K.R. MANGALAM UNIVERSITY

SCHOOL OF ENGINEERING AND TECHNOLOGY

B.Sc. (Hons.) Computer Science with IBM Collaboration

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 endeavors, 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

Welcome to the B.Sc. (Hons.) Computer Science program with IBM Collaboration! This program brings together the best of academic excellence and industry partnership to offer an unparalleled learning experience to our students. Computer Science has emerged as the driving force behind technological advancements, reshaping the world we live in. As we move towards a digital age, the demand for skilled computer science professionals has grown exponentially. With our collaboration with IBM, a global leader in technology and innovation, we aim to equip our students with the knowledge and skills required to excel in this dynamic field. This undergraduate program is carefully designed to provide students with a comprehensive understanding of computer science principles and their practical applications. Throughout the course of four years, students will delve into programming, software development, data structures, algorithms, and other core concepts, all enriched by the expertise and resources of IBM. Our collaboration with IBM brings numerous benefits, including exposure to state-of-the-art technologies, industry-relevant projects, and access to IBM's cutting-edge tools and platforms. This collaboration ensures that our students are not only well-versed in theoretical concepts but also gain valuable hands-on experience, making them job-ready from day one. At the heart of our program lies the commitment to fostering critical thinking, problem-solving, and creativity. Our students will have the opportunity to work on challenging real-world projects, enabling them to apply their knowledge in practical scenarios and develop a problem-solving mindset. We take pride in our student-centric approach, emphasizing a supportive learning environment and personalized guidance from our esteemed faculty members. Our mission is to nurture the next generation of computer science professionals who can drive innovation and make a positive impact on society. As you embark on this educational journey, we encourage you to embrace the opportunities that lie ahead and to immerse yourself in the world of computer science with the guidance of IBM. We are excited to see you grow, learn, and thrive in this collaborative and transformative academic experience.

Objectives of the Program

Objective 1: Acquire Profound Technical Knowledge

The primary objective of pursuing a B.Sc. (Hons.) in Computer Science with IBM Collaboration is to gain a deep understanding of fundamental computer science principles and advanced concepts. Through a well-rounded curriculum and exposure to IBM technologies, students will develop expertise in programming languages, algorithms, data structures, databases, software engineering, artificial intelligence, and other essential areas of computer science.

Objective 2: Master IBM Technologies and Tools

The collaboration with IBM offers a unique opportunity for students to familiarize themselves with industry-leading technologies and tools. By incorporating IBM's software and platforms into the coursework, students can gain hands-on experience in working with cutting-edge solutions like cloud computing, data analytics, Internet of Things (IoT), and machine learning. This objective aims to prepare students to thrive in the modern tech industry and align their skills with industry demands.

Objective 3: Foster Problem-Solving and Critical Thinking Abilities

Another crucial objective of the B.Sc. (Hons.) program with IBM Collaboration is to cultivate strong problem-solving and critical thinking skills. By engaging in real-world case studies, projects, and challenges posed by IBM, students will be encouraged to apply their theoretical knowledge to practical scenarios. This approach helps them develop a systematic approach to analyzing problems, designing effective solutions, and critically evaluating outcomes, preparing them to be competent problem solvers in their future careers.

Objective 4: Enhance Industry Readiness and Employability

Through the collaboration with IBM, the B.Sc. program aims to provide students with a competitive edge in the job market. By integrating industry-relevant projects, internships, and exposure to IBM's work culture, students will gain valuable professional experience even before graduation. This objective seeks to produce well-rounded computer science graduates with both theoretical knowledge and practical skills, making them highly employable in various IT-related roles and contributing to the industry's growth and innovation.

Career Avenues

There is a large scope of B.Sc. (Hons.) program Jobs for graduates in both the private and public sectors. After B.Sc. (Hons.) program with IBM Collaboration Graduation students pursuing the course are not limited to their specific areas resulting in a vast number of jobs.

B.Sc. (Hons.) program with IBM Collaboration Is one of the most diverse courses in terms of, not just employment opportunities across various domains, but also the scope of higher education for graduates. The scope of a B.Sc. (Hons.) program with IBM Collaboration Is quite vast whether a graduate chooses to work or study further. According to the IT industry, there are over 1.5 million jobs that are going unfilled in India right now. This shows that data is the future and so are Data Scientists!

Some of the areas of recruitment are

- Supply Chain.
- Computer Science.
- Advanced Analytics.
- Artificial Intelligence.
- Network Analysis.
- Machine Learning.
- Management Consulting.
- Predictive Modeling.

Prospective Companies

Amazon, Flipkart, Cognizant, Wipro, IBM, Infosys, Deloitte, Walmart, Genpact, Accenture, Microsoft, Reliance

Duration

3 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

Institution 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 the 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 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 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. Out student’s get an opportunity to learn directly by professionals from industry.

Program Outcome (PO)

PO1. Computational Knowledge: Demonstrate understanding and apply foundational mathematical concepts, computing principles, and domain knowledge to conceptualize computing models that address defined problems.
PO2. Problem Analysis: Possess the ability to identify, critically analyze, and formulate complex computing problems, leveraging fundamental principles from computer science and relevant application domains.
PO3. Solution Design and Development: Transform intricate business scenarios and contemporary issues into problem statements, investigate and comprehend them, and propose comprehensive and integrated solutions using cutting-edge technologies.
PO4. Effective Tool Utilization: Select and proficiently employ contemporary computing tools, techniques, and skills necessary for developing innovative software solutions.
PO5. Professional Ethics: Apply and uphold professional ethics and cybersecurity regulations within a global economic environment, ensuring responsible and ethical computing practices.
PO6. Lifelong Learning: Recognize the necessity for continuous learning and cultivate the ability to actively engage in professional development as a computing professional.
PO7. Project Management: Possess the competence to understand and apply management and computing principles in order to successfully manage projects in multidisciplinary environments.
PO8. Proficient Communication: Demonstrate effective communication skills, both within the computing community and with society at large, by comprehending and producing clear and impactful documentation and presentations.

PO9. Individual and Teamwork: Exhibit the ability to effectively collaborate and contribute as a member or leader within diverse teams operating in multidisciplinary environments.
PO10. Innovation and Entrepreneurship: Identify opportunities, demonstrate an entrepreneurial mindset, and leverage innovative ideas to generate value and contribute to the betterment of individuals and society.

Program Educational Objectives (PEO)

PEO1 - Acquire a comprehensive understanding of computer science concepts, theories, and practical applications, along with the ability to apply them in real-world scenarios.

PEO2 - Develop technical and analytical skills required to design, implement, and evaluate computer-based systems and solutions in collaboration with IBM professionals and industry experts.

PEO3 - Foster critical thinking, creativity, and innovation through exposure to emerging technologies and research trends in computer science and their practical applications.

PEO4 - Cultivate professional and ethical values, communication, and leadership skills to effectively collaborate in diverse teams and solve complex problems with a global perspective with a focus on life long learning.

Program Specific Outcomes (PSO)

PSO1 - Demonstrate a comprehensive understanding of computer science concepts, theories, and practical applications.

PSO2 - Design, implement, and evaluate computer-based systems and solutions in collaboration with IBM professionals and industry experts.

PSO3 - Engage in critical thinking, creativity, and innovation by staying updated with emerging technologies and research trends in computer science.

PSO4 - Demonstrate professional and ethical values, effective communication, and leadership skills to solve complex problems in diverse teams with a global perspective.

Program Highlights

- The program offers a curriculum designed in collaboration with IBM, a global technology leader. This means the curriculum is aligned with industry trends, cutting-edge technologies, and real-world challenges, ensuring students receive education relevant to the current and future needs of the IT industry.
- Focusing on the development of intelligent solutions, the specialization concentrates on new technologies within the advanced computing space and the use of intelligent solutions.
- Specifically designed curriculum in consultation with industry insiders and experts
- The program offers internships and co-op opportunities in collaboration with IBM. This provides students with the chance to gain practical, hands-on experience within the company, working on real projects alongside professionals. Such experiences can enhance students' resumes, build professional networks, and increase their chances of future employment with IBM or other top tech companies. Globally Valid Digital Badge and Training at Big Data Analytics Lab
- Ensuring absolute preparedness for successful career progression

Program Scheme

Semester I

SNO	Category	COURSE CODE	COURSE TITLE	L	T	P	C
1	Major	ENBC101	Web Technologies	4		-	4
2	Major	ENBC103	MATLAB Programming	4	-	-	4
3	SEC	SEC050	Linux Environment Lab	-	-	2	2
4	Minor	ENSP101	Clean Coding with Python	4		0	4
5	Major	ENBC151	Web Technologies Lab	-	-	2	1
6	Major	ENBC153	Matlab Programming Lab		-	2	1
7	Minor	ENSP151	Clean Coding with Python Lab	-	-	2	1
8	VAC		Environmental Studies & Disaster Management (Online Moodle)	2	-	-	2
9	Major	ENBC105	Fundamentals of Software Engineering	4	-	-	4
			Total	**18**	**0**	**8**	**23**

Semester II

SN	Category	COURSE CODE	COURSE TITLE	L	T	P	C
1	Minor	ENSP112	Introduction to R Programing	4	-	-	4
2	Major	ENBC102	Introduction to Discrete Structure	3	1	-	4
3	Major	ENBC104	Basics of Operating Systems	3	1	-	4
4	Major	ENBC106	Concepts of Object-Oriented Programming Using C++	3	1	-	4
5	Minor	ENSP164	Introduction to R Programing Lab	-	-	2	1
6	Major	ENBC152	Basics of Operating Systems Lab	-	-	2	1
7	Major	ENBC154	Concepts of Object-Oriented Programming Using C++ Lab	-	-	2	1
8	VAC		Extension Activities (community engagement service)	3	-	-	2
9			Open Elective-I	3	-	-	3
			Total	**19**	**3**	**6**	**24**

Semester III

S.N	Category	Course Code	Course Title	L	T	P	C
1	Major	ENBC201	Introduction to Data Structures	3	1	-	4
2	Minor	ENSP205	Fundamentals of Machine Learning	4	-	-	4
3	Major	ENBC203	Basics of Probability & Statistics	4	-	-	4
4	Major	ENBC205	Introduction to Java Programming	3	1	-	4
5	AEC	AEC011	Life Skills for Professionals-I	3	-	-	3
6	Major	ENBC251	Introduction to Java Programming Lab	-	-	2	1
7	Major	ENBC253	Introduction to Data Structures Lab	-	-	2	1
8	Minor	ENSP257	Machine Learning Lab	-	-	2	1
9	VAC		VAC -3				2
10	Summer Internship	SIBC251	Summer Internship/Project-I		-	-	2
TOTAL				**17**	**2**	**6**	**26**

Semester IV

S. N	Category	Course Code	Course Title	L	T	P	C
1	Major	ENBC202	Fundamentals of Algorithm Design & Analysis	3	1	-	4
2	Major	ENBC204	Introduction to Database Management Systems	3	1	-	4
3	Major	ENBC206	Introduction to Computer Networks	3	1	-	4
4	Major	ENBC252	Introduction to Database Management Systems Lab	-	-	2	1
5	Major	ENBC254	Fundamentals of Algorithm Design & Analysis Lab	-	-	2	1
6	Major	ENBC256	Introduction to Computer Networks Lab	-	-	2	1
7	AEC	AEC012	Life Skills for Professionals-II	3	-	-	3
8	Proj	SIBC252	Minor Project-I				2
9	SEC	SEC036	Competitive Coding Lab			4	2
10	Open Elective		Open Elective-II	3	-	-	3
			TOTAL	**15**	**3**	**10**	**25**

Semester V

S.No	Category	Course Code	Course Title	L	T	P	C
1	Major	ENBC301	Computer Organization and Architecture	4	-	-	4
2	Minor	ENSP309	Big Data Analysis with Scala and Spark	4	-	-	4
3	Minor		Department Elective -I	4	-	-	4
4	Minor		Department Elective -I Lab	-	-	2	1
5	Minor		Department Elective -II	4	-	-	4
6	Minor		Department Elective -II lab	-	-	2	1
7	Minor	ENSP359	Big Data Analysis Lab	-	-	2	1
8	Summer Internship	SIBC351	Summer Internship/Project-II				2
9	AEC	AEC013	Life Skills for Professionals-III	3	-	-	3
10			**TOTAL**	**19**	**0**	**6**	**24**

Semester VI

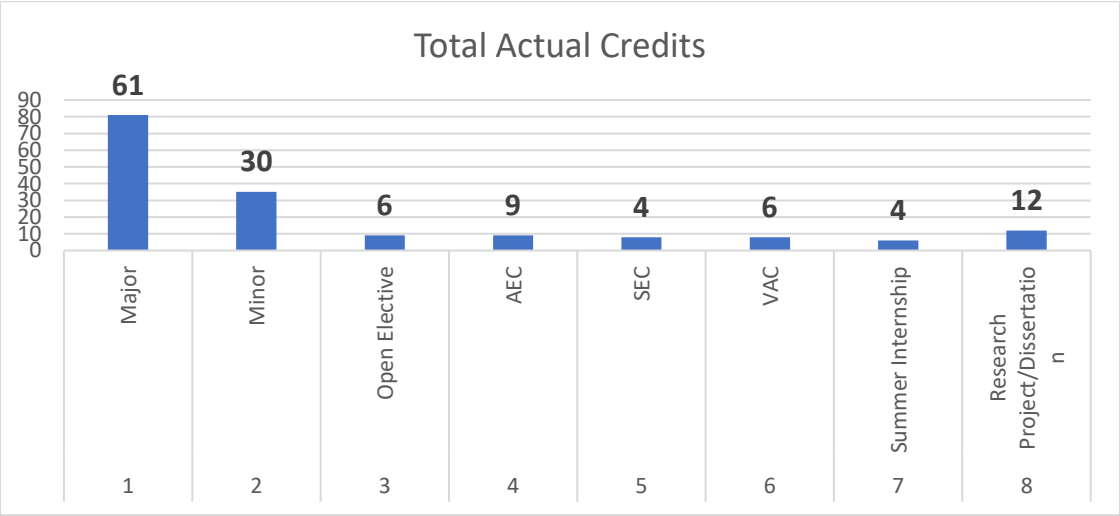
S.No	Category	Course Code	Course Title	L	T	P	C
1	Project	SIBC352	Major Project/Indusrial Training/Startup	-	-	-	12
			TOTAL	**16**	**12**	**8**	**12**
TOTAL CREDITS							**134**

DSE -1						
(i)	ENSP401	Computational Services in The Cloud	4	-	-	4
	ENSP451	Computational Services in The Cloud Lab	-	-	2	1
(ii)	ENSP403	Microsoft Azure Cloud Fundamentals	4	-	-	4
	ENSP453	Microsoft Azure Cloud Fundamentals Lab	-	-	2	1
(iii)	ENSP405	Storage and Databases on Cloud	4	-	-	4
	ENSP455	Storage and Databases on Cloud Lab	-	-	2	1
(iv)	ENSP407	Application Development and DevOps on Cloud	4	-	-	4
	ENSP457	Application Development and DevOps on Cloud Lab	-	-	2	1
DSE -2						
(i)	ENSP409	Mobile Application Development using iOS	4	-	-	4
	ENSP459	Mobile Application Development using iOS Lab	-	-	2	1
	ENSP411	DevOps & Automation	4	-	-	4

(ii)	ENSP461	DevOps & Automation Lab	-	-	2	1
(iii)	ENSP413	.Net FRAMEWORK	4	-	-	4
	ENSP463	.Net FRAMEWORK Lab	-	-	2	1
(iv)	ENSP415	New Age Programming languages	4	0	0	4
	ENSP465	New Age Programming languages Lab	0	0	2	1

Total Credits: 134

Categorization of Courses



Semester I

Fundamentals of Web Technologies

Department:	**Department of Computer Science and Engineering**		
Course Name: Fundamentals of Web Technologies	**Course Code**	**L-T-P**	**Credits**
	ENBC101	4-0-0	4
Type of Course:	Major		
Pre-requisite(s), if any:			
Frequency of offering (check one): Odd semester			
Brief Syllabus: The objective is to Analyse a web page and identify its elements and attributes , Create web pages using XHTML and Cascading Style Sheets. Build dynamic web pages using JavaScript (Client-side programming). Create XML documents and Schemas. Build interactive web applications using AJAX.			
UNIT WISE DETAILS			
Unit Number: 1	**Introduction to Web Technology**	**No. of hours: 8**	
Content Summary: Concept of WWW, Internet and WWW, HTTP Protocol: Request and Response, Web browser and Web servers, Features of Web 2.0, Common terminology: IP Addressing, URLs, Domain names. Website Creation and maintenance, Web Hosting and Publishing Concepts, Search Engines and their working. HTML: Introduction to HTML, HTML Document structure tags, HTML comments, Text formatting, inserting special characters, anchor tag, adding images and sound, lists: types of lists, tables, frames and floating frames, Developing Forms, Image maps, formatting, and fonts, commenting code, color, hyperlink, lists, tables, images, forms, XHTML, Meta tags, Character entities, frames and frame sets,			
Unit Number: 2	**Client-side scripting**	**No. of hours: 12**	
Content Summary: avaScript - Data Types, Control Statements, operators, Built-in and User Defined Functions, Objects in JavaScript, Handling Events. HTML Document Object Model. Page Styling: Separation of content and presentation in HTML, Cascading Style Sheets - Types of Style Sheets – Internal, inline, and External style sheets, customizing common HTML elements, types of CSS selectors.			

Unit Number: 3	Concepts of effective web design	**No. of hours: 12**

Content Summary:
Concepts of effective web design, Web design issues including Browser, Bandwidth and Cache, Display resolution, Look and Feel of the Website, Page Layout and linking, User centric design, Sitemap, Planning and publishing website, Designing effective navigation, Browser architecture and Web site structure.

Unit Number: 4	XML	**No. of hours: 8**

Content Summary:
Introduction to XML-Mark up languages, Features of Mark-up languages, XML Naming rules, building block of XML, Document, Difference between HTML & XML, Components of XML, XML Parser, DTD's Using XML with HTML and CSS.
Introduction to Web Services, UDDI, SOAP, WSDL, Web Service Architecture, Developing and deploying web services. AJAX –Introduction AJAX programming, improving web page performance using AJAX.

***Self-Learning Components:**

https://www.mygreatlearning.com/web-development/free-courses3)
https://www.simplilearn.com/certifications/web-development-courses

Reference Books:

1.Web Technologies, Uttam K. Roy, Oxford University Press
2. HTML Black Book, Stephen Holzner, Wiley Dreamtech.
3. Web Technology, Rajkamal, Tata McGraw-Hill.
4. Web Technologies: A Computer Science Perspective, Jeffrey C. Jackson, Pearson.
5. XML: How to Program, Deitel & Deitel Nieto

Define Course Outcomes (CO)

CO1	Create a well-designed and well-formed, professional Web site utilizing the most current standards and practices
CO2	Demonstrate knowledge in web technologies including HTML, XHTML, CSS, image editing software, web authoring software, and client-side scripting
CO3	Create client-side scripts to add interactivity to Web pages
CO4	Select appropriate Web tools for a Web development project

CO5	Identify Web authoring obstacles created by the availability of various web browsers and mark-up language versions

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

CO-PO Mapping

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

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	
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	
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	
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	
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

FUNDAMENTALS OF WEB TECHNOLOGIES LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: Fundamentals of Web Technologies Lab	**Course Code**	**L-T-P**	**Credits**
	ENBC151	0-0-2	1
Type of Course:	Major		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	Course Outcomes (COs)
CO 1	Analyze a web page and identify its elements and attributes.
CO 2	Create web pages using XHTML and Cascading Style Sheets. ·
CO 3	Build dynamic web pages using JavaScript (Client side programming). ·
CO 4	Create XML documents and Schemas.

