Cloud Fundamentals Lab	**Course Code**	**L-T-P**	**Credits**
	ENSP403	0-0-2	1
Type of Course:	Departmental Elective-II (Minor)		
Pre-requisite(s), if any:			

Proposed lab experiments

Defined Course Outcomes

COs	
CO 1	Acquire a practical understanding of cloud computing through hands-on lab experiments.
CO 2	Develop proficiency in using global data centers.
CO 3	Demonstrate the ability to create virtual machines and handle data storage
CO 4	Analyze and maintain the security and access management system.

Ex. No	Experiment Title	Mapped CO/COs
1	Familiarize students with the lab environment, software, and tools.	CO1
2	Creating and Managing Virtual Machines with Virtual Box	CO1
3	Create and configure virtual machines running different operating systems (e.g., Windows and Linux).	CO2

4	Install and configure Next Cloud on a local server or cloud-based virtual machine.	CO2
5	Set up your (user) accounts and storage quotas.	CO2
6	Try to implement -Test file uploads, sharing, and collaboration features	CO1
7	Install pfSense as a virtual router/firewall.	CO2
8	Configure virtual networks, subnets, and VLANs	CO2
9	Network Setup with pf Sense: for Test routing, port forwarding, and firewall rules.	CO1
10	Identity and Access Management with Key Cloak: Explore user authentication methods, roles, and permissions.	CO3
11	Identity and Access Management with Key Cloak Configure single sign-on (SSO) for different applications.	CO3
12	Install Key Cloak as an identity provider on a virtual machine.	CO3
13	Install Azure CLI and PowerShell on your machines.	CO3
14	Configure single sign-on (SSO) for different applications.	CO3
15	Use Azure CLI and PowerShell to create and manage Azure resources (e.g., VMs, storage accounts).	CO3
16	Create a simple web app (e.g., using HTML/CSS/JS or a web framework).	CO3
17	Deploy the web app to Azure App Service using Azure portal or Azure CLI.	CO3
18	Test the app's accessibility and scalability	CO4
19	Create an Azure SQL Database instance.	CO4
20	Monitor security alerts and take remedial actions.	CO4

STORAGES AND DATABASES ON CLOUD

Department:	**Department of Computer Science and Engineering**		
Course Name: **Storages and Databases on Cloud**	**Course Code**	**L-T-P**	**Credits**
	ENSP405	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			
Brief Syllabus: The course on cloud databases and storage provides a comprehensive understanding of the principles, technologies, and best practices associated with storing and managing data in the cloud. The syllabus covers various topics, starting with an introduction to cloud computing and an exploration of different types of cloud storage and databases, including object storage, block storage, file storage, relational databases, NoSQL databases, and more. Students delve into popular cloud storage and database services, such as Amazon S3, Google Cloud Storage, and Azure Blob Storage gaining practical knowledge of their features, deployment options, scalability, and high availability. The curriculum also includes essential aspects like database design, data migration, security measures, backup and recovery strategies, performance optimization, and monitoring techniques. Real-world case studies provide insights into organizations utilizing cloud storage and databases effectively.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction to Storage on cloud**	**No. of hours: 4**	
Content Summary: Introduction to Cloud Computing, Overview of cloud databases and cloud storages, types of cloud storages(Object, block and file), different types of cloud database management systems, Gartner Magic Quadrant for Cloud Database Management Systems, Advantages of Working with Cloud Databases, Considerations for Cloud Databases, Top Cloud Database, Factors that help in choosing the right cloud database, Challenges involved in using cloud storages			

and databases.		
Unit Number: 2	**Title: Data Integration, Migration, Security and performance on cloud**	**No. of hours: 8**
Content Summary: Techniques, tool, methods and considerations for migrating from premise database to cloud databases Backup, Recovery, and Disaster Planning including automated backups, point-in-time recovery and replication Performance Optimization and Monitoring including query optimization, indexing, caching, and monitoring tools Scalability and High Availability: load balancing, replication, sharding, and auto-scaling, Cloud Data Warehousing		
Unit Number: 3	**Title: AWS Cloud Storage**	**No. of hours: 8**
Content Summary: Introduction to AWS cloud storage, AWS management console, AWS Storage Services, Uploading files and images , Creating a web server, Overview of Amazon S3, Storage Classes, EC2 Instance Storage, network file system Amazon Elastic Block Store, Amazon Elastic file system, Amazon Cloud Front. Brief introduction to Google Cloud Storage, and Azure Blob Storage.		
Unit Number: 4	**Title: Case Study**	**No. of hours: 8**
Content Summary: Case Studies and Real-world Examples of Netflix , Airbnb, Pinterest, spotify, coca-cola etc. Analyzing real-world use cases of organizations using cloud storage and databases, discussing architecture decisions, challenges, and lessons learned.		

***Self-Learning Components:**

1) **E**xplore open-source projects related to cloud databases and storage. discover open-source projects from GitHub, learn from the codebase, and contribute to the development of cloud-based storage and database solutions. (https://github.com/topics/cloud-database)

2) Join the course : https://www.coursera.org/specializations/cloud-computing

2) Join Online Courses like

- https://www.udemy.com/topic/cloud-computing/
- https://www.coursera.org/courses?query=data%20storage
- https://www.mygreatlearning.com/academy/learn-for-free/courses/databases-and-files-systems-in-aws
- https://www.youtube.com/watch?v=EN4fEbcFZ_E
- https://www.codecademy.com/catalog/subject/cloud-computing
- https://www.simplilearn.com/tutorials/cloud-computing-tutorial

2) Read case studies of organizations that have successfully implemented cloud storage and database solutions.

- **Case Study of Delloite:** https://www2.deloitte.com/us/en/pages/consulting/articles/cloud-computing-case-studies.html
- **Case Study of Amazon:** https://aws.amazon.com/solutions/case-studies/amazon/

Learn about their challenges, architectural decisions, and best practices they adopted.

Reference Books:

1) "Database Cloud Storage: The Essential Guide to Oracle Automatic Storage Management" by Nitin Vengurlekar, 2013

2) "Cloud Database Development and Management" by Lee chao, 2013

3) "Advancing Cloud Database Systems and Capacity Planning with Dynamic Applications" by Narendra Kumar Kamila, 2017

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** & Recall the fundamental concepts and principles of cloud storage and databases. Interpret the security considerations and best practices for cloud storage and databases.
CO2	**Express** ideas and solutions for optimizing performance and improving efficiency in cloud storage and databases through indexing, caching, and query optimization techniques.
CO3	**Determine** the requirements of an application or system and determine the most suitable cloud storage and database solutions to meet those requirements.
CO4	**Identify** different types of cloud storage and database services available.
CO5	**Articulate** the best practices and considerations for designing scalable, reliable, and secure cloud storage and database architectures.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels (C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2		P1
C02	C3		P2
CO3	C3		P3

CO4	C2		-
CO5	C3		P1

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	2	1	-	1	-	1	1	2	2	-	3
CO2	2	3	1	1	1	1	1	1	2	2	1	3
CO3	2	2	3	2	2	3	2	2	3	3	3	3
CO4	1	2	-	3	1	1	-	-	2	2	1	2
CO5	-	2	1	1	3	2	-	-	2	3	2	3

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	2	3	1	3	1
CO2	3	2	2	2	1
CO3	3	2	3	3	-
CO4	1	1	2	1	-
CO5	2	3	2	2	-

Relevance of the Syllabus to various indicators

Unit I	**Introduction to cloud databases and storages**
Local	local considerations may include factors like local network connectivity, data centre locations, and latency between local systems and cloud databases.
Regional	regional considerations can include regulatory compliance specific to a particular region, data sovereignty issues, and availability zones
National	national considerations such as national data protection laws and regulations, government policies related to cloud computing, and national cybersecurity guidelines that may impact the usage and storage of data in cloud databases.
Global	Aligns with global trends in internet technologies and network protocols accessibility of cloud databases and storage options across multiple regions, data replication and backup strategies for disaster recovery on a global scale
Employability	The unit syllabus may indirectly contribute to employability by introducing students to cloud computing concepts, cloud databases, and storage technologies
Entrepreneurship	-
Skill Development	The unit syllabus primarily focuses on developing skills related to cloud computing, cloud databases, and storage
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Data Integration, Migration, Security and performance on cloud**