List of Experiments

Ex. No	Experiment Title	Mapped CO/COs
1	Write HTML/Java scripts to display your CV in Web Browser	CO1
2	Creation and annotation of static web pages using any HTML editor.	CO1
3	Write a program to use XML and JavaScript for creation of your homepage.	CO4
4	Write a program in XML for creation of DTD which specifies a particular set of rules.	CO4
5	Create a Stylesheet in CSS/XSL and display the document in Web Browser	CO2
6	Create a Registration Form with Table	CO2
7	CSS : Inline Style , Internal Style ,and External Style Sheets	CO3
8	JavaScript & HTML:	CO2,CO3

	Use user defined function to get array of values and sort them in ascending order · Demonstrate String and Math Object's predefined methods · Demonstrate Array Objects and Date Object's predefined methods · Exception Handling · Calendar Creation : Display all month · Event Handling · Validation of registration form · Open a Window from the current window · Change color of background at each click of button or refresh of a page · Display calendar for the month and year selected from combo box ·OnMouseover event	
9	XML ☐ Create any catalog ☐ Display the catalog created using CSS or XS	CO4

Programming in MATLAB

Department:	**Department of Computer Science and Engineering**		
Course Name: Programming in MATLAB	**Course Code**	**L-T-P**	**Credits**
	ENBC103	4-0-0	4
Type of Course:	Major		
Pre-requisite(s), if any:			
Brief Syllabus: MATLAB is a powerful software tool used in engineering, mathematics, and science for numerical computations, data analysis, and simulation. It has a user-friendly interface, supports arrays, matrices, and complex numbers, and allows scripting for automation. MATLAB includes graphing capabilities, Simulink for system modeling, and extensive mathematical functions for integration, solving equations, and transforms. It is widely used by researchers, engineers, and scientists for a variety of applications.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction to MATLAB**	**No. of hours: 10**	
Content Summary: Brief Introduction, Installation of MATLAB, History, Use of MATLAB, Key features, MATLAB Window, Command Window, Workspace, Command history, Setting directory, Working with the MATLAB user interface, Basic commands, Assigning variables, Operations with variables, Data files and Data types: Character and string, Arrays and vectors, Column vectors, Row vectors, Arithmetic operations, Operators and special characters, Mathematical and logical operators, Solving arithmetic equations.			
Unit Number: 2	**Title: Operations & Plots**	**No. of hours: 14**	

Content Summary:

Crating rows and columns Matrix, Matrix operations: Finding transpose, determinant and inverse, Solving matrix, Trigonometric functions, Complex numbers, fractions, Real numbers, Complex numbers, Working with script tools, Writing Script file, Executing script files, The MATLAB Editor, Saving m files

Plotting vector and matrix data, Plot labelling, curve labelling and editing, Basic Plotting Functions, Creating a Plot Plotting Multiple Data Sets in One Graph, Specifying Line Styles and Colors, Graphing Imaginary and Complex Data Figure, Windows Displaying, Multiple Plots in One Figure, Controlling the Axes, Creating Mesh and Surface About Mesh and Surface Visualizing Subplots.

Unit Number: 3	**Title: MATLAB Simulink**	**No. of hours: 8**

Content Summary:

Introduction of Simulink, Simulink Environment & Interface, Study of Library, Circuit Oriented Design, Equation Oriented Design, Model Subsystem Design, Connect Call back to subsystem, Application. Automating commands with scripts, writing programs with logic and flow control, Control statement, Programming Conditional Statement, Writing functions, Programming, Examples

Unit Number: 4	**Title: Symbolic Math in MATLAB**	**No. of hours: 8**

Calculus: Numerical Integration, Linear Algebra, Roots of Polynomials, Algebraic equations, Differential Equations (1st& 2nd order), Transforms (Fourier, Laplace, etc), Ordinary Differential equations, Examples of few ODEs.

***Self-Learning Components:**

1)https://www.mygreatlearning.com/academy/learn-for-free/courses/matlab.

2) https://www.simplilearn.com/free-matlab-online-course-skillup

Reference Books:

1. Ian. J. Lyod , "Information technology law" , Information Technology Act 2000, its amendment and IT Rules, 2014.

2. Yee fen Lim , "Cyber space law commentaries and Materials", second edition, Galexia Consulting Pty Ltd, Australia.
3. William Stallings and Lawrie Brown "Computer Security: Principles and Practice" (2020).

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** the fundamental concepts and functionalities of MATLAB, including its history, installation process, and key features.
CO2	**Analyze** the different problems related to matrix manipulation, trigonometric functions, complex numbers, and fractions. Utilize script tools to write and execute script files.
CO3	**Apply** MATLAB Simulink to model and simulate systems.
CO4	**Evaluate** capabilities in MATLAB to solve mathematical problems related to calculus, linear algebra, polynomials, algebraic equations, differential equations, and transforms.

Cos Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© Knowledge Understand Apply Analyze Evaluate Create	Affective levels(A) Receiving Responding Valuing Organizing Characterizing	Psychomotor levels(P) Imitation Manipulation Precision Articulation Improving
	C1	-	P1

CO1			
CO2	C2	A3	-
CO3	C3	-	-
CO4	C4	-	P2
CO5	C5	A4	-

CO-PO Mapping

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

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	1	-	-
CO2	3	2	-	2
CO3	2	-	3	-
CO4	-	2	3	2
CO5	-	1	-	3

Relevance of the Syllabus to various indicators

Unit I	Introduction to MATLAB
Local	Understanding MATLAB and its applications can be relevant at the local level for educational institutions, research organizations, and industries that use MATLAB for various data analysis and problem-solving tasks.
Regional	MATLAB can be used in regional projects related to engineering, science, finance, and other fields, contributing to regional development.
National	Its applications in engineering, healthcare, finance, and other sectors can have a significant impact on the nation's progress and development.
Global	Its relevance extends worldwide, and the skills gained can be beneficial for individuals seeking opportunities in international collaborations, research, or global corporations.
Employability	Proficiency in MATLAB can enhance an individual's employability across various industries and job roles.
Entrepreneurship	-
Skill Development	Learning MATLAB helps individuals develop skills in programming, data manipulation, mathematical analysis, and problem-solving, fostering overall skill development.

Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	Operations
Local	Local researchers and students can utilize MATLAB's matrix capabilities for their projects and assignments.
Regional	-
National	-
Global	MATLAB is a globally used tool, and its matrix operations have a wide-reaching impact.
Employability	Proficiency in MATLAB's matrix operations is highly valued in various industries.
Entrepreneurship	-
Skill Development	Learning matrix operations in MATLAB helps individuals develop valuable skills in linear algebra, which are essential for data analysis, engineering, and scientific research.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	MATLAB Simulink
Local	Local researchers and students can utilize Simulink for their projects related to control systems, signal processing, and circuit design.
Regional	-

National	-
Global	proficiency in Simulink can be valuable for individuals seeking opportunities in international collaborations and research projects.
Employability	Proficiency in Simulink is highly valued in engineering and technical industries.
Entrepreneurship	-
Skill Development	Learning Simulink helps individuals develop skills in system-level modeling, simulation, and design, which are essential for engineering and research tasks.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	Symbolic Math in MATLAB
Local	Understanding numerical integration techniques and their applications in calculus is relevant at the local level for educational institutions, research centers, and industries that deal with data analysis, optimization, and simulations.
Regional	-
National	-
Global	Its relevance extends worldwide, and its applications impact global scientific advancements.
Employability	Proficiency in calculus, numerical integration, and differential equations is highly valued in technical industries such as engineering, data analysis, and scientific research.
Entrepreneurship	-
Skill Development	Learning calculus, numerical integration, and differential equations helps individuals develop strong analytical and

	problem-solving skills, which are essential for scientific research and engineering tasks.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4,9,16
NEP 2020	Aligns with the objectives and principles of NEP 2020, such as: Holistic Development, Skill Development Digital Literacy
POE/4th IR	The principles of the Fourth Industrial Revolution, fostering holistic development, skill development, digital literacy, and addressing emerging challenges in the digital era.

PROGRAMMING IN MATLAB LAB

Department:	**Department of Computer Science and Engineering**		
Course Name: Programming in MATLAB Lab	**Course Code**	**L-T-P**	**Credits**
	ENBC153	0-0-2	1
Type of Course:	Major		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	Statements
CO 1	Understand MATLAB Fundamentals (Knowledge) Students will be able to demonstrate a basic understanding of MATLAB syntax, variables, data types, and operators.
CO 2	Apply MATLAB Programming Techniques (Application) Students will be able to write MATLAB programs to solve various mathematical problems, implement algorithms, and manipulate arrays and matrices.
CO 3	Analyze and Evaluate MATLAB Code (Analysis) Students will be able to analyze existing MATLAB code, identify errors, and debug the programs to ensure correct functionality.
CO 4	Create Custom Functions and Plots (Synthesis) Students will be able to create their own user-defined functions in MATLAB, encapsulate code for reusability, and generate complex plots to visualize data..
CO 5	Solve Engineering and Scientific Problems using MATLAB (Evaluation) Students will be able to apply MATLAB to solve real-world engineering and scientific problems, interpret the results, and evaluate the effectiveness of their solutions.

List of Experiments:

Ex. No	**Experiment Title**	**Mapped CO/COs**
1	Program to find the sum of elements in an array.	CO1

2	Program to calculate the factorial of a given number using a loop.	CO1, CO2
3	Program to check whether a given number is prime or not.	CO1
4	Program to find the Fibonacci series up to a given number of terms.	CO1, CO2
5	Program to calculate the roots of a quadratic equation.	CO1, CO2, CO5
6	Program to implement bubble sort for sorting an array.	CO1, CO2
7	Program to calculate the mean, median, and mode of a dataset.	CO1, CO2, CO3, CO4
8	Program to plot a sine wave and cosine wave on the same graph.	CO1, CO2, CO4
9	Program to implement matrix addition and subtraction. or solving a first-order ordinary differential equation	CO1, CO2
10	Program to find the determinant of a 3x3 matrix.	CO1, CO2
11	Program to calculate the area and perimeter of a circle given its radius.	CO1, CO2
12	Program to implement linear regression for a given dataset.	CO1, CO2, CO4
13	Program to convert a decimal number to binary.	CO1, CO2
14	Program to perform element-wise multiplication of two matrices.	CO1, CO2
15	Program to implement the Simpson's 1/3 rule for numerical integration.	CO1, CO2
16	Program to generate a random password of a given length.	CO1, CO2
17	Program to implement the Gauss-Seidel method to solve a system of linear equations. .	CO 5

18	Program to implement a simple calculator with basic arithmetic operations.	CO 4
19	Program to simulate a simple dice rolling game	CO 2
20	Mini Project: Develop a MATLAB program to perform basic image processing operations such as image enhancement, filtering, edge detection, and image segmentation. Apply these techniques to analyze and manipulate images.	CO2, CO3, CO5
21	Mini Project: Design a MATLAB program to process and analyze signals, such as audio signals or ECG signals. Implement filtering, noise reduction, Fourier analysis, and plotting of signal waveforms.	CO2, CO3,CO4, CO5
22	Mini Project: Implement various numerical methods in MATLAB, such as solving systems of linear equations, finding roots of nonlinear equations, numerical integration, and solving ordinary differential equations. Apply these methods to solve engineering and scientific problems.	CO2, CO3, CO5
23	Mini Project: Use MATLAB to analyze and visualize data from real-world datasets. Perform statistical analysis, data interpolation, curve fitting, and generate meaningful visualizations such as plots and graphs to present the results.	CO2, CO3,CO4, CO5
24	Mini Project: Build a MATLAB program to simulate control systems. Design and analyze feedback control systems, implement controllers, and simulate the system's response. Evaluate the stability and performance of the control system under various scenarios.	CO2, CO3, CO5

Linux Environment Lab

Department:	**Department of Computer Science and Engineering**		
Course Name: Linux Environment Lab	**Course Code**	**L-T-P**	**Credits**
	SEC050	0-0-2	2
Type of Course:	SEC		
Pre-requisite(s), if any: Basic understanding of computer systems and familiarity with operating systems			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	**Apply** Linux operating system concepts and commands.
CO 2	**Utilize** scripting and automation techniques in Linux.
CO 3	**Manage** system resources and security in Linux.
CO 4	**Implement** networking and server configurations in Linux.
CO 5	**Explore** emerging technologies and trends in Linux for data science.

List of Experiments:

Ex. No	Experiment Title	Mapped CO/COs
1	Installing Linux Operating System	CO 1
2	Exploring the System	CO 1
3	Working with Directories and Files	CO 1
4	File Manipulation and Redirection	CO 1
5	Searching for Files	CO 1
6	Understanding Display and Window Managers	CO 1
7	User and Group Management	CO 1, CO 3
8	Package Management and Software Building	CO 1, CO 3
9	Device Management and Mounting	CO 1
10	Introduction to Shell Scripting	CO 2
11	Shell Scripting: Control Structures	CO 2
12	Process Management and Automation	CO 2
13	File Permissions and Security	CO 3

14	Network Configuration and Troubleshooting	CO 4
15	Linux Servers and Web Hosting	CO 4
16	Network Security in Linux	CO 4
17	Introduction to Emerging Technologies in Linux	CO 5
18	Docker and Containerization	CO 5
19	Cloud Platforms and Deployment	CO 5
20	Linux for IoT and Embedded Systems	CO 5
21	Data Processing with Linux Tools	CO 5
22	Data Analysis with Linux Tools	CO 5
23	Machine Learning with Linux	CO 5
24	Big Data Analytics with Linux	CO 5
25	Visualization and Reporting	CO 5

Detailed List of experiments

1. Installing Linux Operating System:
    - Session: Introduction to different Linux distributions and their installation methods.
    - Exercise: Install a Linux distribution of choice on a virtual machine or physical hardware.
    - Project: Set up a dual-boot system with Linux and another operating system.
2. Exploring the System:
    - Session: Familiarize with basic system commands and file system navigation.
    - Exercise: Use commands like **ls**, **cd**, **pwd**, and **man** to navigate and explore the file system.
    - Project: Create a directory structure and organize files based on a specific criterion.
3. Working with Directories and Files:
    - Session: Learn directory and file manipulation commands.
    - Exercise: Create, rename, move, and delete directories and files using commands like **mkdir**, **mv**, and **rm**.
    - Project: Develop a script to automate a specific file management task.
4. File Manipulation and Redirection:
    - Session: Understand file manipulation and input/output redirection techniques.
    - Exercise: Use commands like **touch**, **cat**, and redirection operators (>, >>, <) to create, view, and manipulate file contents.
    - Project: Write a script to process a text file and extract specific information.
5. Searching for Files:

- Session: Learn about file searching techniques using commands like **grep** and **find**.
- Exercise: Search for specific patterns or files within directories using **grep** and **find** commands.
- Project: Develop a script to search for files based on user-defined criteria.

6. Understanding Display and Window Managers:
   - Session: Introduce X Window System, display managers, and window managers.
   - Exercise: Configure and customize the window manager settings.
   - Project: Explore different window managers and compare their features and performance.
7. User and Group Management:
   - Session: Understand user and group management commands and concepts.
   - Exercise: Create, modify, and delete user accounts and groups using commands like **useradd**, **usermod**, and **groupadd**.
   - Project: Implement a script to automate user and group management tasks.
8. Package Management and Software Building:
   - Session: Learn package management systems and software installation methods.
   - Exercise: Use package management commands like **apt** or **yum** to install, update, and remove software packages.
   - Project: Build a custom package from source code and install it on the system.
9. Device Management and Mounting:
   - Session: Understand device management, device files, and mounting concepts.
   - Exercise: Identify different types of devices and mount/unmount them using commands like **mount** and **umount**.
   - Project: Automate the mounting process for specific devices upon system startup.
10. Introduction to Shell Scripting:
    - Session: Introduce shell scripting and basic scripting concepts.
    - Exercise: Write simple shell scripts to perform tasks like printing system information or automating repetitive tasks.
    - Project: Develop a script that performs system monitoring and sends alerts when specific conditions are met.
11. Shell Scripting: Control Structures:
    - Session: Explore control structures in shell scripting (if-else, loops).
    - Exercise: Write shell scripts with conditional statements and loops to solve specific problems.
    - Project: Create a script that performs data backup and retention based on user-defined policies.
12. Process Management and Automation:
    - Session: Learn process management commands and techniques.

- Exercise: Manage running processes, monitor resource usage, and control process execution using commands like **ps**, **top**, and **kill**.
- Project: Develop a script that monitors and restarts a specific service if it becomes unresponsive.

13. File Permissions and Security:
    - Session: Understand file permissions, ownership, and basic security measures.
    - Exercise: Set and modify file permissions, change ownership, and manage access control.
    - Project: Create a script that audits file permissions and reports any security vulnerabilities.
14. Network Configuration and Troubleshooting:
    - Session: Configure network interfaces, troubleshoot network connectivity issues.
    - Exercise: Configure network settings manually, diagnose and fix common network problems.
    - Project: Implement a script that automates network configuration for different network scenarios.
15. Linux Servers and Web Hosting:
    - Session: Introduce Linux server administration and web hosting concepts.
    - Exercise: Install and configure server software like Apache or Nginx, host a basic website.
    - Project: Deploy a web application on a Linux server and configure it for optimal performance.
16. Network Security in Linux:
    - Session: Explore network security measures and techniques in Linux.
    - Exercise: Implement firewall rules, set up secure remote access, and monitor network traffic.
    - Project: Design and implement a secure network architecture for a given scenario.
17. Introduction to Emerging Technologies in Linux:
    - Session: Discuss emerging technologies and trends in the Linux ecosystem.
    - Exercise: Explore technologies like containers, cloud platforms, IoT, etc., and their integration with Linux.
    - Project: Research and present a case study on the application of an emerging technology in a real-world data science project.
18. Docker and Containerization:
    - Session: Understand containerization concepts and Docker fundamentals.
    - Exercise: Build, run, and manage containers using Docker commands.
    - Project: Containerize a data science application or workflow using Docker.
19. Cloud Platforms and Deployment:
    - Session: Introduce cloud computing platforms and deployment strategies.

- Exercise: Deploy applications on cloud platforms like AWS, Google Cloud, or Azure.
- Project: Design and deploy a scalable and fault-tolerant data science solution on a cloud platform.

20. Linux for IoT and Embedded Systems:
    - Session: Discuss the role of Linux in IoT and embedded systems.
    - Exercise: Set up and configure a Raspberry Pi or similar device running a Linux distribution.
    - Project: Develop a small-scale IoT project using Linux and connected devices.
21. Data Processing with Linux Tools:
    - Session: Explore command-line tools for data processing and manipulation.
    - Exercise: Use tools like **awk**, **sed**, and **grep** to extract, transform, and analyze data.
    - Project: Develop a data processing pipeline using Linux tools for a specific data analysis task.
22. Data Analysis with Linux Tools:
    - Session: Introduce data analysis tools and frameworks available in the Linux environment.
    - Exercise: Utilize tools like R, Python, or SQL to perform data analysis tasks on Linux.
    - Project: Analyze a real-world dataset using Linux tools and generate insights or visualizations.
23. Machine Learning with Linux:
    - Session: Discuss the integration of Linux with machine learning frameworks and libraries.
    - Exercise: Install and configure machine learning tools like TensorFlow or scikit-learn on Linux.
    - Project: Develop a machine learning model using Linux-based tools for a given problem.
24. Big Data Analytics with Linux:
    - Session: Explore big data analytics tools and technologies on Linux.
    - Exercise: Set up and utilize tools like Hadoop or Apache Spark for big data processing and analysis.
    - Project: Perform large-scale data analysis using Linux-based big data tools on a sample dataset.
25. Visualization and Reporting:
    - Session: Introduce visualization tools and techniques for data representation.
    - Exercise: Use tools like Matplotlib, Tableau, or R libraries to create visualizations on Linux.
    - Project: Develop a dashboard or report presenting insights from a data analysis project using Linux-based visualization tools.