Local	local considerations may include factors like local network connectivity, data centre locations, and latency between local systems and cloud databases.
Regional	regional considerations can include regulatory compliance specific to a particular region, data sovereignty issues, and availability zones
National	national considerations such as national data protection laws and regulations, government policies related to cloud computing, and national cybersecurity guidelines that may impact the usage and storage of data in cloud databases.
Global	Aligns with global trends in internet technologies and network protocols accessibility of cloud databases and storage options across multiple regions, data replication and backup strategies for disaster recovery on a global scale
Employability	local considerations may include factors like local network connectivity, data centre locations, and latency between local systems and cloud databases.
Entrepreneurship	regional considerations can include regulatory compliance specific to a particular region, data sovereignty issues, and availability zones
Skill Development	national considerations such as national data protection laws and regulations, government policies related to cloud computing, and national cybersecurity guidelines that may impact the usage and storage of data in cloud databases.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**AWS Cloud Storage**

Local	Primarily focuses on cloud storage technologies and their local implementation within the AWS ecosystem.
Regional	-
National	Contributes to national network security strategies and protocols
Global	AWS, Google Cloud, and Azure are global cloud service providers, enabling businesses worldwide to access their storage services and infrastructure.
Employability	Acquiring knowledge and skills in these cloud storage platforms enhances employability in the field of cloud computing and storage management.
Entrepreneurship	Knowledge of AWS, Google Cloud, and Azure storage services can empower entrepreneurs to build and scale their businesses by utilizing scalable and reliable cloud storage solutions for their applications and data.
Skill Development	The unit syllabus focuses on developing skills related to AWS cloud storage, including uploading files and images, creating a web server, utilizing different storage services
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Case Studies**
Local	Addresses local understanding and implementation of internet-based services
Regional	-
National	Contributes to national digital communication strategies and multimedia applications
Global	Aligns with global trends in internet telephony,

	multimedia applications, and SEO
Employability	Develops skills in internet telephony, multimedia applications, and SEO
Entrepreneurship	-
Skill Development	Develops knowledge and skills in internet telephony, multimedia applications, and SEO
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	It supports NEP's objective of enhancing the use of technology in education and research by providing students with knowledge and skills in cloud databases and storage, which are essential components of modern digital infrastructure
POE/4th IR	The course can incorporate discussions on ethical considerations in cloud databases and storage, addressing issues such as data privacy, security, and responsible data management.

STORAGES AND DATABASES ON CLOUD LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: Storages and Databases on Cloud Lab	**Course Code**	**L-T-P**	**Credits**
	ENSP455	0-0-2	1
Type of Course:	Departmental Elective-II (Minor)		
Pre-requisite(s), if any:			

Proposed lab experiments

Defined Course Outcomes

COs	.
CO 1	Gain hands-on experience in using cloud storage services for data storage and retrieval.
CO 2	Implement backup and recovery strategies for cloud-based storage and databases.
CO 3	Optimize database performance in the cloud through indexing and query optimization techniques.
CO 4	Analyse real-world case studies to understand the challenges and solutions related to cloud storage and databases.

Ex. No	Experiment Title	Mapped CO/COs
1	**Create accounts and configure cloud storage services such as Amazon S3, Google Cloud Storage, or Azure**	CO1, CO4,

	Blob Storage: Practice creating buckets/containers, uploading files, setting access permissions, and managing storage resources.	CO5
2	**Perform operations on object storage, including uploading, downloading, and deleting files:** Explore advanced features like versioning, lifecycle policies, and metadata management.	CO1, CO2, CO5
3	**Perform Block Storage Configuration:** Create and attach storage volumes to virtual machines, perform formatting and mounting, and understand snapshotting and resizing operations.	CO1, CO3, CO5
4	**File Storage Implementation:** Work with shared file systems such as Amazon EFS, Google Cloud Filestore, or Azure Files. They configure file shares, mount them on virtual machines, and explore features like file locking and access control.	CO1, CO4, CO5
5	**Database Provisioning and Management:** • learn to set up and configure cloud databases like Amazon RDS, Google Cloud SQL, or Azure Database Services • create database instances, manage security settings, and perform basic administration tasks such as backups and restores.	CO1, CO3, CO4, CO5
6	**Data Migration to the Cloud:** • Practice migrating databases from on-premises or other cloud providers to the selected cloud database service. • Explore different migration methods, tools, and validate the successful transfer of data	CO1, CO3, CO4, CO5
7	**Database Design and Schema Management:** • Work on designing and creating database schemas for various scenarios. • Implement tables, relationships, indexes, and constraints to support specific use cases and understand database modelling concepts.	CO1, CO3, CO5, CO6
8	**Data Security and Access Control:** Configure security measures for cloud databases, including	CO1, CO5