References:

1. "Linux Journey" (Website): A comprehensive online tutorial that covers various topics in Linux, from basic commands to advanced system administration. It provides interactive exercises and practical examples. Website: linuxjourney.com
2. "Linux Documentation Project" (Website): Offers a vast collection of documentation, guides, how-tos, and tutorials on Linux. It covers a wide range of topics and provides detailed explanations and examples. Website: tldp.org
3. "LinuxCommand.org" (Website): Provides a beginner-friendly guide to learning the command line in Linux. It covers basic to advanced command-line usage and offers practical examples and exercises. Website: linuxcommand.org
4. "The Linux Command Line" by William E. Shotts (Book): This book is a comprehensive guide to the Linux command line interface. It covers essential commands, file system navigation, text processing, shell scripting, and more. It includes practical examples and exercises to reinforce learning. [ISBN-13: 978-1593279523]
5. "Linux Bible" by Christopher Negus (Book): A comprehensive guide to Linux system administration and usage. It covers a wide range of topics, including installation, command-line usage, networking, security, and more. It provides step-by-step instructions and real-world examples. [ISBN-13: 978-1119578884]
6. "Linux Pocket Guide" by Daniel J. Barrett (Book): A concise reference guide to essential Linux commands and configurations. It provides quick explanations and examples of commonly used commands, file management, text processing, and system administration tasks. [ISBN-13: 978-1492082809]
7. "Linux Academy" (Online Learning Platform): Offers a wide range of Linux courses and hands-on labs for learners of all levels. It covers various topics, including Linux system administration, shell scripting, DevOps tools, and cloud platforms. Website: linuxacademy.com

Clean Coding with Python

Department:	**Department of Computer Science and Engineering**		
Course Name: Clean Coding with Python	**Course Code**	**L-T-P**	**Credits**
	ENSP101	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			
Brief Syllabus: Python is a language with a simple syntax, and a powerful set of libraries. It is an interpreted language, with a rich programming environment, including a robust debugger and profiler. While it is easy for beginners to learn, it is widely used in many scientific areas for data exploration. This course is an introduction to the Python programming language for students without prior programming experience. This course covers data types, control flow, object-oriented programming, and graphical user interface-driven applications. The examples and problems used in this course are drawn from diverse areas such as text processing, simple graphics creation and image manipulation, HTML and web programming, and genomics.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction to Clean Code**	**No. of hours: 8**	
Content Summary: What is Bad Code? What is Clean Code? Purpose of Clean Code Thought of experienced programmers Meaningful Names Intention Revealing Names Make Meaningful Distinctions Use Pronounceable Names Avoid Encodings and Mental Mappings Difference between smart and professional programmer Class and Method Names Function Size Matters Blocks and Indenting Do only one thing within a function One level of abstraction per function Use Descriptive Names Function Arguments Advantages of Having Less Arguments Command Query Separation			

Prefer Exceptions to Returning Error Codes Extract Try/Catch Blocks Error Handling Is One Thing		
Unit Number: 2	**Title: Introduction to Python**	**No. of hours: 10**
Content Summary: What is Python?, Advantages and disadvantages, Downloading and installing, Which version of Python, Running Python Scripts, Using the interpreter interactively, Using variables, String types: normal, raw and Unicode String operators and expressions, Math operators and expressions, Writing to the screen, Reading from the keyboard, Indenting is significant, The if and elif statements, While Loops, Using List, Dictionaries, Using the for statement, Opening, reading and writing a text file, Using Pandas, the python data analysis library and data frames, Grouping, aggregating and applying, merging and joining, Dealing with syntax errors, Exceptions, Handling exceptions with try/exception.		
Unit Number: 3	**Title: Data Handling and Use Cases**	**No. of hours: 10**
Content Summary: RE Pattern Matching, Parsing Data, Introduction to Regression, Types of Regression, Use Cases, Exploratory data analysis, Correlation Matrix, Visualization using Matplotlib, Implementing linear regression.		
Unit Number: 4	**Title: Advance Concepts**	**No. of hours: 12**
Content Summary: Machine Learning - Algorithm Algorithms – Random forest Super vector Machine Random Forest Build your own model in python Comparison between random forest and decision tree		
***Self-Learning Components:** • Object-oriented programming concepts, • Numpy • File Handling • Jupyter Notebook • PyCharm		
Reference Books: 1. IBM Material		

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** Python syntax and semantics and be fluent in the use of Python flow control and Functions.
CO2	**Implement** **P**ython programs using core data structures like Lists, Dictionaries, and use of Strings Handling methods.
CO3	**Apply** Machine Learning Algorithm on real world problems.

CO4	**Interpretation** of Data, Data Handling and Use Cases.

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	Receiving	P2
CO2	C4, C6	Responding	P2
CO3	C3, C5	Valuing	P4, P5
CO4	C5	Characterizing	P4, P5

CO-PO Mapping

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

CO-PSO Mapping

PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	-	2	-
CO2	2	-	2	-
CO3	3	3	2	-
CO4	3	3	2	-

Relevance of the Syllabus to various indicators

Unit I	Introduction to Clean Code
Local	Yes
Regional	-
National	-

Global	Yes
Employability	Yes
Entrepreneurship	Yes
Skill Development	Yes
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	Introduction to Python
Local	Yes
Regional	-
National	-
Global	Yes
Employability	Yes
Entrepreneurship	Yes
Skill Development	Yes
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	Data Handling and Use Cases
Local	Yes
Regional	-
National	-
Global	Yes
Employability	Yes
Entrepreneurship	Yes
Skill Development	Develops knowledge and skills in data handling & map associated use cases.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	Advance Concepts
Local	Yes
Regional	-
National	-
Global	Yes

Employability	Yes
Entrepreneurship	Yes
Skill Development	Yes
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	The course indirectly aligns with several SDGs. For example, by introducing machine learning algorithms and data analysis techniques, students can learn to address social and environmental challenges through data-driven decision-making. This aligns with SDG 9 (Industry, Innovation, and Infrastructure) and SDG 11 (Sustainable Cities and Communities). Additionally, the course can touch upon topics related to data privacy and responsible use of technology, contributing to SDG 16 (Peace, Justice, and Strong Institutions).
NEP 2020	The NEP emphasizes the development of critical thinking, problem-solving skills, and multidisciplinary education. This course promotes these objectives by providing students with the opportunity to apply Python programming to real-world problems, interpret data, and develop analytical skills. The focus on machine learning algorithms also aligns with the NEP's emphasis on multidisciplinary learning, as it combines concepts from computer science and statistics.
POE/4th IR	The course content aligns well with the demands of the Fourth Industrial Revolution. Python programming, machine learning, and data analysis are key components of the 4IR, and acquiring these skills can prepare students for the future job market. The course enables students to understand and apply machine learning algorithms, which are increasingly utilized in various industries to automate processes, make data-driven decisions, and develop intelligent systems.

Clean Coding with Python Lab

Department:	**Department of Computer Science and Engineering**		
Course Name: Clean Coding with Python Lab	**Course Code**	**L-T-P**	**Credits**
	ENSP151	0-0-2	1
Type of Course:	Minor		
Pre-requisite(s), if any: Basic understanding of computer systems			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Develop solutions to simple computational problems using Python programs.
CO 2	Solve problems using conditionals and loops in Python. Develop Python programs by defining functions and calling them.
CO 3	Implement Python lists, tuples and dictionaries for representing compound data.
CO 4	Implementation of Machine Learning Algorithms.

List of Experiments:

Ex. No	Experiment Title	Mapped CO/COs
1	Develop programs to understand the control structures of python	CO 1
2	Develop programs to implement list	CO 3
3	Develop programs to implement Dictionary	CO 3
4	Develop programs to implement tuples	CO 3
5	Develop programs to implement function with stress on scoping	CO 2
6	Develop programs to implement classes and objects	CO 3
7	Develop programs to implement exception handling.	CO 1
8	Develop programs to implement linear search and binary search.	CO 2
9	Develop programs to implement insertion sort	CO 2
10	Develop programs to implement bubble sort.	CO 2
11	Develop programs to implement quick sort.	CO 2
12	Develop programs to implement heap sort.	CO 2

Fundamentals of Software Engineering

Department:	**Department of Computer Science and Engineering**		
Course Name: Fundamentals of Software Engineering	**Course Code**	**L-T-P**	**Credits**
	ENBC105	4-0-0	4
Type of Course:	Major		
Pre-requisite(s), if any:			
Brief Syllabus: Software engineering is the branch of computer science that creates practical, cost-effective solutions to computing and information processing problems, preferentially by applying scientific knowledge, developing software systems in the service of mankind. This course covers the fundamentals of software engineering, including understanding system requirements, finding appropriate engineering compromises, effective methods of design, coding, and testing, team software development, and the application of engineering tools. The course will combine a strong technical focus with a capstone project providing the opportunity to practice engineering knowledge, skills, and practices in a realistic development.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction**	**No. of hours: 6**	
Content Summary: Introduction- Notion of Software as a Product – characteristics of a good Software Product. Engineering aspects of Software production – the necessity of automation. Job responsibilities of Programmers and Software Engineers as Software developers.			
Unit Number: 2	**Title: Process Models and Program Design Techniques**	**No. of hours: 10**	
Content Summary: Software Development Process Models – Code & Fix model, Waterfall model, Incremental model, Rapid Prototyping model, Spiral (Evolutionary) model. Good Program Design Techniques – Structured Programming, Coupling and Cohesion, Abstraction and Information Hiding. Software Modelling Tools –Data flow Diagrams and UML.			
Unit Number: 3	**Title: Verification and Validation**	**No. of hours: 10**	

Content Summary: Black-Box Testing and White-Box Testing, Static Analysis, Symbolic Execution, and Control Flow Graphs – Cyclomatic Complexity. Introduction to testing of Real-time Software Systems.		
Unit Number: 4	**Title: Software Project Management**	**No. of hours: 14**
Content Summary: Management Functions and Processes, Project Planning and Control, Organization and Intra-team Communication, Risk Management. Software Cost Estimation – underlying factors of critical concern. Metrics for estimating costs of software products – Function Points. Techniques for software cost estimation – Expert judgement, Delphi cost estimation, Work break-down structure and Process breakdown structure, COCOMO, and COCOMO-II.		
Reference Books: 1. Carlo Ghezzi, Fundamentals of Software Engineering, 2nd Edition, PHI, 2002. 2. Ian Sommerville, Software Engineering, 9th Edition, Pearson, 2011. 3. Berzins and Luqi, Software Engineering with Abstraction, 1st Edition, Addison-Wesley, 1991. 4. Martin L. Shooman, Software Engineering – Design, Reliability and Management, McGraw-Hill Education, 1984.		

Define Course Outcomes (CO)

COs	Statements
CO1	Understand the characteristics of a good software product and its role as a software developer.
CO2	Analyze and apply different software development process models.
CO3	Utilize software modeling tools for effective software development.
CO4	Implement software testing techniques and quality assurance measures.
CO5	Develop project management and software cost estimation skills.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels©	Affective levels(A)	Psychomotor levels(P)
	1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A3	P1
C02	C3	A4	P2

CO3	C3	A2	P3
CO4	C3	A2	P4
CO5	C6	A5	P3

CO-PO Mapping

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

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	**Introduction**
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	**Process Models and Program Design Techniques**
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	**Verification and Validation**
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	-
Unit IV	**Software Project Management**
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.
SDG	SDG 4
NEP 2020	-

POE/4th IR	Emphasizes the responsible and ethical development and deployment of the systems.

Fundamentals of Software Engineering Lab

Department:	**Department of Computer Science and Engineering**		
Course Name: Fundamentals of Software Engineering Lab	**Course Code**	**L-T-P**	**Credits**
		0-0-2	1
Type of Course:	Major		
Pre-requisite(s), if any: Basic understanding of computer systems			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Demonstrate understanding of software engineering principles and concepts through practical implementation.
CO 2	Apply software engineering lifecycle models and methodologies to develop and maintain software systems.
CO 3	Design software development processes that align with technical understanding and meet specified requirements.
CO 4	Analyze software requirements using appropriate modeling techniques and tools.
CO 5	Generate test case specifications and implement test cases based on given software requirements.

Ex. No	Experiment Title	Mapped CO/COs
1	Student Result Management System	
2	Library management system	
3	Inventory control system	
4	Accounting system	
5	Fast food billing system	
6	Bank loan system	
7	Blood bank system	
8	Railway reservation system	
9	Automatic teller machine	
10	Video library management system	

11	Hotel management system	
12	Hostel management system	
13	E-ticking	
14	Share online trading	
15	Hostel management system	
	Complete the following tasks for any five mentioned topics from the above list.	
1	Write the complete problem statement	CO1
2	Write the software requirement specification document	CO1, CO3
3	Draw the entity relationship diagram	CO2, CO4
4	Draw the data flow diagrams at level 0 and level 1	CO2, CO4
5	Draw use case diagram	CO2, CO4
6	Draw activity diagram of all use cases.	CO2, CO3
7	Draw state chart diagram of all use cases	CO2. CO3
8	Draw sequence diagram of all use cases	CO2, CO3
9	Draw collaboration diagram of all use cases	CO2, CO3
10	Assign objects in sequence diagram to classes and make class diagram	CO2, CO3
11	Create test cases for the testing of the modules	CO1, CO5

Semester II

Introduction to R Programming

Department:	**Department of Computer Science and Engineering**		
Course Name: Introduction to R Programming	**Course Code**	**L-T-P**	**Credits**
	ENSP112	3-0-1	4
Type of Course:	Minor		
Pre-requisite(s), if any:			
Brief Syllabus: Introduction to R: basics of R programming, including installing R and RStudio, understanding the R environment, and working with R packages. Data Types and Data Structures: different data types in R, such as numeric, character, logical, and factors. vectors, matrices, arrays, lists, and data frames. Data Manipulation: import, export, and manipulate data in R. data cleaning, sub setting, merging, transforming, and reshaping. Data Visualization: packages in R for creating visualizations, such as ggplot2 and lattice. Plots like charts, histograms, scatter plots, and other graphical representations to analyse and present data. Statistical Analysis: statistical analyses using R. descriptive statistics, hypothesis testing, t-tests, analysis of variance (ANOVA), regression analysis, and chi-square tests.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Getting Started with R and R Workspace**		**No. of hours: 4**
Content Summary: Introducing R, R as a programming Language, the need of R, Installing R, RStudio, RStudio's user interface, console, editor, environment pane, history pane, file pane, plots pane, package pane, help and viewer pane, R Workspace, R's working directory, R Project in R Studio, absolute and relative path, Inspecting an Environment, Inspect existing Symbols, View the structure of object, Removing symbols, Modifying Global Options, Modifying warning level, Library of Packages, Getting to know a package, Installing a Package from CRAN, Updating Package from			

CRAN, Installing package from online repository, Package Function, Masking and name conflicts.

Unit Number: 2	**Title: Basic Objects and Basic Expressions**	**No. of hours: 8**

Content Summary:

Vectors, Numeric Vectors, Logical Vectors, Character Vectors, subset vectors, Named Vectors, extracting element, converting vector, Arithmetic operators, create Matrix, Naming row and columns, subsetting matrix, matrix operators, creating and subsetting an Array, Creating a List, extracting element from list, subsetting a list, setting value, creating a value of data frame, subsetting a data frame, setting values, factors, useful functions of a data frame, loading and writing data on disk, creating a function, calling a function, dynamic typing, generalizing a function. Assignment Operators, Conditional Expression, using if as expression and statement, using if with vectors, vectorized if: ifelse, using switch, using for loop, nested for loop, while loop.

Unit Number: 3	**Title: Working with Basic Objects and Strings**	**No. of hours: 8**

Content Summary:

Working with object function, getting data dimensions, reshaping data structures, iterating over one dimension, logical operators, logical functions, dealing with missing values, logical coercion, math function, number rounding functions, trigonometric functions, hyperbolic functions, extreme functions, finding roots, derivatives and integration, Statistical function, sampling from a vector, Working with random distributions, computing summary statistics, covariance and correlation matrix, printing string, concatenating string, transforming text, Formatting text, formatting date and time, formatting date and time to string, finding string pattern, using group to extract data, reading data.

Unit Number: 4	**Title: Working with Data**	**No. of hours: 8**

Content Summary:

Visualize and Analyze Data: Reading and Writing Data, importing data using built-in-function, READR package, export a data frame to file, reading and writing Excel worksheets, reading and writing native data files, loading built-in data sets, create scatter plot, bar chart, pie chart, histogram and density plots, box plot, fitting linear model and regression tree.

Unit Number: 5	Title: Statistics, Advanced Charts	No. of hours: 10

Content Summary:

Statistics Co-relation, Bi- Variate Regression, T test, Paired T test, Anova, proportion, chi-square Test, statistics for bi-variate Association.

Advanced Charts: Bar charts for mean, Scatter plots for grouped data, Scatter plot matrics, 3 D Scatter plots, Charts for multiple variables

***Self-Learning Components:**

1) Database connectivity with R: https://www.geeksforgeeks.org/database-connectivity-with-r-programming/
2) Building Packages: https://bookdown.org/rdpeng/RProgDA/building-r-packages.html
3) Designing GUI: Building interactive application and connecting it with database.

- https://www.r-bloggers.com/2010/10/creating-guis-in-r-with-gwidgets/
- https://www.youtube.com/watch?v=C5R5SdYzQBI

4)Load the mtcars dataset (https://gist.github.com/seankross/a412dfbd88b3db70b74b)by using the code data(mtcars). Find the minimum, mean, median and maximum of the variable mpg in the mtcars dataset using just one line of code. We have not covered a function that does this yet, so the main point of this question is to get you used to using the resources you have available to find an answer. Describe the process you used (searched online? use the class textbook?) to find the answer.
5) https://github.com/ossu

Course Outcomes (CO)

COs	Statements
CO1	**Understand** the syntax and structure, principles and concepts of data manipulation and analysis, of the R programming language.

CO2	**Express** the process and results of statistical analyses and visualizations in a clear and concise manner.
CO3	**Determine** the suitable statistical tests and models for analyzing data in R.
CO4	**Identify** appropriate R packages and functions for specific data manipulation and analysis tasks.
CO5	**Articulate** the steps and logic behind data analysis and visualization tasks performed in R.
CO6	**Design** and create visually appealing and informative data visualizations using R packages.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 6. Knowledge 7. Understand 8. Apply 9. Analyze 10. Evaluate 11. Create	Affective levels(A) 5. Receiving 6. Responding 7. Valuing 8. Organizing 9. Characterizing	Psychomotor levels(P) 6. Imitation 7. Manipulation 8. Precision 9. Articulation 10. 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
CO1	2	2	1	3	2	2	1	1	2	2
CO2	2	3	2	2	3	2	2	3	2	2
CO3	2	3	2	2	3	2	2	3	2	-
		2	2	1	2	2	1	2	2	

CO4	-									2
CO5	2	3	2	2	3	2	2	3	2	2
CO6	2	3	-	2	3	2	1	3	2	2

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	**Getting Started with R and R Workspace**
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	**Basic Objects and Basic Expressions**
Local	Statistical Analysis of data is performed using R.
Regional	Statistical Analysis of data is performed using R.
National	Statistical Analysis of data is performed using R.
Global	Aligns with global trends in internet technologies and network protocols
Employability	ecognizing the relevance of R programming skills for enhancing employability in data analysis and statistical roles.
Entrepreneurship	Exploring the potential of R programming for entrepreneurial ventures in data analysis and statistical consulting.
Skill Development	Developing proficiency in R programming language and associated tools for effective data analysis and statistical modeling.
Professional Ethics	Promoting ethical practices in data analysis and statistical modeling using R, including data privacy, confidentiality, and responsible data handling.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	Working with Basic Objects and Strings
Local	-

Regional	-
National	-
Global	-
Employability	Employability in data analysis and statistics-related roles.
Entrepreneurship	-
Skill Development	The content summary covers various skills such as data manipulation, statistical analysis, working with random distributions, and text processing, which contribute to skill development in data analysis and programming.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Working with Data ,Statistics and advanced charts**
Local	he statistical techniques and charts covered in the syllabus can be applied to local data sets or specific local contexts for analysis and decision-making.
Regional	The statistical analysis methods can be used to study regional trends and patterns, and the charts can visually represent regional data.
National	Can be utilized to analyze national-level data and draw conclusions about various aspects, such as employment trends or gender disparities.
Global	Aligns with global trends in internet telephony, multimedia applications, and SEO can be applied to global data sets, enabling the study of global trends and patterns.
Employability	Develops skills in internet telephony, multimedia applications, and SEO he syllabus covers statistical analysis

	skills that are relevant for employability, as data analysis is increasingly important in many industries and job roles.
Entrepreneurship	The statistical techniques and charts can be applied to analyze data related to entrepreneurship, such as market trends, consumer behavior, or financial performance.
Skill Development	focuses on developing statistical analysis skills, including correlation, regression, and hypothesis testing, which contribute to overall skill development.
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

Introduction to R Programing Lab

Department:	**Department of Computer Science and Engineering**		
Course Name: Introduction to R Programing Lab	**Course Code**	**L-T-P**	**Credits**
	ENSP164	0-0-2	1
Type of Course:	Major		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO1	Understand the syntax and structure, principles and concepts of data manipulation and analysis, of the R programming language.
CO2	Express the process and results of statistical analyses and visualizations in a clear and concise manner.
CO3	Determine the suitable statistical tests and models for analyzing data in R.
CO4	Identify appropriate R packages and functions for specific data manipulation and analysis tasks.
CO5	Articulate the steps and logic behind data analysis and visualization tasks performed in R.
CO6	Design and create visually appealing and informative data visualizations using R packages.

List of Experiments

Ex. No	Experiment Title	Mapped CO/COs
1	Download and install R-Programming environment and install basic packages using install.packages() command in R	CO1
2	Learn all the basics of R-Programming (Data types, Variables, Operators etc,.)	CO1
3	Write a program to find list of even numbers from 1 to n using R-Loops.	CO1, CO5
4	Create a function to print squares of numbers in sequence.	CO1, CO5
5	Write a program to join columns and rows in a data frame using cbind() and rbind() in R.	CO1, CO4
6	Implement different String Manipulation functions in R.	CO1, CO4
7	Implement different data structures in R (Vectors, Lists, Data Frames)	CO1, CO4
8	Write a program to read a csv file and analyze the data in the file in R.	CO1, CO4
9	Create a data set and do statistical analysis on the data using R.	CO1,CO3, CO4
10	Create an example vector and use the cut() function on it. Explain your results.	CO1, CO4,CO5
11	Look up the functions arrange() and relocate(). Input the variable phisp from cacounty in each function. What are the functions doing?	CO1, CO4
12	Write an R script to do the following: a) simulate a sample of 100 random data points from a normal distribution with mean 100 and standard deviation 5 and store the result in a vector. b) visualize the vector created above using different plots. c) test the hypothesis that the mean equals 100. d) use wilcox test to test the hypothesis that mean equals 90.	CO1-CO5

13	Using the Algae data set from package DMwR to complete the following tasks. a) create a graph that you find adequate to show the distribution of the values of algae a6. b) show the distribution of the values of size 3. c) check visually if oPO4 follows a normal distribution. d) produce a graph that allows you to understand how the values of NO3 are distributed across the sizes of river. e) using a graph check if the distribution of algae a1 varies with the speed of the river. f) visualize the relationship between the frequencies of algae a1 and a6. Give the appropriate graph title, x-axis and y-axis title.	CO1, CO2, CO4, CO6
14	Read the file Coweeta.CSV and write an R script to do the following: a) count the number of observations per species. b) take a subset of the data including only those species with at least 10 observations. c) make a scatter plot of biomass versus height, with the symbol colour varying by species, and use filled squares for the symbols. Also add a title to the plot, in italics. d) log-transform biomass, and redraw the plot.	CO1, CO2, CO4, CO6
15	The built-in data set mammals contain data on body weight versus brain weight. Write R commands to: a) Find the Pearson and Spearman correlation coefficients. Are they similar? b) Plot the data using the plot command . c) Plot the logarithm (log) of each variable and see if that makes a difference.	CO1, CO2, CO4, CO6

16	In the library MASS is a dataset UScereal which contains information about popular breakfast cereals. Attach the data set and use different kinds of plots to investigate the following relationships: a) relationship between manufacturer and shelf b) relationship between fat and vitamins c) relationship between fat and shelf d) relationship between carbohydrates and sugars e) relationship between fibre and manufacturer f) relationship between sodium and sugars	CO1, CO2, CO5, CO6
17	Write R script to: a) Do two simulations of a binomial number with n = 100 and p = .5. Do you get the same results each time? What is different? What is similar? b) Do a simulation of the normal two times. Once with n = 10, μ = 10 and σ = 10, the other with n = 10, μ = 100 and σ = 100. How are they different? How are they similar? Are both approximately normal?	CO1, CO2, CO5
18	Mini Project Create a database medicines that contains the details about medicines such as {manufacturer, composition, price}. Create an interactive application using which the user can find an alternative to a given medicine with the same composition.	CO1, CO4, CO5
19	Mini Project Create a database songs that contains the fields {song_name, mood, online_link_play_song}. Create an application where the mood of the user is given as input and the list of songs corresponding to that mood appears as the output. The user can listen to any song form the list via the online link given.	CO1, CO4, CO5
20	Project Choose a dataset of interest, such as a public dataset from government or research sources, and perform exploratory data analysis using R. Generate descriptive statistics, create visualizations (e.g., plots, charts, maps), and derive meaningful insights from the data.Choose a dataset of interest, such as a	CO1, CO4, CO5

	public dataset from government or research sources, and perform exploratory data analysis using R. Generate descriptive statistics, create visualizations (e.g., plots, charts, maps), and derive meaningful insights from the data.	
21	Project Implement a machine learning algorithm using R and apply it to a relevant problem. You can explore supervised learning techniques like classification or regression, unsupervised learning techniques like clustering or dimensionality reduction, or even delve into natural language processing or image recognition tasks.	CO1, CO4, CO5

Discrete Mathematics

Department:	**Department of Computer Science and Engineering**		
Course Name: Discrete Mathematics	**Course Code**	**L-T-P**	**Credits**
	ENBC102	3-1-0	4
Type of Course:	Major		
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.			
UNIT WISE DETAILS			
Unit Number: 1	**Propositional Logics & Relations**	**No. of hours: 8**	
Content Summary: **Mathematical Logic:** Propositional and Predicate Logic, Propositional Equivalences, 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, Equivalence Relations, Partially Ordering.			
Unit Number: 2	**Title: Counting, Mathematical Induction and Discrete Probability**	**No. of hours: 8**	
Content Summary: Basics of Counting, Pigeonhole Principle, Permutations and Combinations, Inclusion-Exclusion Principle, Mathematical Induction, Probability, Bayes' Theorem.			
Unit Number: 3	**Title: Group Theory**	**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.			
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 Colouring, Euler's formulae.

***Self-Learning Components:**

1. Time-table Scheduling using Graph Colouring
2. Network Analysis, Routing & Optimization, using graph theory.
3. Combinatorial Optimization & Error Detection & correction using The Pigeonhole Principle.
4. Scheduling and Task Prioritization, using Partial orderings
5. Rules based system and Algorithm design using conditional statements.

Reference Books:

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.

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
CO1	3	2	2	-	2	-	2	-	-	-
CO2	1	2	-	-	3	-	1	-	-	-
CO3	-	-	-	-	3	-	2	1	-	3
CO4	-	-	-	-	3	-	2	-	-	3
CO5	-	-	-	-	3	-	2	-	-	3

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

CO-PSO Mapping

PO	PO1	PO2	PO3	PSO4
CO1	3	2	2	1
CO2	2	3	2	1

CO3	2	3	2	1
CO4	2	3	2	1
CO5	2	3	2	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 Development	Graph theory provides a powerful framework for representing 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

Basics of Operating System

Department:	**Department of Computer Science and Engineering**		
Course Name: Basics of Operating System	**Course Code**	**L-T-P**	**Credits**
	ENBC104	3-1-0	4
Type of Course:	Major		
Pre-requisite(s), if any:			

Brief Syllabus:
Operating systems course is intended as a general introduced 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 OS**	**No. of hours: 6**

Content Summary:
Introduction: Concept of Operating Systems, Generations of Operating systems, Types of Operating Systems, OS Services, System Calls, Layered System, Kernel, Types of Kernels (Monolithic/Macro Kernel and Micro Kernel), Virtual Machine.

Unit Number: 2	**Title: Processes and Threads**	**No. of hours: 12**

Content Summary:

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: Basic Concept, Type of Scheduling (Preemptive Scheduling, Non-preemptive Scheduling), Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time.

Unit Number: 3	**Title: Memory Management**	**No. of hours: 12**

Content Summary:
Memory Management: Address Binding, Dynamic Loading and Linking Concepts, Logical and Physical Addresses, Contiguous Allocation, Fragmentation, Paging, Segmentation, Combined Systems, Virtual Memory, Demand Paging, Page fault,

Page replacement algorithms, Global Vs Local Allocation, Thrashing, Working Set Model.		
Unit Number: 4	**Title: Process-Synchronization & Deadlocks**	**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, Message Passing, Classical IPC Problems: Reader's & Writer Problem, Dinning Philosopher Problem etc. Definition of Deadlocks, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.		

Define Course Outcomes (CO)

COs	**Statements**
CO1	Recall and comprehend the fundamental concepts of operating systems.
CO2	Analyze and evaluate the components and mechanisms related to processes and threads in operating systems.
CO3	Compare and contrast different process scheduling algorithms and their impact on system performance.
CO4	Apply memory management techniques and understand virtual memory concepts in operating systems.
CO5	Evaluate process synchronization mechanisms and understand the causes and prevention of deadlocks.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels©	Affective levels(A)	Psychomotor levels(P)
	1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving

CO1	C2	A1	P1
CO2	C4	A2	P3
CO3	C5	A3	P3
CO4	C3	A2	P2
CO5	C5	A4	P4

CO-PO Mapping

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

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	**Introduction to OS**
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	**Processes and Threads**
Local	Local industries and organizations that rely on computing systems will benefit from employees with knowledge of these concepts.
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 to 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	**Memory Management**

Local	Local educational institutions can benefit from teaching this course to provide students with a strong understanding of these fundamental concepts.
Regional	-
National	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	**Process-Synchronization & Deadlocks**
Local	Can benefit from teaching this course to build a strong foundation in computer science.
Regional	To meet the demand for skilled professionals in the region.
National	Can contribute to the national workforce and address the challenges of concurrent programming.
Global	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.

Basics of Operating Systems Lab

Department:	**Department of Computer Science and Engineering**		
Course Name: Basics of Operating Systems Lab	**Course Code**	**L-T-P**	**Credits**
	ENBC152	**0-0-2**	**1**
Type of Course:	**Major**		
Pre-requisite(s), if any:			

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 Python programs using the I/O system calls of UNIX/Linux operating system (open, read, write, close, fcntl, seek, stat, opendir, readdir).	CO1
11	Write a Python program to simulate the MVT (Multiple Variable Tasks) memory management technique.	CO5
12	Write a Python program to simulate the MFT (Multiple Fixed Tasks) memory management technique.	CO5
13	Write a Python program to simulate the Banker's Algorithm for Deadlock Avoidance and Prevention.	CO5
14	Write a Python program to implement the Producer-Consumer problem using semaphores using UNIX/Linux system calls.	CO3
15	Write Python programs to illustrate the following IPC (Inter-Process Communication) mechanisms: a) Pipes	CO3
16	Write Python 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 Fo5ile 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 techniques Single level directory	CO5

Concepts of Object Oriented Programming using C++

Department:	**Department of Computer Science and Engineering**		
Course Name: Concepts of Object Oriented Programming using C++	**Course Code**	**L-T-P**	**Credits**
	ENBC106	4-0-0	4
Type of Course:	Major		
Pre-requisite(s), if any: Basics of C programming			

Brief Syllabus:
The objective of this course is to introduce object-oriented programming. To explore and implement the vari ous features of OOP such as inheritance, polymorphism, Exceptional handling using programming language C++. After completing this course student can easily identify the basic difference between the programming approaches like procedural and object oriented.

UNIT WISE DETAILS

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

Content Summary:
Procedure Oriented and Object-Oriented Approach. Basic Concepts: Objects, classes, Principals like Abstraction, Encapsulation, Inheritance and Polymorphism. Dynamic Binding, Message Passing. Characteristics of Object-Oriented Languages, Functions, Returning values from functions, Data Types

Unit Number: 2	**Title:CLASSES AND OBJECTS**	**No. of hours: 10**

Content Summary:
Abstract data types, Object & classes, attributes, methods, C++ class declaration, Local Class and Global Class, State identity and behaviour of an object, Local Object and Global Object, Scope resolution operator, Friend Functions, Inline functions, Constructors and destructors, instantiation of objects, Types of Constructors, Static Class Data, Array of Objects, Constant member functions and Objects, Memory management Operators.

Unit Number: 3	**Title: INHERITANCE & POLYMORPHISM**	**No. of hours: 10**

Content Summary:
Inheritance, Types of Inheritance, access modes – public, private & protected, Abstract Classes, Ambiguity resolution using scope resolution operator and Virtual

base class, Aggregation, composition vs classification hierarchies, Overriding inheritance methods, Constructors in derived classes, Nesting of Classes Polymorphism, Type of Polymorphism – Compile time and runtime, Function Overloading, Operator Overloading (Unary and Binary) Polymorphism by parameter, Pointer to objects, this pointer, Virtual Functions, pure virtual functions.		
Unit Number: 4	**Title: STRINGS AND EXCEPTION HANDLING**	**No. of hours: 10**
Content Summary: Manipulating strings, String Manipulation Functions, formatted and Unformatted Input output. Exception handling, rethrowing exception, Exception Handling Techniques		
***Self-Learning Components:** 1. Students should explore Platforms like LeetCode, HackerRank for C++		
Reference Books: 1. E. Balagurusamy ,"Object Oriented Programming with C++", Mc Graw Hill,6th Edition,2013. 2. Schildt Herbert, "C++: The Complete Reference", Wiley DreamTech, 2005.Parasons, "Object Oriented Programming with C++", BPB Publication, 1999. 3. Steven C. Lawlor, "The Art of Programming Computer Science with C++", Vikas Publication, 2002. 4. Yashwant Kanethkar, "Object Oriented Programming using C++", BPB, 2004		

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** object oriented programming concepts.
CO2	**Applying** the concepts of object-oriented paradigm (Classes, Objects, inheritance, polymorphism etc.) for designing solution of a given programming problem
CO3	**Developing** applications that can manipulate data stored in files
CO4	**Developing** applications by considering all possible scenarios thereby employing appropriate exception handling.

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

CO-PO Mapping

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

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

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	2	1
CO2	2	3	2	1
CO3	2	3	2	1

CO4	2	3	2	1

Relevance of the Syllabus to various indicators

Unit I	Introduction
Local	-
Regional	-
National	-
Global	Aligns with global trends in programming languages
Employability	Proficiency in procedure-oriented and object-oriented approaches is highly valued by employers in the software development industry.
Entrepreneurship	Entrepreneurs in the software industry can benefit greatly from understanding procedure-oriented and object-oriented approaches
Skill Development	Develops skills in Visual Code using C++
Professional Ethics	-
Gender	These concepts are equally applicable and accessible to individuals of all genders pursuing careers in software development.
Human Values	-
Environment & Sustainability	-
Unit II	CLASSES AND OBJECTS
Local	Understanding abstract data types, object and classes, and other concepts covered in the course can be valuable at the local level for developing software solutions tailored to meet the specific needs of local communities or organizations.
Regional	The concepts covered, such as abstract data types and classes, can have regional relevance by enabling the development of software systems that address the requirements and preferences of a specific region or geographic area.
National	It contributes to the development of software infrastructure, applications, and systems that support national industries, governance, and public services.
Global	These concepts are widely used across countries and industries, facilitating collaboration and interoperability on a global scale.
Employability	These skills are highly sought after by employers, as they form the foundation for building robust and scalable software solutions.

Entrepreneurship	These concepts enable them to design innovative and scalable software products, laying the groundwork for successful ventures.
Skill Development	These skills include critical thinking, problem-solving, and designing modular and reusable software components.
Professional Ethics	promotes ethical practices by emphasizing proper design and encapsulation principles, which lead to well-structured and maintainable code.
Gender	Both men and women can equally benefit from and contribute to the field of software development, utilizing these concepts.
Human Values	promotes human values by fostering efficient and user-friendly software development practices, which can contribute to providing value to users and stakeholders.
Environment & Sustainability	promoting efficient and optimized software design, it indirectly contributes to reducing energy consumption and supporting environmental sustainability efforts.
Unit III	**INHERITANCE & POLYMORPHISM**
Local	Understanding abstract data types, object and classes, and other concepts covered in the course can be valuable at the local level for developing software solutions tailored to meet the specific needs of local communities or organizations.
Regional	The concepts covered, such as abstract data types and classes, can have regional relevance by enabling the development of software systems that address the requirements and preferences of a specific region or geographic area.
National	It contributes to the development of software infrastructure, applications, and systems that support national industries, governance, and public services.
Global	These concepts are widely used across countries and industries, facilitating collaboration and interoperability on a global scale.
Employability	These skills are highly sought after by employers, as they form the foundation for building robust and scalable software solutions.
Entrepreneurship	
Skill Development	These skills include critical thinking, problem-solving, and designing modular and reusable software components.
Professional Ethics	promotes ethical practices by emphasizing proper design and encapsulation principles, which lead to well-structured and maintainable code.
Gender	Both men and women can equally benefit from and contribute to the field of software development, utilizing these concepts.
Human Values	promotes human values by fostering efficient and user-friendly software development practices, which can contribute to providing value to users and stakeholders.

Environment & Sustainability	promoting efficient and optimized software design, it indirectly contributes to reducing energy consumption and supporting environmental sustainability efforts.
Unit IV	**STRINGS, FILES AND EXCEPTION HANDLING**
Local	
Regional	
National	
Global	
Employability	These skills are highly sought after by employers, as they form the foundation for building robust and scalable software solutions.
Entrepreneurship	
Skill Development	These skills include critical thinking, problem-solving, and designing modular and reusable software components.
Professional Ethics	promotes ethical practices by emphasizing proper design and encapsulation principles, which lead to well-structured and maintainable code.
Gender	Both men and women can equally benefit from and contribute to the field of software development, utilizing these concepts.
Human Values	promotes human values by fostering efficient and user-friendly software development practices, which can contribute to providing value to users and stakeholders.
Environment & Sustainability	promoting efficient and optimized software design, it indirectly contributes to reducing energy consumption and supporting environmental sustainability efforts.
SDG	SDG 9
NEP 2020	-
POE/4th IR	Aligns with the concepts of Design, Efficiency ,Problem Solving, Abstraction and System Analysis

Concepts of Object Oriented Programming Using C++ Lab

Department:	**Department of Computer Science and Engineering**		
Course Name: Concepts of Object Oriented Programming Using C++ Lab	**Course Code**	**L-T-P**	**Credits**
	ENBC154	**0-0-2**	**1**
Type of Course:	**Major**		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Demonstrate class object concepts by using C++.
CO 2	Develop programs using inheritance and polymorphism.
CO 3	Demonstrate the significance of constructors and destructor.
CO 4	Construct generic classes using template concepts.
CO5	Implement the concept of file handling.

Ex. No	Experiment Title	Mapped CO/COs
1	Write a program for Functions with default arguments	CO1
2	Simple Classes for understanding objects, member functions and Constructors .Classes with primitive data members	CO1
3	Write a program for Classes with constant data members, Classes with static member functions	CO1
4	Write a program for Classes with pointers as data members – String Class	CO1
5	Write a program for Classes with arrays as data members	CO1
6	Implementation of Call by Value, Call by Address and Call by Reference	CO1

7	Write a Program to illustrate New and Delete Keywords for dynamic memory allocation	CO1
8	Write a Program Containing a Possible Exception. Use a Try Block to Throw it and a Catch Block to Handle it Properly.	CO1
9	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	CO2,CO3
10	Write a Program to Demonstrate the Catching of All Exceptions.	CO1
11	Write a program fir passing object as argument to a function with help of a program to add marks of two students in two different subjects respectively. Marks of first student in "sub1" should be added with marks of second student in "sub1" and respectively for marks of "sub2" added for both students and then displayed.	CO2,CO3
12	Write a program to illustrate the concept of one class with two objects by taking student data.	CO3
13	Write a program to show the relationship of class and object to display roll no., grade and fee paid by student.	CO2,CO3
14	Write a program to define the member function outside and inside the class.	CO2,CO4
15	Write a program to read and display the information of N persons to illustrate the concept of array of objects.	CO2
16	Write a program to add two numbers to illustrate the use of friend function.	CO2
17	Write a program to assign and copy values to illustrate the concept of parametrized and copy constructor.	CO2,CO4
18	Write a program to show the order of constructor and destructor.	CO2
19	Write a program to add two numbers using binary operator overloading.	CO2,CO3
20	Write a program to illustrate the assignment operator overloading.	CO5
21	Sample Programs using inheritance in and accessing objects of different derived classes (a) Write a program to compute the marks explaining the concept of multiple inheritance.	CO3,CO4
22	Write a program to find the factorial of a number using inheritance	CO2,CO3
23	Sample Programs using polymorphism and virtual functions (using pointers) (a) Write a program to find the volume of cylinder and cuboid using function overloading.	CO5,CO4

	(b) Write a program to reverse a string using pointers.	
24	Write a program to explain the relationship of inheritance and virtual function.	CO4,
25	Project2: Create Tic Tac Toe game using C++ concepts	CO4
26	Project 3: Quiz Game: Design a quiz game program where users can answer multiple-choice questions from various topics. The program should keep track of the score and provide feedback on the user's performance.	CO4,CO5

Semester III

Introduction to Data Structures

Department:	**Department of Computer Science and Engineering**		
Course Name: Introduction to Data Structures	**Course Code**	**L-T-P**	**Credits**
	ENBC201	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, Evaluation of postfix expression using stacks.