	setting up user accounts, managing roles and permissions, and implementing encryption techniques to protect data at rest and in transit	
9	**Performance Tuning and Optimization:** Learn performance tuning techniques for cloud databases, including query optimization, index creation, caching strategies, and monitoring tools to identify and resolve performance bottlenecks	CO1, CO2, CO5
10	**High Availability and Scalability:** • Explore features like replication, automatic scaling, and load balancing to ensure high availability and scalability of databases in the cloud • Configure and test failover scenarios and evaluate the performance impact of scaling operations.	CO1, CO3, CO5
11	**Project** **Backup and Recovery Strategies:** • Practice implementing backup and recovery mechanisms for cloud databases • perform regular backups, schedule automated backups, and simulate recovery scenarios to restore databases to a consistent state.	CO1, CO5
12	**Project** **Cloud Data Warehousing:** • Gain hands-on experience with cloud data warehousing platforms like Amazon Redshift • load data, run complex queries, and analyze performance optimization techniques for data warehousing.	CO1, CO2, CO5, CO6
13	**Project** Design and implement an inventory management system that utilizes cloud storage for storing inventory data. Include features such as real-time inventory tracking, order management, and reporting capabilities.	CO1, CO2, CO6
14	**Project** Build a document management system that leverages cloud storage for storing and organizing documents. Include features like full-text search, document tagging, and access	CO1, CO2, CO5s

	control for secure document sharing.	

APPLICATION DEVELOPMENT AND DEVOPS ON CLOUD

Department:	**Department of Computer Science and Engineering**		
Course Name: **Application Development And Devops On Cloud**	**Course Code**	**L-T-P**	**Credits**
	ENSP407	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any: Nil			
Brief Syllabus: The syllabus aims to equip students with practical skills and theoretical knowledge to design, develop, and deploy applications in cloud environments while implementing DevOps practices to enhance software development, delivery, and operations on the cloud. It prepares them for a career in the dynamic and rapidly growing field of cloud computing and DevOps, where demand for skilled professionals is high due to the increasing adoption of cloud technologies in various industries.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction to Cloud Computing**	**No. of hours: 10**	
Content Summary: Overview of Cloud Computing: Definition, Characteristics, Advantages, and Disadvantages. Cloud Service Models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS). Cloud Deployment Models: Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud. Cloud Providers: Introduction to popular cloud service providers (e.g., AWS, Azure, Google Cloud). Cloud Architecture: Understanding cloud components, virtualization, and multi-			

tenancy.

Introduction to DevOps: Principles, Practices, and Benefits.

Unit Number: 2	Title: Cloud-Based Application Development	No. of hours: 10

Content Summary:

Cloud Native Applications: Principles and architecture for building applications optimized for cloud environments.

Containerization: Docker and Kubernetes for application deployment, scaling, and management.

Microservices Architecture: Designing, developing, and deploying microservices-based applications.

Serverless Computing: Introduction to serverless architecture and Function as a Service (FaaS).

Application Security in the Cloud: Best practices for securing cloud-based applications.

Cloud-Based Databases: Database options and considerations in cloud environments.

Unit Number: 3	Title: DevOps Practices in Cloud	No. of hours: 10

Content Summary:

Continuous Integration (CI) and Continuous Deployment (CD) in Cloud: Setting up CI/CD pipelines.

Infrastructure as Code (IaC): Automating infrastructure provisioning using tools like Terraform and CloudFormation.

Configuration Management: Managing application configurations in a cloud-based environment.

Monitoring and Logging in Cloud: Tools and techniques for monitoring application performance and gathering logs.

Auto-scaling and Load Balancing: Scaling applications dynamically based on demand.

High Availability and Disaster Recovery: Designing and implementing resilient

applications in the cloud.

Unit Number: 4	Title: Cloud-Based DevOps Tools and Best Practices	No. of hours: 10

Content Summary:

Cloud-Based Version Control: Using Git and other version control tools in cloud-based development. Collaboration and Communication Tools: Utilizing cloud-based collaboration tools for distributed teams. Cloud-Based Testing and Quality Assurance: Strategies for testing applications in cloud environments. Cost Management in Cloud: Optimizing cloud resource usage and cost control. Performance Optimization in Cloud: Techniques for improving application performance in cloud environments. Case Studies and Real-World Projects: Analyzing successful cloud-based application development and DevOps projects.

***SELF-LEARNING COMPONENTS:**

- https://elearn.nptel.ac.in/shop/iit-workshops/completed/azure-devops-and-micro-services-azure-kubernetes-deployment-models/
- https://nptel.ac.in/courses/106105167

Please Note:

1)Students are supposed to learn the components on self-basis

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components.

Reference Books:

1. Jez Humble and David Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Pearson Education, Inc., 2011.
2. Thomas Erl, Ricardo Puttini, and Zaigham Mahmood, Cloud Computing: Concepts, Technology & Architecture, Prentice Hall, 2013.
3. Arun Eapen, Docker on Amazon Web Services: Build, deploy, and manage your container applications at scale on AWS, Packt Publishing, 2017.
4. Sam Newman, Building Microservices: Designing Fine-Grained Systems, O'Reilly Media, Inc., 2015.
5. Mark Richards and Neal Ford, Fundamentals of Software Architecture: An Engineering Approach, O'Reilly Media, Inc., 2020.

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** the fundamental concepts and principles of cloud computing, including different cloud service models and deployment models. Gain insights into various cloud providers and their offerings to make informed decisions for application development and deployment.
CO2	**Acquire** practical knowledge and hands-on experience in developing cloud-native applications, utilizing containerization with Docker and orchestration using Kubernetes. Demonstrate proficiency in designing and implementing microservices-based architectures.
CO3	**Master** the principles and practices of DevOps in a cloud environment. Learn how to set up Continuous Integration (CI) and Continuous Deployment (CD) pipelines, automate infrastructure provisioning with Infrastructure as Code (IaC), and manage application configurations effectively.
CO4	**Develop** the skills to monitor, log, and optimize the performance of cloud-based applications. Explore auto-scaling and load balancing techniques to ensure high availability and disaster recovery strategies for resilient applications.
CO5	**Apply** cloud-based DevOps tools and best practices to collaborate effectively in distributed teams. Gain knowledge of cloud-based version control, testing, cost management, and performance optimization techniques to deliver efficient and cost-effective cloud solutions.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels (C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	-	P1
CO2	C3	-	P2
CO3	C3	-	P3
CO4	C1	-	-
CO5	C1	-	P1

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	3	3	2	-	2	1	-	3	3	-	1	3
CO2	3	3	2	-	3	-	-	-	-	-	-	3
CO3	3	3	2	2	3	1	-	-	-	-	-	3
CO4	-	3	2	2	3	-	-	-	-	-	1	3
CO5	-	3	2	3	2	-	-	-	-	-	-	3

Justification for mapping must be relevant.

1=weakly mapped
2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4	PSO5
CO1	3	1	2	-	-
CO2	2	1	-	-	-
CO3	2	-	-	1	1
CO4	-	-	1	-	-
CO5	2	1	-	1	1

Relevance of the Syllabus to various indicators

Unit I	**Introduction to Cloud Computing**
Local	Addresses local understanding of the Internet and its impact on society
Regional	Addresses regional internet connectivity and network infrastructure requirements
National	Contributes to national digital literacy and internet connectivity strategies
Global	Aligns with global trends in internet technologies and network protocols
Employability	Develops skills in using internet-based services and understanding network protocols
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in internet technologies and network protocols
Professional Ethics	-
Gender	-

Human Values	-
Environment & Sustainability	-
Unit II	**Cloud-Based Application Development**
Local	Addresses local understanding of the Internet and its impact on society
Regional	-
National	Contributes to national digital literacy and internet connectivity strategies
Global	Aligns with global trends in internet technologies and network protocols
Employability	Develops skills in using internet-based services and understanding network protocols
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in internet technologies and network protocols
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**DevOps Practices in Cloud**
Local	Addresses local network security needs and practices
Regional	-
National	Contributes to national network security strategies and protocols
Global	Aligns with global trends in network security techniques and

	protocols
Employability	Develops skills in network programming and network security techniques
Entrepreneurship	-
Skill Development	Develops knowledge and skills in client-server programming and network security
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Cloud-Based DevOps Tools and Best Practices**
Local	Addresses local understanding and implementation of internet-based services
Regional	-
National	Contributes to national digital communication strategies and multimedia applications
Global	Aligns with global trends in internet telephony, multimedia applications, and SEO
Employability	Develops skills in internet telephony, multimedia applications, and SEO
Entrepreneurship	-
Skill Development	Develops knowledge and skills in internet telephony, multimedia applications, and SEO
Professional Ethics	-
Gender	-

Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	Aligns with the concepts of internet telephony, multimedia applications, and SEO

APPLICATION DEVELOPMENT AND DEVOPS ON CLOUD LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: **Application development and devops on cloud lab**	**Course Code**	**L-T-P**	**Credits**
	ENSP457	0-0-2	1
Type of Course:	Minor		
Pre-requisite(s), if any: Nil			

Proposed lab experiments

Defined Course Outcomes

COs	
CO 1	**Understand** the fundamental concepts of cloud computing, cloud service models, and cloud-based application development
CO 2	**Acquire** hands-on experience in deploying and managing applications on cloud platforms using various cloud services and tools
CO 3	**Implement** DevOps practices in cloud-based application development, including continuous integration, continuous deployment, and version control
CO 4	**Demonstrate** proficiency in using containerization (Docker and Kubernetes), Infrastructure as Code (IaC), and serverless computing for cloud-based application deployment and scaling
CO 5	**Develop** skills in application monitoring, logging, security, and performance optimization for cloud-based applications, and apply cost optimization strategies for cloud resources

Experiment No.	Experiment Title	Mapped COs
1	Setting up a cloud-based development environment	CO1
2	Deploying a web application on a cloud platform	CO1, CO2
3	Containerizing and deploying with Docker and Kubernetes	CO2, CO4
4	Implementing Continuous Integration (CI)	CO2, CO3
5	Implementing Infrastructure as Code (IaC)	CO2, CO4
6	Configuring auto-scaling for an application	CO2
7	Implementing high availability and disaster recovery	CO2
8	Building a serverless application	CO2, CO4
9	Implementing application monitoring and logging	CO2
10	Creating and managing a cloud-based database	CO2
11	Integrating third-party APIs with an application	CO2
12	Implementing security best practices	CO2
13	Managing application configurations	CO2
14	Performing load testing on an application	CO2
15	Migrating an application to the cloud	CO2
16	Implementing DevOps practices	CO3
17	Deploying a multi-tier application architecture	CO1, CO2
18	Integrating CI/CD pipeline with version control	CO2, CO3
19	Implementing blue-green deployment	CO3
20	Implementing Canary deployment	CO3
21	Performance optimization in cloud	CO2, CO4
22	Implementing feature flags	CO2, CO3

23	Implementing cost optimization strategies	CO2, CO5
24	Deploying a microservices-based application	CO2, CO4
25	Real-world project showcasing cloud-based development	CO1, CO2, CO3, CO4, CO5

MINOR PROJECT-III

Department:	**Department of Computer Science and Engineering**		
Course Name: Minor Project-III	**Course Code**	**L-T-P**	**Credits**
	ENSI451	---	2
Type of Course:	Project		
Pre-requisite(s), if any: NA			

- Students expected to develop a basic project that demonstrates the application of learnings from studied subjects.
- Students are required to submit a hard copy of project file as per the template (Provided at the end of Handbook). File needs to be submitted in spiral bind.
- Project will be evaluated on the scale of 100 with following evaluation criteria.
    - Project idea & features (10)
    - Literature review (10)
    - Tools & Techniques employed (10)
    - Methodology (10)
    - Presentation of Results and its usefulness (20)
    - Implementation and its understand ability (10)
    - Meetings & comments by guide (20)
    - Research paper (10)

File format for Minor project

1.	Abstract	Page No.
2.	Introduction (description of broad topic)	
3.	Motivation	
4.	Literature Review	
5.	Gap Analysis	
6.	Problem Statement	
7.	Objectives	
8.	Tools/platform used	
9.	Methodology	
10.	Experimental Setup	
11.	Evaluation Metrics	
12.	Results And Discussion	
13.	Conclusion & Future Work	
14.	References	
15.	Annexure I: Responsibility Chart	
16.	Annexure II: Screenshots of all the MS-Team Meetings with links (online)/ handwritten comments(offline) from guide	
17.	Annexure III Complete implementation code	
18.	Annexure IV Research Paper (Published/Submitted)	