Queues: ADT Queue and its operation, Array based implementation of linear Queues, Circular Queues, Priority queues

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 singly linked list: creation, insertion, deletion, search and display (based on the different position as specified by the user).Linked representation of Stacks & Queues.

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, traversing binary trees, Searching, Insertion and Deletion in binary search trees.

Graphs: Terminology and Representations, 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, insertion, selection, bubble, quick, merge, heap sort, algorithms and their analysis

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	**Evaluate** the efficiency of different data structures in terms of time and space complexity.
CO2	**Implement** a given Search problem (Linear Search and Binary Search).
CO3	**Demonstrate** an understanding of how data structures are implemented and their logical organization.
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© 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
CO1	3	3	3	-	-	2	-	1	-	-
CO2	3	3	2	-	-	2	-	-	-	-
CO3	3	3	3	-	-	3	-	-	-	-
CO4	3	3	3	-	-	3	-	-	-	-

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

CO-PSO Mapping

PSO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	3	-
CO2	3	3	3	-
CO3	3	-	2	-
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	-
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 9
NEP 2020	-
POE/4th IR	Aligns with the concepts of Design, Efficiency ,Problem Solving, Abstraction and System Analysis

Introduction to Data Structures Lab

Department:	**Department of Computer Science and Engineering**		
Course Name: Introduction to Data Structures Lab	**Course Code**	**L-T-P**	**Credits**
	ENBC253	**0-0-2**	**1**
Type of Course:	**Major**		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Equip the students with knowledge of algorithms, different types of data structures and analysis of space and time complexity of the algorithms
CO 2	Implementation of stack and queues and their related operations and provide an overview of dynamic Data Structure like linked list and its related operations.
CO 3	Equip the students with knowledge of algorithms and operations related to tree and graph data structures and their practical applications
CO 4	Implementing and analysing searching and sorting algorithms

List of Experiments

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	
9	Implement the linear data structure : Queue by performing Insertion and Deletion operation	CO2
10	Implement Circular Queue by performing Insertion and Deletion operation	
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
26	**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.	
27	**Mini Project 2:** Implement a maze solver using data structures like stacks or queues. The program should take an 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.	
28	**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.	

Fundamentals of Machine Learning

Department:	**Department of Computer Science and Engineering**		
Course Name: Fundamentals of Machine Learning	**Course Code**	**L-T-P**	**Credits**
	ENSP205	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			
Brief Syllabus: The objective of this course is to provide a rigorous training on the fundamental concepts, algorithms, and theories in artificial intelligence and machine learning. The lectures will cover various supervised and unsupervised learning algorithms that will help students to design and develop AI systems to solve real-world problems. Help student understand what machine learning is. How business can use machine learning in different domains to gain competitive advantage. Student is able to differentiate between different learning algorithms. To understand different data science processes, tools and techniques. Gain a fundamental understanding of the concepts and techniques that underpin machine learning algorithms			
Unit Number: 1	**Title: Introduction to Machine Learning**	**No. of hours: 8**	
Content Summary: Learning systems, real world applications of machine learning, why machine learning, variable types and terminology, function approximation, **Types of machine learning**: Supervised learning, unsupervised learning, Reinforcement learning **Linear Regression**: Weights and Features, Applications, Cost Functions, Finding best fit line, Gradient Descent Algorithm: Learning Algorithm, First order derivatives, Linear regression using gradient descent, Learning rate, Logistic Regression, Sigmoid Function, Cost Function for Logistic Regression, Multi-class classification, Probability Distribution, SoftMax Function, Polynomial Regression. **Performance Metrics:** Classification (Confusion Matrix, Accuracy, Precision, Recall, F1-score, ROC-AUC), Regression (MSE, MAE, RMSE, R2 Score).			
Unit Number: 2	**Title: Supervised Learning**	**No. of hours: 14**	
Content Summary: Decision Tree, Selecting Best Splitting Attribute, CART (Gini Index). ID3 (Entropy, Information Gain), Hyperparameters in Decision tree, Issues in Decision tree learning. Overfitting and Underfitting, Bias and Variance, Cross			

Validation. Bootstrap and Aggregation, Random Forest. Feature Engineering, Feature Selection, Feature Extraction.
Artificial Neural Network, Neural network representation, Perceptron model, Stepwise v/s Sigmoid function, Multilayer perceptron model, Matrix Calculus (Jacobian, Hessian Matrix), Computation Graph, Backpropagation Algorithm, Activation Functions, Stochastic Gradient Descent, Batch Gradient Descent, Overfitting Problem, Regularization (Ridge, Lasso, Elastic).
Bayesian Learning: Bayes theorem and concept learning, Naïve Bayes classifier, Gibbs Algorithm, Support Vector Machines, Hyperplane, Support Vectors, Kernels, K-nearest neighbour

Unit Number: 3	**Title: Unsupervised Learning**	**No. of hours: 08**

Content Summary:
Unsupervised learning Unsupervised learning (clustering, Association rule learning, Dimensionality reduction), Common distance Measures, k-means clustering, Elbow method, Hierarchical Clustering – agglomerative and divisive, Dendogram, Similarity measures for hierarchical clustering, DBSCAN, Cluster Quality (R index, Silhouette Coefficient), Dimensionality Reduction, Principal Component Analysis, T-distributed Stochastic Neighbour Embedding

Unit Number: 4	**Title: Evolutionary Algorithms and Deep Learning**	**No. of hours: 10**

Content Summary:
Population Based Algorithms: Genetic Algorithm, Fitness Function, Selection, Crossover, Mutation, Swarm Optimization, Particle Swarm Optimization, Ant-Colony Optimization, Reinforcement Learning, Actors, State, Reward Policy, Actions, Convolutional Neural Networks, Deep Learning for Sequential Data, Recurrent Neural Network, LSTM

Note: -It is expected that the students will design, analyse, and implement the programs that learn from experience.

***Self-Learning Components: -**
The students are expected to work on a project based on supervised learning/unsupervised learning/Evolutionary algorithms and present the progress at the end of the semester.

Reference Books:

1. T. M. Mitchell, Machine Learning (1 ed.), McGraw Hill, 2017. ISBN 978-1259096952.
2. E. Alpaydin, Introduction to Machine Learning (4 ed.), Phi, 2020. ISBN 978-8120350786

Define Course Outcomes (CO)

COs	Statements
CO1	Define the fundamental concepts and principles of artificial intelligence (AI) and machine learning (ML).
CO2	Apply supervised learning algorithms such as linear regression, logistic regression, decision trees, and support vector machines.
CO3	Analyze the strengths and limitations of different ML algorithms and approaches.
CO4	Design and develop ML pipelines for solving real-world problems.

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	A3	-
CO2	C3	A2	P2
CO3	C4	A4	-
CO4	C5	A5	P5

CO-PO Mapping

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

CO4	-	-	3	3	-	-	-	2	-	3

Justification for mapping

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	**Designing a learning system, Regression and Performance Metrics**
Local	-
Regional	-
National	-
Global	Machine learning has a global impact as it underpins advancements in various domains, including healthcare, climate modeling, financial markets, and social media. The ability to design learning systems and apply machine learning algorithms contributes to global innovation, collaboration, and the development of intelligent systems that benefit people worldwide.
Employability	Acquiring knowledge and skills in machine learning enhances employability prospects in the rapidly growing field of data science. Proficiency in designing learning systems, implementing machine learning algorithms, and understanding performance metrics makes individuals valuable assets to organizations seeking data-driven solutions, leading to increased employability opportunities.
Entrepreneurship	Machine learning offers opportunities for entrepreneurship by enabling the development of innovative products and services.
Skill Development	Studying machine learning develops essential skills such as problem-solving, data analysis, programming, and critical

	thinking. These skills are valuable in various professional domains beyond machine learning itself and contribute to overall skill development in areas such as data science, artificial intelligence, and decision-making.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Decision Tree and Artificial Neural Network**
Local	-
Regional	-
National	-
Global	Decision trees and machine learning algorithms are extensively used in diverse fields such as finance, marketing, climate modeling, and social sciences.
Employability	Acquiring knowledge and skills in decision trees, machine learning, and related algorithms enhances employability prospects in fields such as data analysis, artificial intelligence, and data science.
Entrepreneurship	Entrepreneurs can leverage their understanding of decision trees and machine learning algorithms to develop innovative products and services.
Skill Development	Studying decision trees, ensemble learning, neural networks, and other machine learning techniques develops critical thinking, problem-solving, and data analysis skills.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Unsupervised learning**
Local	-
Regional	-
National	-
Global	In the global context, unsupervised learning techniques contribute to the analysis and understanding of large-scale global datasets. Global industries, research organizations, and international collaborations can leverage clustering and dimensionality reduction to identify global trends, patterns, and similarities.

Employability	Proficiency in unsupervised learning, clustering, and dimensionality reduction enhances employability prospects in data analysis, machine learning, and data science roles.
Entrepreneurship	Entrepreneurs can leverage unsupervised learning techniques to identify market segments, customer preferences, and emerging trends.
Skill Development	The study of unsupervised learning, clustering, and dimensionality reduction develops critical thinking, problem-solving, and data analysis skills.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Population Based Algorithms**
Local	-
Regional	-
National	-
Global	The application of population-based algorithms extends to global challenges, such as climate modeling, disaster management, and global supply chain optimization.
Employability	Proficiency in population-based algorithms enhances employability prospects, particularly in fields related to optimization, data science, and artificial intelligence.
Entrepreneurship	Entrepreneurs can leverage population-based algorithms to develop innovative solutions and optimization-driven businesses. By understanding these algorithms, entrepreneurs can identify opportunities for optimization in various domains, create products or services based on population-based algorithms, and contribute to entrepreneurship in the digital era.
Skill Development	Studying population-based algorithms develops critical thinking, problem-solving, and algorithmic design skills.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 9, 11
NEP 2020	Integration of Emerging Technologies, Skill Development and Multidisciplinary Approach
POE/4th IR	Advanced Optimization, Data-driven Systems and Technological Disruption

Fundamentals of Machine Learning Lab

Department:	**Department of Computer Science and Engineering**		
Course Name: Fundamentals of Machine Learning Lab	**Course Code**	**L-T-P**	**Credits**
	ENSP257	**0-0-2**	**1**
Type of Course:	Minor		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Explain the use of Machine Learning Models in business and understand machine learning models can be used to solve business problems.
CO 2	Compare machine learning algorithms such as supervised, unsupervised, and reinforcement learning models.
CO 3	Identify the performance of different machine learning models and compare them to optimize the results.
CO 4	Make use continuous and discrete data set to fit regression and classification models.

List of Programs

Ex No	Experiment Title	Mapped CO/COs
1	Prediction using simple linear regression	CO1
2	Prediction using multiple linear regression	CO1
3	Classification using Logistics regression	CO1
4	Classification using linear discriminant analysis	CO1
5	Classification using support vector machine.	CO2
6	Classification using Guassian Naïve Bayes	CO2
7	Classification using decision Tree	CO2
8	Classification using Random Forest.	CO1
9	Classification using K nearest neighbour.	CO4
10	Write a program to Retrieve Data for a machine Learning project.	CO3

11	Write a program to Conduct Exploratory Data Analysis using Python	CO3
12	Write a program to Clean the Data using Python	CO4
13	Write a program for Data Modeling using Python	CO4
14	Write a program to implement multiple linear regression.	CO2
15	Write a program to scale the data and implement linear regression using sklearn.	CO2
16	Write a program to implement multiple logistic regression.	CO2
17	Write a program for graphical representation of data.	CO1
18	Write a program to implement genetic algorithms.	CO4
19	Write a program to implement CNN.	CO3
20	Write a program to implement LSTM.	CO3

Basics of Probability And Statistics

Department:	**Department of Computer Science and Engineering**		
Course Name: Basics of Probability And Statistics	**Course Code**	**L-T-P**	**Credits**
	ENBC203	4-0-0	4
Type of Course:	Major		
Pre-requisite(s), if any: Basics of Probability and Statistics			

Brief Syllabus:

The Probability and Statistics course is designed to provide students with a strong foundation in the principles and applications of probability and statistics in the context of data science. The course will cover various topics, including probability functions, random variables, discrete and continuous distributions, correlation and regression analysis, central limit theorem, and modeling uncertainty. Students will also explore real-world examples and utilize programming languages for statistical analysis and data visualization.

UNIT WISE DETAILS

Unit Number: 1	**Title: Basic Probability**	**No. of hours: 8**

Content Summary:

Definition of probability, conditional probability, independent events, Bayes' theorem, Bernoulli trials, Random variables, discrete random variable, probability mass function, continuous random variable, probability density function, cumulative distribution function, properties of cumulative distribution function, Two dimensional random variables and their distribution functions, Marginal probability function, Independent random variables.

Unit Number: 2	**Title: Probability Distributions for Data Science**	**No. of hours: 8**

Content Summary:

Binomial distribution, Poisson distribution, Poisson approximation to the binomial distribution, Normal distribution and its properties, Exponential distribution, Gamma distribution, Evaluation of statistical parameters for these distributions.

Unit Number: 3	**Title: Descriptive Statistics for Data Science**	**No. of hours: 8**

Content Summary:

Measures of central tendency: mean, median, mode, Measures of dispersion: variance, standard deviation, range, Skewness and kurtosis, Moments, Expectation, Linear correlation and correlation coefficient, Rank correlation coefficient.

Unit Number: 4	Title: Statistical Inference for Data Science	No. of hours: 8

Content Summary:
Hypothesis formation and testing, large sample tests for proportions, means, and standard deviations, Small sample tests: t-test, F-test, chi-square test, Test of significance for correlation coefficients, Goodness of fit tests, Independence of attributes tests.

Unit Number: 5	Title: Curve Fitting and Regression for Data Science	No. of hours: 8

Content Summary:
Curve fitting using the method of least squares, Fitting straight lines, parabolas, and general curves, Correlation analysis: coefficient of correlation, rank correlation, Simple linear regression: regression coefficients, lines of regression, Multiple linear regression: coefficient of multiple correlation, multiple regression equations.

***Self-Learning Components:**

- **Probability Simulation**: practice probability concepts through simulations using tools like Python's NumPy library or R programming language.
  https://pll.harvard.edu/course/data-science-probability
  https://www.mygreatlearning.com/academy/learn-for-free/courses/probability-for-data-science
  https://www.udemy.com/course/statistics-probability-for-data-science/
- **Data Analysis using R**: Students can learn and apply statistical techniques using R, an open-source statistical programming language, to analyze real-world datasets.
  https://www.coursera.org/learn/data-analysis-r
  https://www.udemy.com/course/data-analysis-with-r/
- **Hypothesis Testing with Excel**: Students can learn how to perform hypothesis testing using Excel's built-in statistical functions and conduct statistical analyses on data sets.
  https://www.coursera.org/learn/hypothesis-testing-python-excel
- **Introduction to Data Visualization**: Students can explore data visualization techniques and tools such as Tableau or matplotlib to effectively present statistical findings and insights.
  udemy.com/course/introduction-to-data-visualization/
- **Introduction to Machine Learning**: Students can gain an understanding of basic machine learning algorithms and their applications in data analysis and prediction, using tools like scikit-learn or TensorFlow.
  https://www.coursera.org/learn/machine-learning-duke
  https://onlinecourses.nptel.ac.in/noc22_cs29/preview

Define Course Outcomes (CO)

COs	Statements
CO1	**Demonstrate** understanding of various probability distributions and their applications in data science.
CO2	**Apply** statistical techniques and probability distributions to analyze and interpret data in data science applications.
CO3	**Utilize** statistical measures and methods to summarize and interpret data in data science projects.
CO4	**Evaluate** statistical inference techniques and apply them to make data-driven decisions in data science projects.
CO5	**Develop** statistical modeling and analysis techniques to solve data science problems.

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	-	-	-
C02	-	-	-
CO3	-	A4	-
CO4	C4	A4	P4
CO5	C5	A5	P5

CO-PO Mapping

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

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	Basic Probability
Local	Addresses local understanding probability of events
Regional	-
National	Contributes to national digital literacy (probability concepts are fundamental to understanding data and making informed decisions in the digital realm)
Global	Aligns with global trends in probability concepts apply universally in analyzing and predicting outcomes.
Employability	-
Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-

Unit II	Probability Distributions for Data Science
Local	Addresses local understanding probability distributions can be applied to analyze and model various online phenomena.
Regional	-
National	Contributes to national digital literacy probability distributions that play a role in understanding and analyzing data in the digital landscape.
Global	Aligns with global trends probability distributions which are applicable in analyzing data worldwide.
Employability	-
Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	Descriptive Statistics for Data Science
Local	-
Regional	-
National	Contributes to national network security strategies and protocols (understanding statistical measures helps in analyzing and evaluating network security).
Global	Aligns with global trends in network security techniques and protocols (statistical analysis is essential in assessing and improving network security worldwide).
Employability	-
Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	Statistical Inference for Data Science
Local	-
Regional	-
National	-
Global	Aligns with global trends in network security techniques and protocols as statistical analysis is essential in assessing and improving network security worldwide.
Employability	-

Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit V	Curve Fitting and Regression for Data Science
Local	Addresses local understanding and implementation of statistics which is applied in analyzing and optimizing internet-based services.
Regional	-
National	Contributes to national statistics aids in analyzing and enhancing digital communication.
Global	Aligns with global trends in applied statistics which is relevant in analyzing and improving global digital services).
Employability	Develops skills in knowledge of applied statistics supports data analysis and optimization in these areas.
Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4 (Quality Education)
NEP 2020	-
POE/4th IR	Aligns with the concepts of internet telephony, multimedia applications, and SEO (the syllabus content covers relevant topics in these areas).

Introduction to Java Programming

Department:	**Department of Computer Science and Engineering**		
Course Name: Introduction to Java Programming	**Course Code**	**L-T-P**	**Credits**
	ENBC251	3-1-0	4
Type of Course:	Major		
Pre-requisite(s), if any: C Programming			
Frequency of offering (check one): Odd semester			
Brief Syllabus: The objective is to impart programming skills used in this object-oriented language java. The course explores all the basic concepts of core java programming like object, classes, data types, features, operators, control structures, interfaces, packages, applets, AWT, Swings. The students are expected to learn it enough so that they can develop the basic applications as well as web solutions like creating applets etc.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction to Java**	**No. of hours: 12**	
Content Summary: Concepts of OOP, Features of Java, How Java is different from C++, Environmental setup, Basic syntax, Objects and classes, Basic Data Types, Variable Types, Modifier Types, Basic operators, Loop Control, Decision Making, Strings and Arrays, Methods, I/O. Introducing classes, objects and methods: defining a class, adding variables and methods, creating objects, constructors.			
Unit Number: 2	**Title: Arrays and Strings**	**No. of hours: 8**	
Content Summary: Classes: String and String Buffer classes, Wrapper classes: Basics types, using super, Multilevel hierarchy, abstract and final classes, Object class, Access protection, Inheritance, Overriding, Polymorphism, Abstraction, Encapsulation, Interfaces, Packages, Exploring java.util package.			

Unit Number: 3	Title: Exceptional Handling & Multithreading	No. of hours: 12
Content Summary: Exception Hierarchy, Exception Methods, Catching Exceptions, Multiple catch Clauses, Uncaught Exceptions Java's Built-in Exception. Creating, Implementing and Extending thread, thread priorities, synchronization suspending, resuming and stopping Threads, Multi- threading.		
Unit Number: 4	**Title: Input/output Programming & File handling**	**No. of hours: 8**
Basics Streams, Byte and Character Stream, predefined streams, Reading and writing from console and files. Reading data from files using input streams, Writing data to files using output streams.		
***Self-Learning Components:** 1. Students should explore Platforms like LeetCode, HackerRank for JAVA 2. Students should explore IDE like eclipse, Netbeans etc. 3. Students should demonstrate their learning through PowerPoint presentations		
Reference Books: 1. Herbert Schildt, —Java – The Complete Reference‖, Oracle Press. 2. Cay S. Horstmann, —Core Java Volume – I Fundamentals‖, Pearson.		

Define Course Outcomes (CO)

COs	Statements
CO1	Recognize features of object-oriented design such as encapsulation, polymorphism inheritance and composition of systems based on object identity.
CO2	Articulate re-usable programming components using Abstract Class, Interfaces and other permitted ways in packages.
CO3	Apply access control mechanism to safeguard the data and functions that can be applied by the object.

CO4	Design GUI applications using pre-built frameworks available in Java.

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

CO-PO Mapping

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

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	Introduction to Java
Local	-
Regional	-
National	-
Global	Java is widely used worldwide, and the skills gained from the course have global relevance in software development.
Employability	Understanding object-oriented programming and Java is valuable in the job market, as many industries and organizations rely on Java for software development.
Entrepreneurship	contribute to entrepreneurship by providing individuals with the skills and knowledge needed to develop software applications or start a technology-related business
Skill Development	contributes to skill development, particularly in programming, object-oriented design, and Java development
Professional Ethics	encourages ethical programming practices, such as writing clean code, following best practices, and respecting intellectual property rights
Gender	These concepts are equally applicable and accessible to individuals of all genders pursuing careers in software development.
Human Values	promotes human values such as teamwork, collaboration, and effective communication, which are essential in the software development industry.
Environment & Sustainability	promoting efficient programming practices and emphasizing code optimization
Unit II	Arrays and Strings
Local	-
Regional	-
National	It contributes to the development of software infrastructure, applications, and systems that support national industries, governance, and public services.
Global	Java is widely used worldwide, and the skills gained from the course have global relevance in software development.
Employability	Understanding object-oriented programming and Java is valuable in the job market, as many industries and organizations rely on Java for software development.
Entrepreneurship	contribute to entrepreneurship by providing individuals with the skills and knowledge needed to develop software applications or start a technology-related business

Skill Development	contributes to skill development, particularly in programming, object-oriented design, and Java development
Professional Ethics	encourages ethical programming practices, such as writing clean code, following best practices, and respecting intellectual property rights
Gender	These concepts are equally applicable and accessible to individuals of all genders pursuing careers in software development.
Human Values	promotes human values such as teamwork, collaboration, and effective communication, which are essential in the software development industry.
Environment & Sustainability	promoting efficient programming practices and emphasizing code optimization
Unit III	**Exceptional Handling & Multithreading**
Local	-
Regional	-
National	It contributes to the development of software infrastructure, applications, and systems that support national industries, governance, and public services.
Global	Java is widely used worldwide, and the skills gained from the course have global relevance in software development.
Employability	Understanding object-oriented programming and Java is valuable in the job market, as many industries and organizations rely on Java for software development.
Entrepreneurship	contribute to entrepreneurship by providing individuals with the skills and knowledge needed to develop software applications or start a technology-related business
Skill Development	contributes to skill development, particularly in programming, object-oriented design, and Java development
Professional Ethics	encourages ethical programming practices, such as writing clean code, following best practices, and respecting intellectual property rights
Gender	These concepts are equally applicable and accessible to individuals of all genders pursuing careers in software development.
Human Values	promotes human values such as teamwork, collaboration, and effective communication, which are essential in the software development industry.
Environment & Sustainability	promoting efficient programming practices and emphasizing code optimization
Unit IV	**Input/output Programming & Event Handling**
Local	-
Regional	-
National	-

Global	Java is widely used worldwide, and the skills gained from the course have global relevance in software development.
Employability	Understanding object-oriented programming and Java is valuable in the job market, as many industries and organizations rely on Java for software development.
Entrepreneurship	contribute to entrepreneurship by providing individuals with the skills and knowledge needed to develop software applications or start a technology-related business
Skill Development	contributes to skill development, particularly in programming, object-oriented design, and Java development
Professional Ethics	encourages ethical programming practices, such as writing clean code, following best practices, and respecting intellectual property rights
Gender	These concepts are equally applicable and accessible to individuals of all genders pursuing careers in software development.
Human Values	promotes human values such as teamwork, collaboration, and effective communication, which are essential in the software development industry.
Environment & Sustainability	promoting efficient programming practices and emphasizing code optimization
SDG	SDG 9
NEP 2020	-
POE/4th IR	Aligns with the concepts of Design, Efficiency ,Problem Solving, Abstraction and System Analysis

Introduction to Java Programming Lab

Department:	**Department of Computer Science and Engineering**		
Course Name: Introduction to Java Programming Lab	**Course Code**	**L-T-P**	**Credits**
	ENBC251	**0-0-2**	**1**
Type of Course:	**Major**		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Apply the concepts learned of operators, if-else, loops and arrays to java based application development.
CO 2	Demonstrate the use of various types of inheritances, polymorphisms, class objects, inheritances, packages and other concepts to basic and complex java programming problems.
CO 3	Demonstrate graphical applications based on java applets, swings and event handling
CO 4	Apply knowledge of event handling and AWT controls to create some new dynamic graphical applications.

List of Experiments:

Ex No	Experiment Title	Mapped CO/COs
1	Sample Programs using Objects and classes, Variable Types, Modifier Types, operators, Loops Decision Making, Strings and Arrays, a. WAP to display "Hello, it's a first program in java". b. WAP to find sum of two integers taken as input from user at runtime. c. WAP to find sum of two float numbers taken as command line arguments d. WAP to find changed case of entered character. e. WAP to find maximum of 3 integer numbers taken as input from user at runtime.	CO1

2	Sample Programs using Inheritance, Overriding, Polymorphism, Interfaces, Packages a. WAP in java to illustrate the concept of interfaces. b. Write a program in java to showcase uses of super keyword	CO1
3	Sample Programs using exception handling and threads a. Write a program to demonstrate the use of nesting of try-catch block b. WAP in java to illustrate the concept of using multiple catch clauses to handle different types of exceptions. c. WAP in java to create a user defined Exception and throw it explicitly.	CO2
4	Sample Programs using event handling and AWT controls	CO1
5	Sample Programs using swings Write an applet which will display "HAPPY" and "DEEPAVALI" as: The word "HAPPY" will roll from top to bottom and "DEEPAVLI" from bottom to "top" . Both will run at the same speed and stop simultaneously at the center of the applet.	CO3
6	WAP in java to create a frame with various AWT controls (like choice, list, TextField and Buttons) and handle the events thrown by them.	CO3
7	WAP in java to create a frame with AWT controls (like label, push buttons, Checkbox, Checkbox Group) and handle various events generated by them.	CO4
8	WAP to create a package as MyPack having a class with three methods: max, fact and show. Use it in other folder with setting classpath and without setting class path.	CO2
9	WAP to create a frame and illustrate the concept of using an adapter class in place of interfaces for handling various mouse events generated over frame window.	CO3
10	Write a program to display "hello" in different color where user clicks left mouse button and "world" where right mouse button is clicked. Use black background.	CO2
11	a. Demonstrate thread using Thread class and Runnable interface b. Demonstrate various thread methods using a program	CO3
12	Write a java program to create an abstract class named Shape that contains two integers and an empty method named printArea(). Provide three classes named Rectangle, Triangle and Circle such that each one of the classes extends the class Shape. Each one of the classes contain only the method printArea( ) that prints the area of the given shape.	CO4

13	a. WAP to create class with "name" as String and "age" as integer data members. The class should have two methods to take input from user and display the data. b. WAP to find factorial of a number using class and object.	CO3
14	Write a java program that implements a multi-thread application that has three threads. First thread generates random integer every 1 second and if the value is even, second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of cube of the number.	CO4
15	Create an Frame with one single button with caption "Click". On clicking the button will open a new Frame with title "Factorial". The frame will have two three controls :TextField, Label and button. On clicking button calculate the factorial entered in TextField control.	CO4
16	Project 1: Simple Calculator: Build a basic calculator application that performs arithmetic operations like addition, subtraction, multiplication, and division. You can add a user interface using Java Swing or JavaFX for a more interactive experience.	CO4
17	Project 2: Tic-Tac-Toe Game: Implement the classic Tic-Tac-Toe game where two players take turns marking X or O on a 3x3 grid. Allow players to play against each other.	CO4
18	Project 3: Quiz Application: Design a quiz application that presents multiple-choice questions to users and keeps track of their scores. Include features like a timer, question randomization, and a scoring system.	CO4
19	Project 4: Hangman Game: Create a Hangman game where players guess letters to uncover a hidden word. Include features such as displaying the word's progress, tracking incorrect guesses, and providing hints.	CO4

Career Development Program-I

Department:	**Department of Computer Science and Engineering**		
Course Name: Career Development Program-I	**Course Code**	**L-T-P**	5. **Credits**
	AEC011	3-0-1	3
Type of Course:	Life Skills for Professionals - I		
Pre-requisite(s), if any:			
Frequency of offering (check one): Odd / Even			
Brief Syllabus: The basics of Quantitative Aptitude are starting from Simplification questions then Number system, Percentage, and Average. If you are just starting learning Quantitative aptitude, you must start with simplification and the number system. The main quantitative Aptitude topics are Number System, Simplification, Percentage, simple interest and compound interest, Profit and loss, Ratio and Proportion, Time and Work, Time, Speed, and Distance, Average, Probability, Permutation, and Combination, Equations and Equalities. Data interpretation questions can be solved by using all quantitative aptitude topics concepts. Life skills for professionals' course are designed for engineering learner to enhance and develop interpersonal skills that characterize a person's relationships with other professionals. This program will teach skills which will prepare them for a successful career in their industry. The main topics will include verbal communicational skills, non- verbal communication skills, Active listening skills, written communication skills and presentation skills.			

12. UNIT WISE DETAILS

Unit Number: 1	**Title: Introduction to number system**	**No. of hours: 8**
Content Summary: Number system, Divisibility, Unit digit, Last two-digit, Remainder, Number of zero, Factor, LCM & HCF, Simplification.		
Unit Number: 2	**Title: Ratio & its application**	**No. of hours: 8**
Content Summary: Ratio, Mixture, Average, Partnership.		

Unit Number: 3	Title: communication	No. of hours: 6
Content Summary: Introduction to Communication, Types of communication, Verbal & Nonverbal Communication, Barriers to Communication, Body language, Listening Skills.		
Unit Number: 4	**Title: Personality development**	**No. of hours: 8**
Content Summary: Paragraph writing, Professional Speaking (Elocutions, Debate, describing incidents and developing positive nonverbal communication. Articulation and pronunciation. Communicating with confidence. Using appropriate tone pitch and volume.		
***Self-Learning Components:** https://www.youtube.com/watch?v=0pNGYM0ltlw https://www.youtube.com/watch?v=0gUgm4zB2F4		
Reference Books: 1. Qunatitative Aptitude by R.S Agarawal 2. Quicker math by M.Tyra 3. Communication skills by G.H. Hook		

Define Course Outcomes (CO)

COs	Learner will develop self confidence in their communication abilities and enabling them to express themselves assertively.
CO 1	Learner will develop the ability for advanced critical thinking and the ability to formulate logical arguments.
CO 2	The learner to apply different value systems and moral dimensions while taking decisions.
CO 3	Apply mathematical techniques to quantitative theory.
CO 4	Soft skills include attributes and personality traits that help learner to interact with others and succeed.
CO 5	Learner will cultivate self-confidence, problem solving and critical thinking abilities

Semester IV

Fundamentals of Algorithm Design & Analysis

Department:	**Department of Computer Science and Engineering**		
Course Name: Fundamentals of Algorithm Design & Analysis	**Course Code**	**L-T-P**	**Credits**
	ENBC202	3-1-0	4
Type of Course:	**Programme Core**: 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 analyse 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, tree traversals), string matching, elements of computational geometry.			
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 behaviour, 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.

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 Colouring 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. String matching.

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	Apply algorithmic problem-solving strategies.
CO4	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	A3	P2
C02	C4	A4	P3
CO3	C3	A4	P4
CO4	C4	A5	P5

CO-PO Mapping

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

CO-PSO Mapping

PO	PSO1	PSO2	PSO3
CO1	2	1	3
CO2	2	1	2
CO3	3	1	3
CO4	3	1	2

Relevance of the Syllabus to various indicators

Unit I	**Introduction to algorithm**
Local	-
Regional	-

National	-
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	-
Regional	-
National	-
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	-
Regional	-
National	-
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	-
Regional	-
National	-
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	Integration of Critical Thinking, Computational Thinking and Skill Development
POE/4th IR	Automation and Efficiency and Data Analysis.

Fundamentals Analysis and Design of Algorithms Lab

Department:	**Department of Computer Science and Engineering**		
Course Name: Fundamentals Analysis and Design of Algorithms Lab	**Course Code**	**L-T-P**	**Credits**
	ENBS254	0-0-2	1
Type of Course:	Major		

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

Introduction to Database Management Systems

Department:	**Department of Computer Science and Engineering**		
Course Name: Introduction to Database Management Systems	**Course Code**	**L-T-P**	**Credits**
	ENBC204	3-1-0	4
Type of Course:	Major		
Pre-requisite(s), if any:			
Brief Syllabus: This course introduces the basic concept of database, Database modelling languages, E-R modelling and Transaction Processing.			
UNIT WISE DETAILS			
Unit Number: 1	**Title:** Introduction to database	**No. of hours:** 10	
Content Summary: Overview of DBMS, DBMS system vs file system, Data independence and abstraction level, Architecture of DBMS, Schemas, Instances and various DBMS models.			
Unit Number: 2	Title: Relational Query Languages	**No. of hours:** 10	
Content Summary: Data Modelling: Data modeling using Entity relationship Model: ER Model Concepts, notation of ER diagram, mapping constraints, Keys, concept of super key, candidate key, primary key, generalization and specialization Relational Modelling: Concepts, constraints, Language, Relational Database Design by ER and EER mapping, Relational Algebra, Relational Calculus, relational Algebra and its fundamental operations			
Unit Number: 3	Title: Database design and Transaction Processing	No. of hours: 10	

Content Summary: Database design: Functional Dependencies, lossless decomposition and Normalization (1NF, 2NF, 3NF, BCNF, 4NF) Transaction management: transaction concept, ACID properties, state of transaction, serializability, checkpoints and deadlock handling.		
Unit Number: 4	Title: Introduction to SQL	No. of hours: 10
Content Summary: Introduction to SQL: characteristics and advantages of SQL, SQL data types, SQL commands and operators, Tables, views and indexes, Queries and sub-queries, aggregate function, insert, alter and update operations		
***Self-Learning Components:** • **PostgreSQL** • **MongoDB** **Note:** Students will give presentations and submit projects based on self-learning components for evaluation.		
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
CO1	Analyze the key components and concepts of DBMS, including data independence, architecture, schemas and various DBMS models.
CO2	Apply data modeling techniques using ER model and understanding the concepts of keys
CO3	Evaluate the principles and techniques of relational modeling and the fundamental operations of relational algebra.
CO4	Design and implement effective database designs by analyzing functional dependencies and normalization.

CO5	Explain transaction processing, concurrency control and database recovery protocols in databases.

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	A4	P4
C02	C3	A2	P2
CO3	C5	A3	P3
CO4	C6	A4	P5
CO5	C2	A3	P4

CO-PO Mapping

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

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

CO-PSO Mapping

CO	PSO1	PSO2	PSO3	PSO4
CO1	3	2	2	2
CO2	2	2	2	2

CO3	3	2	2	2
CO4	2	2	2	3
CO5	2	2	2	3

Relevance of the Syllabus to various indicators

Unit I	Introduction to Database
Local	The syllabus on DBMS is relevant locally as it provides foundational knowledge and skills in managing databases, which are essential for businesses, organizations, and institutions at the local level.
Regional	The syllabus is also relevant regionally as the principles and concepts of DBMS are applicable and implemented in various industries and sectors within the region, contributing to the efficient management and utilization of data.
National	The syllabus holds national relevance as DBMS is a fundamental aspect of information management and technology infrastructure across the country. It helps in maintaining data integrity, supporting decision-making processes, and ensuring efficient data storage and retrieval.
Global	In today's interconnected world, the knowledge and skills related to DBMS have global significance. The concepts and principles covered in the syllabus align with international standards and practices in managing databases, making it relevant in a global context.
Employability	The syllabus on DBMS enhances employability as it equips students with the necessary knowledge and skills sought after by employers. Proficiency in DBMS is in demand across various industries, including IT, finance, healthcare, e-commerce, and more.
Entrepreneurship	Understanding DBMS is valuable for aspiring entrepreneurs. It enables them to design and implement effective database systems for their ventures, ensuring efficient data management and supporting business operations.
Skill Development	The syllabus contributes to skill development by providing theoretical knowledge and practical skills in DBMS. Students develop skills in data modeling, database design, SQL programming, and transaction management, which are transferable to various domains.
Professional Ethics	The syllabus indirectly addresses professional ethics by emphasizing data integrity, security, and privacy aspects of DBMS. Students learn about ethical considerations in handling sensitive data and ensuring responsible use of database systems.
Gender	

Human Values	The syllabus indirectly promotes human values by fostering responsible and ethical use of data. It encourages students to consider the impact of their actions on individuals, society, and the broader human community.
Environment & Sustainability	-
Unit II	**Data Modelling and Languages**
Local	The syllabus on Data Modeling and Relational Modeling is relevant locally as it provides foundational knowledge and skills in database design and management, which are essential for businesses, organizations, and institutions at the local level
Regional	The syllabus is also relevant regionally as database design and management principles are applicable and implemented in various industries and sectors within the region, contributing to efficient data organization and retrieval.
National	The syllabus holds national relevance as database design and management are fundamental aspects of information management and technology infrastructure across the country. It supports data integrity, effective decision-making, and efficient data storage and retrieval at a national level.
Global	In today's interconnected world, the knowledge and skills related to data modeling and relational modeling have global significance. The concepts and techniques covered in the syllabus align with international standards and practices in database design and management, making it relevant in a global context.
Employability	The syllabus on Data Modeling and Relational Modeling enhances employability as it equips students with the necessary knowledge and skills sought after by employers. Proficiency in data modeling, ER diagrams, relational algebra, and database design is in high demand across various industries globally.
Entrepreneurship	Understanding data modeling and relational modeling is valuable for aspiring entrepreneurs. It enables them to design and implement effective database systems for their ventures, ensuring efficient data management and supporting business operations.
Skill Development	The syllabus contributes to skill development by providing theoretical knowledge and practical skills in data modeling, ER diagrams, relational algebra, and database design. Students develop skills in conceptualizing data structures, mapping relationships, and performing database operations, which are transferable and valuable in various domains.

Professional Ethics	The syllabus indirectly addresses professional ethics by emphasizing data integrity, privacy, and responsible data management practices. Students learn about ethical considerations in designing databases and handling sensitive information.
Gender	
Human Values	The syllabus indirectly promotes human values by fostering responsible data management practices. It encourages students to consider the impact of their data modeling decisions on individuals, society, and ethical considerations.
Environment & Sustainability	
Unit III	**Database design and Transaction Processing**
Local	The syllabus on Database Design and Transaction Management is relevant locally as it provides foundational knowledge and skills in designing efficient and reliable databases, which are essential for businesses, organizations, and institutions at the local level.
Regional	The syllabus is also relevant regionally as database design and transaction management principles are applicable and implemented in various industries and sectors within the region, contributing to effective data management and transaction processing.
National	The syllabus holds national relevance as efficient database design and transaction management are crucial for information management and technology infrastructure across the country. It supports data integrity, data consistency, and reliable transaction processing at a national level.
Global	In today's interconnected world, the knowledge and skills related to database design and transaction management have global significance. The concepts and techniques covered in the syllabus align with international standards and practices, making it relevant in a global context.
Employability	The syllabus on Database Design and Transaction Management enhances employability as it equips students with the necessary knowledge and skills sought after by employers. Proficiency in functional dependencies, normalization, ACID properties, and transaction management is in high demand across various industries globally.
Entrepreneurship	Understanding database design and transaction management is valuable for aspiring entrepreneurs. It enables them to design and implement efficient and

	scalable database systems for their ventures, ensuring reliable data management and transaction processing.
Skill Development	The syllabus contributes to skill development by providing theoretical knowledge and practical skills in database design, functional dependencies, normalization, and transaction management. Students develop skills in identifying functional dependencies, normalizing databases, and ensuring data consistency and reliability through transaction management.
Professional Ethics	The syllabus indirectly addresses professional ethics by emphasizing data integrity, data consistency, and responsible data management practices. Students learn about the importance of maintaining the ACID properties in transactions and handling potential issues such as deadlocks.
Gender	
Human Values	The syllabus indirectly promotes human values by fostering responsible data management practices. It encourages students to consider the impact of their database design decisions on individuals, society, and ethical considerations related to data privacy and security.
Environment & Sustainability	
Unit IV	**Introduction to SQL**
Local	The syllabus on Introduction to SQL is relevant locally as it equips individuals with the necessary skills to interact with and manipulate local databases. SQL is widely used in various local industries and organizations for data management and analysis purposes.
Regional	The syllabus holds regional relevance as SQL is a widely adopted standard for database management across different regions. The ability to work with SQL databases is valuable in regional industries and sectors that rely on efficient data storage and retrieval.
National	SQL is extensively used in national databases and information systems, making the syllabus highly relevant at a national level. The knowledge and skills gained from the syllabus enable individuals to work with national-scale databases and contribute to data-driven decision-making processes.
Global	SQL is a globally recognized and standardized language for database management. The syllabus aligns with international SQL standards and practices, enabling individuals to work with databases on a global scale and collaborate across borders.

Employability	Proficiency in SQL is highly sought after by employers worldwide. The syllabus enhances employability by providing individuals with the necessary skills to work with databases, execute SQL commands, perform data analysis, and contribute to effective data management
Entrepreneurship	Knowledge of SQL is valuable for entrepreneurs as it allows them to design and manage their own databases, extract meaningful insights from data, and make informed business decisions. The syllabus fosters entrepreneurial skills by enabling individuals to leverage SQL for their ventures.
Skill Development	The syllabus contributes to skill development by covering various aspects of SQL, including data types, commands, operators, table management, queries, and data manipulation operations. Individuals develop practical skills in working with databases and retrieving and manipulating data using SQL.
Professional Ethics	The syllabus indirectly addresses professional ethics by emphasizing responsible data management practices. Students learn to handle data securely, respect privacy regulations, and adhere to ethical considerations when accessing and modifying databases.
Gender	
Human Values	The syllabus indirectly promotes human values by emphasizing responsible data management practices, privacy protection, and ethical considerations. Students learn to handle data with integrity, respect individual privacy rights, and consider the societal impact of data-driven decision-making.
Environment & Sustainability	
SDG	SDG4, SDG8, SDG9
NEP 2020	Competency-based approach to education: The syllabus topics provide a foundation of knowledge and skills necessary for understanding and working with relational databases. By covering concepts such as data abstraction, data independence, entity-relationship modeling, and database security, the syllabus supports the development of competencies related to database management and data governance.
POE/4th IR	The syllabus provides students with the foundational knowledge and skills necessary to navigate the data-driven landscape of the 4IR and apply entrepreneurial principles when managing and utilizing databases for business purposes.

Introduction to Database Management Systems Lab

Department:	**Department of Computer Science and Engineering**		
Course Name: Introduction to Database Management Systems Lab	**Course Code**	**L-T-P**	**Credits**
	ENBC252	**0-0-2**	**1**
Type of Course:	**Major**		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Define and apply mapping constraints to transform an ER model into a relational schema
CO 2	Demonstrate an understanding of keys (super key, candidate key, primary key) and their roles in database design
CO 3	Perform data manipulation operations such as insertion, deletion, and updating using SQL commands
CO 4	Create and manage database objects like tables, views, and indexes using SQL statements

List of Experiments

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, SSID) SUBJECT (Subcode, Title, Sem, Credits) IAMARKS (USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)	CO1, CO2
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, CO2
3	Consider the below Database:	CO3, CO4

	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 5. Find Coen's movies with McDormand	
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. x. Create a view which lists the daily sales date wise for the last one week	CO3, CO4
5	**Database Schema for a Student Library scenario** Student(**Stud_no : integer,** Stud_name: string) Membership(**Mem_no: integer, Stud_no: integer**) Book(**book_no: integer**, book_name:string, author: string) Iss_rec**(iss_no:integer**, iss_date: date, **Mem_no: integer, book_no: 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 student names with their membership numbers iv. List all the issues for the current date with student and Book names	CO3, CO4

	v. List the details of students who borrowed book whose author is CJDATE vi. Give a count of how many books have been bought by each student vii. Give a list of books taken by student with stud_no as 5 viii. List the book details which are issued as of today ix. Create a view which lists out the iss_no, iss _date, stud_name, book name x. Create a view which lists the daily issues-date wise for the last one week	
6	**Database Schema for a Employee-pay scenario** employee(**emp_id : integer**, emp_name: string) department(**dept_id: integer**, dept_name:string) paydetails(**emp_id : integer**, **dept_id: integer**, basic: integer, deductions: integer, additions: integer, DOJ: date) payroll**(emp_id : integer**, pay_date: date) 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 the employee details department wise iv. List all the employee names who joined after particular date v. List the details of employees whose basic salary is between 10,000 and 20,000 vi. Give a count of how many employees are working in each department vii. Give a names of the employees whose netsalary>10,000 viii. List the details for an employee_id=5 ix. Create a view which lists out the emp_name, department, basic, deductions, netsalary x. Create a view which lists the emp_name and his netsalary	CO3, CO4
7	**Database Schema for a Video Library scenario** Customer(cust_no: integer,cust_name: string) Membership(**Mem_no: integer**, **cust_no: integer**) Cassette(**cass_no:integer**, cass_name:string, Language: String) Iss_rec**(iss_no: integer**, iss_date: date, **mem_no: integer**, **cass_no: 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 customer names with their membership numbers iv. List all the issues for the current date with the customer names and cassette names	CO3, CO4

<table>
<tr><td></td><td>v. List the details of the customer who has borrowed the cassette whose title is " The Legend"
vi. Give a count of how many cassettes have been borrowed by each customer
vii. Give a list of book which has been taken by the student with mem_no as 5
viii. List the cassettes issues for today
ix. Create a view which lists outs the iss_no, iss_date, cust_name, cass_name
x. Create a view which lists issues-date wise for the last one week</td><td></td></tr>
<tr><td>8</td><td>Database Schema for a student-Lab scenario
Student(stud_no: integer, stud_name: string, class: string)
Class(<u>class: string,</u> descrip: string)
Lab(<u>mach_no: integer</u>, Lab_no: integer, description: String)
Allotment(Stud_no: Integer, mach_no: integer, dayof week: string)
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 machine allotments with the student names, lab and machine numbers.
iv. List the total number of lab allotments day wise
v. Give a count of how many machines have been allocated to the 'CSIT' class
vi. Give a machine allotment details of the stud_no 5 with his personal and class details
vii. Count for how many machines have been allocated in Lab_no 1 for the day of the week as "Monday"
viii. How many students class wise have allocated machines in the labs
ix. Create a view which lists out the stud_no, stud_name, mach_no, lab_no, dayofweek
x. Create a view which lists the machine allotment details for "Thursday".</td><td>CO3, CO4</td></tr>
<tr><td>9</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');</td><td>CO3, CO4</td></tr>
</table>

> SAVEPOINT B;
> INSERT INTO class VALUES (7, 'Bravo');
> SAVEPOINT C
> SELECT * FROM class;
> ROLLBACK TO B;
> SELECT * FROM class;
> ROLLBACK TO A;

10 **Consider the following two tables: SHOP and ACCESSORIES** — CO3, CO4

Table: SHOP

ID	ShopName	Area
S01	ABC Computronics	CP
S02	All Infotech Media	GK II
S03	Tech Shoppe	CP
S04	Geek Tenco Soft	Nehru Place
S05	Hitech Tech Store	Nehru Place

Table: ACCESSORIES

No	Name	Price	Id
A01	Motherboard	12000	S01
A02	Hard Disk	5000	S01
A03	Keyboard	500	S02
A04	Mouse	300	S01
A05	Motherboard	13000	S02
A06	Keyboard	400	S03
A07	LCD	6000	S04
T08	LCD	5500	S05
T09	Mouse	350	S05
T10	Hard Disk	450	S03

i. Perform Cartesian product or Cross join of both tables.
ii. To display the Name and Price of all the Accessories in ascending order of their price.
iii. To display ID and ShopName of all shops located in Nehru Place.
iv. To display minimum and maximum price of all accessories.
v. To display Name, Price of all accessories and their respective ShopName where they are available.

11 **In continuation with experiment no. 10, find the output of the following SQL queries based on above mentioned tables:** — CO3, CO4

i. SELECT DISCTINCT NAME FROM ACCESSORIES WHERE PRICE >= 5000;
ii. SELECT AREA, COUNT(*) FROM SHOP GROUP BY AREA;
iii. SELECT COUNT(DISTINCT AREA) FROM SHOP;

iv. SELECT NAME, PRICE*0.05 DISCOUNT FROM ACCESSORIES WHERE ID IN ('S02', 'S03');

12 **Consider the following two tables: PRODUCT and CLIENT.** CO3, CO4

Table: Product

P_ID	ProdName	Manufacturer	Price	ExpiryDate
TP01	Talcom Powder	LAK	40	2011-06-26
FW05	Face Wash	ABC	45	2010-12-01
BS01	Bath Soap	ABC	55	2010-09-10
SH06	Shampoo	XYZ	120	2012-04-09
FW12	Face Wash	XYZ	95	2010-08-15

Note:

- P_ID is the primary key.

Table: Client

C_ID	ClientName	City	P_ID
1	Cosmetic Shop	Delhi	FW05
6	Total Health	Mumbai	BS01
12	Live Life	Delhi	SH06
15	Pretty One	Delhi	FW05
16	Dreams	Bengaluru	TP01
14	Expressions	Delhi	NULL

Note:

- C_ID is the primary key.
- P_ID is the foreign key referencing P_ID of Client Table.

i. To display the ClientName and City of all Mumbai and Delhi based clients in Client table.
ii. Increase the price of all the products in Product Table by 10%.
iii. To display the ProdName, Manufacturer, ExpiryDate of all the products that expired on or before '2010-12-31'.
iv. To display C_ID, ClientName, City of all the clients including the ones that have not purchased a product and their corresponding ProdName sold.
v. Display the distinct Manufacturer from Product table.
vi. Display the ClientName, C_ID who belong to a city starts with 'M'

13 Consider the following schema for a Library Database: BOOK(Book_id, Title, Publisher_Name, Pub_Year) BOOK_AUTHORS(Book_id, Author_Name) PUBLISHER(Name, Address, Phone) BOOK_COPIES(Book_id, Programme_id, No-of_Copies) BOOK_LENDING(Book_id, Programme_id, Card_No, CO3, CO4

	Date_Out, Due_Date) LIBRARY_PROGRAMME(Programme_id, Programme_Name, Address) Write SQL queries to 1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each Programme, etc. 2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017. 3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation. 4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query. 5. Create a view of all books and its number of copies that are currently available in the Library	
14	Consider the following schema for Order Database: SALESMAN(Salesman_id, Name, City, Commission) CUSTOMER(Customer_id, Cust_Name, City, Grade, Salesman_id) ORDERS(Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id) Write SQL queries to 1. Count the customers with grades above Bangalore's average. 2. Find the name and numbers of all salesman who had more than one customer. 3. List all the salesman and indicate those who have and do not have customers in their cities (Use UNION operation.) 4. Create a view that finds the salesman who has the customer with the highest order of a day. 5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.	CO3, CO4
15	Consider the schema for Movie Database: ACTOR(Act_id, Act_Name, Act_Gender) DIRECTOR(Dir_id, Dir_Name, Dir_Phone) MOVIES(Mov_id, Mov_Title, Mov_Year, Mov_Lang, Dir_id) MOVIE_CAST(Act_id, Mov_id, Role) RATING(Mov_id, Rev_Stars) Write SQL queries to 1. List the titles of all movies directed by 'Hitchcock'. 2. Find the movie names where one or more actors acted in two or more movies. 3. List all actors who acted in a movie before 2000 and in a movie after 2015 (use JOIN operation). 4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title. 5. Update rating of all movies directed by 'Steven Spielberg' to 5.	CO3, CO4
16	Consider the schema for College Database: STUDENT(USN, SName, Address, Phone, Gender) SEMSEC(SSID, Sem, Sec)	CO3, CO4

	CLASS(USN, SSID) COURSE(Subcode, Title, Sem, Credits) IAMARKS(USN, Subcode, SSID, Test1, Test2, Test3, FinalIA) Write SQL queries to 1. List all the student details studying in fourth semester 'C' section. 2. Compute the total number of male and female students in each semester and in each section. 3. Create a view of Test1 marks of student USN '1BI15CS101' in all Courses. 4. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students. 5. Categorize students based on the following criterion: If FinalIA = 17 to 20 then CAT = 'Outstanding' If FinalIA = 12 to 16 then CAT = 'Average' If FinalIA< 12 then CAT = 'Weak' Give these details only for 8th semester A, B, and C section students.	
17	Consider the schema for 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) Write SQL queries to 1. Make a list of all project numbers for projects that involve an employee whose last name is 'Scott', either as a worker or as a manager of the department that controls the project. 2. Show the resulting salaries if every employee working on the 'IoT' project is given a 10 percent raise. 3. Find the sum of the salaries of all employees of the 'Accounts' department, as well as the maximum salary, the minimum salary, and the average salary in this department. 4. Retrieve the name of each employee who works on all the projects controlled by department number 5 (use NOT EXISTS operator). 5. For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than Rs. 6,00,000.	CO3, CO4

Introduction to Computer Network

Department:	**Department of Computer Science and Engineering**		
Course Name: Introduction to Computer Network	**Course Code**	**L-T-P**	**Credits**
	ETBC206	3-1-0	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, Transmission Media, LAN: Wired LAN, Wireless LANs, Connecting LAN and Virtual LAN, Techniques for Bandwidth utilization: Multiplexing - Frequency division, Time division and Wave division, Concepts on spread spectrum.

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: 10**
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:** 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

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		
C02	C2		
CO3	C3		
CO4	C6		
CO5			

CO-PO Mapping

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

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

CO-PSO Mapping

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

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 Ethics	Ethical considerations are important when working with 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	-
POE/4th IR	

Introduction to Computer Networks Lab

Department:	**Department of Computer Science and Engineering**		
Course Name: Introduction to Computer Networks Lab	**Course Code**	**L-T-P**	**Credits**
	ENBC256	**0-0-2**	**1**
Type of Course:	**Major**		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	To gain hands-on experience working with network hardware, software, and tools.
CO 2	Network Configuration and Troubleshooting.
CO 3	Network Design and Implementation.
CO 4	To measure and evaluate network performance using tools and techniques.

List of Experiments

Ex. No	**Experiment Title**	**Mapped CO/COs**
1	Create a simple network with multiple PCs, switches, and routers.	CO1
2	Assign IP addresses to devices and configure basic connectivity.	CO2
3	Test connectivity between PCs using ping and trace routes.	CO3
4	Configure VLANs on switches and assign ports to specific VLANs.	CO4
5	Enable inter-VLAN routing using a router or Layer 3 switch.	CO2,CO3
6	Test connectivity between PCs in different VLANs.	CO3
7	Set up a network with multiple routers.	CO3,CO4
8	Configure static routes on routers to enable communication between networks.	CO4
9	Verify routing tables and test connectivity between networks.	CO3
10	Set up a network with a private IP address space.	CO3
11	Configure NAT on a router to enable translation between private and public IP addresses.	CO3,CO4

12	Test connectivity between devices on the private network and the Internet.	CO4
13	Create a wireless network using access points and wireless clients.	CO2
14	Simulate network issues such as connectivity problems, routing errors, or misconfigurations.	CO1
15	Design and implement a network traffic monitoring.	CO2
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.	CO3
17	Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes.	CO2
18	Write a program for error detection and correction for 7/8 bits ASCII codes using CRC.	CO3
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.	CO2
20	Design and deploy TCP based Multithreaded HTTP client server for accessing student activity data in the institute.	CO2,CO3
21	Design and deploy TCP based Multithreaded FTP client server to share institute level notices.	CO4
22	Design and deploy TCP based Multithreaded Chat client server for your class.	CO4
23	Design and deploy UDP based Multithreaded Chat client server for your class.	CO3
24	Examining real-world network deployments.	CO3
25	Case studies of network failures and their resolutions.	CO2

Career Development Program-II

Department:	**Department of Computer Science and Engineering**		
Course Name: Career Development Program-II	**Course Code**	**L-T-P**	**Credits**
	AEC012	3-0-0	3
Type of Course:	Ability Enhancement course		
Pre-requisite(s), if any:			
Frequency of offering (check one): Odd / Even			
Brief Syllabus: The basics of Quantitative Aptitude are starting from Simplification questions then Number system, Percentage, and Average. If you are just starting learning Quantitative aptitude, you must start with simplification and the number system. The main quantitative Aptitude topics are Number System, Simplification, Percentage, simple interest and compound interest, Profit and loss, Ratio and Proportion, Time and Work, Time, Speed, and Distance, Average, Probability, Permutation, and Combination, Equations and Equalities. Data interpretation questions can be solved by using all quantitative aptitude. Throughout the syllabus there will be opportunities for practical application interactive exercises and assessments to reinforce the concepts taught. The learners may be required to give presentation, engage in role plays, participate in group discussions, and complete written assessments to demonstrate their communication and skill development.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Percentage and its application**	**No. of hours: 8**	
Content Summary: Percentage, Profit and loss, Simple interest, Compound interest			
Unit Number: 2	**Title: Ratio & its application, Speed and number of arrangements**	**No. of hours: 8**	

Content Summary: Time & Work, Time & Distance, Train, Boat & Stream, Permutation & combination, Probability		
Unit Number: 3	**Title: log, BODMAS and mensuration**	**No. of hours: 8**
Inequalities, Log, progression, Mensuration		
Unit Number: 4	**Title: Personality development**	**No. of hours: 6**
Content Summary: Presentation Skills, Telephone etiquettes, LinkedIn Profile and professional networking, Video resumes & Mock interview sessions.		
Contents beyond Syllabus: https://www.youtube.com/watch?v=0pNGYM0ltlw https://www.youtube.com/watch?v=0gUgm4zB2F4		
Reference Books: Quantitative Aptitude by R.S Agarawal Quicker math by M.Tyra Business English, Pearson, 2008 Communication skills by G.H. Hook		

Course Outcomes (COs)	
COs	Learner will develop self confidence in their communication abilities and enabling them to express themselves assertively
CO 1	To develop the ability for advanced critical thinking and the ability to formulate logical arguments.
CO 2	To develop different value systems and moral dimensions taking decisions.
CO 3	Apply mathematical techniques to quantitative theory.
CO 4	To develop in them vital communication skills which should be integral to personal, social and professional interactions
CO 5	The learner will apply their business etiquette and communication in their core field.
COs	Learner will develop self confidence in their communication abilities and enabling them to express themselves assertively

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	C3		P2
CO3	C3		P3
CO4	C1		-
CO5	C1		P1

Competitive Programming 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:	Skill Enhancement Course (SEC)		
Pre-requisite(s), if any: None			
Brief Syllabus: Introduction to Competitive Coding, Data Structures and Algorithms, Time and Space Complexity Analysis, Problem Solving Techniques, Advanced Data Structures, Coding Paradigms, Online Judges and Contest Platforms, Tips and Tricks for Competitive Coding, Mock Contests and Practice Sessions, Self-Learning Components			

S.N	Experiment Index	COs
1	Introduction to Competitive Coding • Overview of competitive coding and its importance in the field of 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/ ▪ 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/	CO5
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
	Self-Learning Component:	CO5

10	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: 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/

Course Outcomes

CO1	Proficiency in Algorithms and Data Structures: Demonstrate proficiency in implementing and analyzing various algorithms and data structures commonly used in competitive programming.
CO2	Efficient Problem Solving: 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	Algorithmic Thinking: Cultivate algorithmic thinking and problem-solving skills by identifying patterns, applying appropriate algorithms, and selecting optimal data structures for a given problem.
CO4	Code Optimization and Complexity Analysis: 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	Competitive Programming Skills: 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 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

- 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.
- 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.
- Minimum spanning tree: Find the minimum spanning tree of a graph. This can be solved using Prim's algorithm or Kruskal's algorithm.
- Topological sorting: Find a topological ordering of a graph. This can be solved using Kahn's algorithm.
- Strongly connected components: Find the strongly connected components of a graph. This can be solved using Tarjan's algorithm.
- Bipartite matching: Find a maximum bipartite matching in a graph. This can be solved using the Hungarian algorithm.
- 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

Semester V

Computer Organization & Architecture

Department:	**Department of Computer Science and Engineering**		
Course Name: Computer Organization & Architecture	**Course Code**	**L-T-P**	**Credits**
	ENBC301	4-0-0	4
Type of Course:	Major (Core)		
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 analyzing 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:**
1. BSim Documentation

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/

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

Reference Books:
1. "Computer Organization & Architecture", Smruti Ranjan Sarangi, McGraw Hill
2. "Computer System Architecture", Mano M. Morris, Pearson.
3. "Computer Organization and Embedded Systems", 6th Edition by Carl Hamacher, McGraHill Higher Education
4. "Computer Architecture and Organization", 3rd Edition by John P. Hayes, WCB/McGraw-Hill
5. "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
CO1	3	-	-	-	-	-	-	-	-	-
CO2	-	3	2	-	-	-	-	-	-	-
CO3	-	-	-	3		-	-	-	-	-
CO4	-	-	3	-	-		-	-	-	-
CO5	2	-	-	-	-	-	-	-	-	-

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

CO-PSO Mapping

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

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.

Big Data Analytics with Spark ,Scala

Department:	**Department of Computer Science and Engineering**		
Course Name: Big Data Analytics with Spark + Scala	**Course Code**	**L-T-P**	**Credits**
	ENSP309	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			
Brief Syllabus: This syllabus covers the core concepts and techniques of Apache Spark and Scala in big data analytics. By the end of the course, you will have a solid foundation in using Spark and Scala to manipulate, analyze, and gain insights from large-scale datasets, enabling you to tackle real-world big data challenges.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction to Apache Spark**	**No. of hours: 4**	
Content Summary: Apache Spark and Installation: Introduction to Apache Spark, Features of Apache Spark, Apache Spark Stack, Introduction to RDD's, RDD's Transformation, What is good and bad in Map Reduce, Why to use Apache Spark.			
Unit Number: 2	**Title: Spark: A Hadoop Replacement**	**No. of hours: 8**	
Content Summary: Java, Scala or Python?, Scala, Packages, Data Types, Classes, Calling Functions, Operations, Control Structures.			
Unit Number: 3	**Title: Resilient Distributed Datasets (RDD) and SQL DataFrames**	**No. of hours: 8**	

Content Summary:
Introduction to RDD, RDD
Operations, Creating RDDs, Transformations, map, flatMap, filter, union, intersection, substract, distinct, sample, Actions, Working with key/value pair RDD, Data Shuffling, Spark SQL, SQL Tables and Views, unmanaged and managed tables, create SQL database and tables, create view, reading tables into DataFrame, DataFrame Reader, DataFrame Writer, Parquet, JSON, reading JSON file into DataFrame, reading CSV file, reading Avro, ORC, Image file,

Unit Number: 4	**Title: Spark Streaming**	**No. of hours: 8**

Content Summary:
Evolution of Apache Spark Stream Processing Engine, Micro-batch stream processing, DStreams, philosophy of structured streaming, programming model, Stream DataSource and sink, structured streaming application, streaming DataFrame Operations, joining two streaming DataFrames, working with socket Data Source, Rate Data Source, File Data Source, Kafka Data Source, Custom Data Source, Working with Data Sinks, Kafka Data Sinks, Foreach Data Sinks, Console Data Sinks, Memory Data Sinks, Output modes and Triggers.

***Self-Learning Components:**

- Big Data Analytics with Apache Spark by DataCamp https://www.datacamp.com/tutorial/apache-spark-tutorial-machine-learning
- Big Data Analysis with Scala and Spark by Coursera
- Apache Spark with Scala" by Udemy https://www.udemy.com/course/apache-spark-programming-in-scala/
- https://intellipaat.com/apache-spark-scala-training
- Apache Spark. EdX. https://www.edx.org/learn/apache-spark

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:

- Tom White "Hadoop: The Definitive Guide" Third Edit on, O"reily Media, 2012.
- Gerard Maas and Francois Garillot , "Stream Processing with Apache Spark: Mastering Structured Streaming and Spark Streaming", O'Reilly, 2019.
- "Spark: The Definitive Guide" by Bill Chambers and Matei Zaharia.

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** Gain a comprehensive understanding of Apache Spark and its ecosystem, including Spark Core, Spark SQL, Spark Streaming, and Spark MLlib. Understand the concepts and features of distributed computing and in-memory processing offered by Spark.
CO2	**Express** Clearly express the objectives, requirements, and challenges of big data analysis to stakeholders. Communicate the advantages and potential impact of utilizing Apache Spark and Scala for data analysis, emphasizing the scalability, performance, and versatility of the Spark framework.
CO3	**Determine** Assess the suitability of Apache Spark and Scala for the specific big data analysis task. Consider factors such as data volume, complexity, processing needs, and available computing resources to determine the appropriate Spark components and techniques to employ.
CO4	**Identify** Identify relevant datasets and variables to analyze within the big data using Apache Spark. Apply Spark's data manipulation, querying, and transformation capabilities to preprocess and clean the data, ensuring data quality and consistency.
CO5	**Articulate** Clearly articulate the insights, findings, and outcomes derived from the big data analysis using Apache Spark and Scala. Present the results in a meaningful and actionable manner.
CO6	**Design** Develop Spark applications and scripts that leverage Spark's APIs and libraries for distributed data analysis, machine learning, and streaming processing. Optimize the performance and resource utilization of Spark jobs.

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	C3		P2
CO3	C3		P3
CO4	C1		-

CO5	C1		P1

CO-PO Mapping

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

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

CO-PSO Mapping

PO	PO1	PO2	PO3	PSO4
CO1	3	2	1	1
CO2	2	3	1	1
CO3	3	2	1	2
CO4	2	2	1	1
CO5	2	3	1	1
CO6	1	2	1	2

Relevance of the Syllabus to various indicators

Unit I	**Introduction to Apache Spark**
Local	Understanding Apache Spark is locally relevant for organizations and businesses that use or plan to use Spark for data processing and analytics
Regional	
National	
Global	Spark is globally relevant, with its usage spanning multinational corporations, international research collaborations, and organizations dealing with diverse data sets.

Employability	Knowledge of Apache Spark and its features enhances a candidate's employability in big data analytics and data engineering roles, given the widespread adoption of Spark in the industry.
Entrepreneurship	
Skill Development	
Professional Ethics	
Gender	
Human Values	
Environment & Sustainability	
Unit II	**Spark: A Hadoop Replacement**
Local	Local businesses, organizations, and academic institutions may use one or more of these languages for software development and data analysis.
Regional	
National	
Global	All three languages have global relevance, being widely used in multinational corporations, international research collaborations, and open-source projects.
Employability	Proficiency in Java, Scala, or Python can enhance a candidate's employability, as these languages are in high demand for various software development and data-related job roles.
Entrepreneurship	
Skill Development	The syllabus covers essential programming concepts that are transferable across these languages, allowing students to develop programming skills that are valuable in diverse software projects.
Professional Ethics	
Gender	
Human Values	
Environment & Sustainability	
Unit III	**Resilient Distributed Datasets (RDD) and SQL DataFrames**
Local	Understanding Spark and its operations is locally relevant for organizations, businesses, and academic institutions
Regional	
National	
Global	Spark is globally relevant, with its usage spanning multinational corporations, international research collaborations, and open-source communities.
Employability	Proficiency in Apache Spark and its various operations can enhance a candidate's employability in big data analytics and

	data engineering roles, given the widespread adoption of Spark in the industry.
Entrepreneurship	
Skill Development	he syllabus covers essential technical skills related to Spark operations, Spark SQL, and working with different file formats, contributing to students' skill development in big data processing.
Professional Ethics	
Gender	
Human Values	
Environment & Sustainability	
Unit IV	**Spark Streaming**
Local	Understanding the evolution of Apache Spark in stream processing is locally relevant for organizations, businesses, and academic institutions that use Spark for real-time data processing and analytics
Regional	
National	
Global	Spark's stream processing evolution is globally relevant, with its usage spanning multinational corporations, international research collaborations, and open-source communities.
Employability	Proficiency in Apache Spark's stream processing features can enhance a candidate's employability in roles related to real-time data processing, data engineering, and stream analytics.
Entrepreneurship	
Skill Development	The syllabus covers essential technical skills related to stream processing with Spark, contributing to students' skill development in real-time data analytics.
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

Big Data Analytics Lab

Department:	**Department of Computer Science and Engineering**		
Course Name: Big Data Analytics Lab	**Course Code**	**L-T-P**	**Credits**
	ENSP359	**0-0-2**	**1**
Type of Course:	**Major**		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Understand the process of data collection, cleaning, and preprocessing to ensure data quality and suitability for analysis.
CO 2	Learn various big data analysis techniques and algorithms to extract meaningful insights from large datasets.
CO 3	Develop skills in visualizing and communicating analytical findings effectively using appropriate data visualization tools.
CO 4	Apply big data analytics techniques to real-world applications and address business challenges.

Ex. No	Experiment Title	Mapped CO/COs
1	Installing and configuring Apache Spark	
2	Installing and configuring the Scala IDE	CO2
3	Installing and configuring JDK	
4	Word Count: Perform a word count on a large text dataset using Spark and Scala.	CO2
5	File Management tasks in Hadoop: Create a directory in HDFS at given path(s). Log Analysis: Analyze server logs to extract useful information such as error rates, response times, and traffic patterns using Spark and Scala.	CO2, CO3
6	Create Spark RDD using parallelize with spark Context. Parallelize () method and using Spark shell	CO2

7	Write a scripts in Spark to Read all text files from a directory into a single RDD	CO3
8	Write a spark program to load a CSV file into Spark RDD using a Scala	CO1, CO2
9	Write a Spark Streaming program for adding 1 to the stream of integers in a reliable, fault tolerant manner, and then visualize them.	CO2, CO3
10	Web Scraping: Scrape data from websites using Spark and Scala, and perform analysis on the extracted data.	CO2, CO3
11	Time Series Analysis: Analyze time series data using Spark and Scala to identify patterns and trends.	
12	Anomaly Detection: Detect anomalies in large-scale datasets using Spark MLlib and Scala.	CO2, CO3
13	Network Traffic Analysis: Analyze network traffic data to detect anomalies and patterns using Spark and Scala.	
14	Develop a streaming application by- Connecting to a Stream, Preparing the Data in the Stream, Performing Operations on Streaming Dataset, creating a Query, Starting the Stream Processing and Exploring the data.	CO3
15	Create a Structured streaming job by Initializing Spark, acquiring streaming data from sources, declaring the operations we want to apply to the streaming data and outputting the resulting data using Sinks.	
16	Create a small but complete Internet of Things (IoT)-inspired streaming program.	CO3
17	Define the schema in Structured Streaming to handle the data at different levels.	
18	Develop any Spark Streaming application and do the following : a) Create a Spark Streaming Context, b) Define one or several DStreams from data sources or other DStreams c) Define one or more output operations to materialize the results of these	CO2, CO3
19	Movie Recommendation System: Build a movie recommendation system using collaborative filtering with Spark MLlib and Scala.	
20	E-commerce Recommendation System: Build a recommendation system for an e-commerce platform using collaborative filtering with Spark MLlib and Scala.	CO2, CO4

DSE- I

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:	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, VM Placement, VM Migration, VM Migration Demo, VM clustering, Design Issues in VM Clustering, Need of Dockers and Containers, Docker Eco-System, Hypervisor vs Docker.			
Unit Number: 2	**Title: Microservices**	**No. of hours: 10**	
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 VM Consolidation 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	Statements
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© 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
CO1	3	3	1	1				2	1	2
CO2	2	2	2	1			1	3		1
CO3	3	3	3		1	2		2	1	
CO4	3	3	2	1		2		2	2	

CO-PSO Mapping

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

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

List of Programs :

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: 12**	
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: 12**
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** Microsoft Azure For Dummie by Timothy L. Warner and Katrina Dow, and Michael Washam		

Define Course Outcomes (CO)

COs	Statements
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© 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
CO1	2	–	–	–	–	1	1	1	–	–
CO2	3	3	3	–	3	–	–	–	–	2
CO3	3	3	2	3	3	–	–	–	–	–
CO4	2	2	2	2	2	–	–	–	–	–
CO5	2	2	–	2	2	–	–	–	–	2

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4
CO1	2	–	–	–
CO2	3	3	3	–
CO3	3	3	–	–
CO4	2	2	–	2

CO5	2	2	2	2

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:	**Minor**		
Pre-requisite(s), if any:			

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:	CO3

	Explore user authentication methods, roles, and permissions.	
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:	Elective -I		
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) explore 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) Attend webinars or read case studies of organizations that have successfully implemented cloud storage and database solutions. Learn about their challenges, architectural decisions, and best practices they adopted.
3) Join online forums and communities focused on cloud computing and database management. Participate in discussions, ask questions, and share your knowledge. Platforms like Stack Overflow, Reddit, and specialized forums like the AWS Developer Forums or the Google Cloud Community can be valuable sources of information.

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.
CO6	**Design** data storage architecture for cloud-based applications.

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	C3		P2
CO3	C3		P3
CO4	C1		-

CO5	C1		P1

CO-PO Mapping

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

CO-PSO Mapping

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

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.

Storage and Databases on Cloud Lab

Department:	**Department of Computer Science and Engineering**		
Course Name: Storage and Databases on Cloud Lab	**Course Code**	**L-T-P**	**Credits**
	ENSP455	**0-0-2**	**1**
Type of Course:	**Minor**		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	Understand the principles and concepts of cloud storage and databases.
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 Blob Storage:** Practice creating buckets/containers, uploading files, setting access permissions, and managing storage resources.	CO1, CO4, 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 setting up user accounts, managing roles and permissions, and implementing encryption techniques to protect data at rest and in transit	CO1, CO5
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	**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	**Cloud Data Warehousing:** • Gain hands-on experience with cloud data warehousing platforms like Amazon Redshift	CO1, CO2, CO5, CO6

	• load data, run complex queries, and analyze performance optimization techniques for data warehousing.	

DES-II

Mobile Application Development using iOS

Department:	**Department of Computer Science and Engineering**		
Course Name: Mobile Application Development using iOS	**Course Code**	**L-T-P**	**Credits**
	ENSP409	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any: Basics of Android			

Brief Syllabus:
The objective of the course is to provide skills to develop applications for OS X and iOS. It includes introduction to development framework Xcode. Objective-C is used as programming language to develop the applications. Objective-C is the superset of the C programming language and provides object-oriented capabilities and a dynamic runtime. Objective-C inherits the syntax, primitive types, and flow control statements of C and adds syntax for defining classes and methods.

UNIT WISE DETAILS

Unit Number: 1	**Title: Introduction to IDE and SDK of iOS App Development**	**No. of hours: 10**

Content Summary:
Xcode-The SDK environment, Supporting tools, Advance settings. Development Technique, Fundamental of Object-Oriented Programming, The MVC architecture.

Unit Number: 2	**Title: Objective-C**	**No. of hours: 10**

Content Summary:
Introduction to Objective C, Primitive Data Types, Conditions, Loops, Functions, Arrays, Pointers, Structures, Classes, Objects, Foundation, Memory Management, Inheritance, Categories, Protocols, Predicates, Blocks, Multi-Threading.
Objects Send and Receive Messages concept, Use of Pointers to Keep Track of Objects, Methods - Return Values.

Unit Number: 3	**Title: Encapsulating Data**	**No. of hours: 10**

Content Summary:
Properties of Encapsulation of an Object's Values, Declare Public Properties for Exposed Data, Use Accessor Methods to Get or Set Property Values, Concept of Dot Syntax, Properties Are Backed by Instance Variables.
Dealing with Errors: Use NSError for Most Errors, Some Delegate Methods Alert You to Errors, Some Methods Pass Errors by Reference

Unit Number: 4	**Title: Developing iOS Applications**	**No. of hours: 10**

Content Summary:
iOS App Anatomy, Design Principles, Creating a Basic Hello World App with interface elements, UI View & Controller, UI Elements, Trigger Actions, Storyboard, Device Orientations, Using Gestures, Popovers and Modal Dialogs, Creating Universal Apps, Status Bar, Navigation Bar, Tab Bar, Content Views (e.g. Image view, Map View etc.), UI Table View and Table View Controller, Core Data, Test your App, Publishing your App.

***Self-Learning Components:**
1. XCode Documentation

References:
1. https://www.tutorialspoint.com/objective_c/objective_c_quick_guide.htm
2. https://www.coursera.org/learn/introduction-to-ios-mobile-application-development
3. https://www.geeksforgeeks.org/classes-objects-in-objective-c/

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

Text Book:

1. Effective objective C 2.0, Matt Galloway, Effective software development series, Scott Meyers.

Reference Books:

1. Programming in Objective-C (5th Edition) (Developer's Library) by Stephen G. Kochan.
2. iOS 6 Development Unleashed: Developing Mobile Applications for Apple iPhone, iPad, and iPod Touch by Robert McGovern

Online References:

1. https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html
2. https://www.digitalocean.com/community/tutorials/objective-c-hello-world-tutorial

Course Outcomes (CO)

COs	Statements
CO1	Create iPhone apps using Objective-C and Apple's new programming language, use industry tools and frameworks such as Cocoa, Xcode, UIKit, Git.
CO2	Understand and know how to use properly UIKit, asynchronous code, Core Image, NSURL Session and JSON Map Kit and Core Location, Auto Layout, Source Control, Core Data, Animation, and the app submission process.
CO3	Read and write programs based on Objective-C, also have a strong grasp of Objective-C objects
CO4	Organize their code professionally using objects and blocks, prototype several entry- level apps and try to publish on App store.

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
CO1	3	-	-	-	-	-	-	-	-	-
CO2	-	3	2	-	-	-	-	-	-	-
CO3	-	-	-	3		-	-	-	-	-
CO4	-	-	3	-	-		-	-	-	-

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	Introduction to IDE and SDK of iOS App Development
Local	-
Regional	-
National	-
Global	Xcode is the official IDE provided by Apple for iOS app development. It is available globally and widely used by developers worldwide. Xcode includes a suite of tools, such as Interface Builder, Instruments, and iOS Simulator, along with an extensive SDK for building iOS apps.
Employability	-
Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-

Environment & Sustainability	-
Unit II	Objective-C
Local	-
Regional	-
National	
Global	Objective C can be used globally with its syntax and syntactic rules
Employability	-
Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	Encapsulating Data
Local	In programming, "local" usually refers to variables, data, or methods that are confined to a specific scope, such as within a function or a block. Local variables are only accessible within the block or function where they are declared.
Regional	-
National	-
Global	
Employability	
Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	Developing iOS Applications
Local	-
Regional	-
National	"National" might represent initiatives or policies related to iOS app development adopted or regulated at the national level. For example, it could include national-level educational programs or government-supported initiatives promoting digital skills and app development.
Global	
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	Skill Development / Coding / Programming Software

Mobile Application Development using iOS Lab

Department:	**Department of Computer Science and Engineering**		
Course Name: Mobile Application Development using iOS Lab	**Course Code**	**L-T-P**	**Credits**
	ENSP459	**0-0-2**	**1**
Type of Course:	**Minor**		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Create iPhone apps using Objective-C and Apple's new programming language, use industry tools and frameworks such as Cocoa, Xcode, UIKit, Git.
CO 2	Understand and know how to use properly UIKit, asynchronous code, Core Image, NSURL Session and JSON Map Kit and Core Location, Auto Layout, Source Control, Core Data, Animation, and the app submission process.
CO 3	Read and write programs based on Objective-C, also have a strong grasp of Objective-C objects
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.	CO2
2	Case study of Windows and MAC systems	CO2
3	Case Study of XCode based on MAC Systems	CO2
4	Design an App for UISwitch based on Objective-C language	CO1
5	Design an App for UISlider based on Objective-C language	CO1
6	Design an App for UIStepper based on Objective-C language	CO1
7	Write a program for creating Story Boards	CO1
8	Design an App for UIAnimation based on Objective-C language	CO1

9	Create a Simple Calculator using Objective-C Language	CO1
10	Design an App for UIProgress Bar based on Objective-C language	CO1
11	Design an App for UIDatePicker Bar based on Objective-C language	CO1
12	Write an Objective-C program to print factorial of a given number	CO3
13	Write an Objective-C program to print Fibonacci series	CO3
14	Write an Objective-C program that displays the Phrase "Hello World"	CO3
15	Write an Objective-C program for displaying the value of variables	CO3
16	Write an Objective-C program for displaying the sum and subtraction of two variables	CO3
17	Write an Objective-C program for displaying the multiplication and division of the two variables	CO3
18	Write an Objective-C program that demonstrate control structure of Objective-C language	CO3
19	Create a Button using Objective-C	CO3
20	Write an Objective-C program to print the value of a variable inside a text, place it in parentheses, and insert a backslash just prior to the opening parenthesis	CO3
21	Write an Objective-C program to print Floyd's Triangle.	CO3
22	Write an Objective-C program to print palindrome of a number.	CO3
23	Write an Objective-C program to print pyramid.	CO3
24	Write an Objective-C program to find greatest number in between three numbers	CO3
25	Write an Objective-C program to check whether a number is even or odd.	CO3
	Mini Project 1: Make an interactive project based on iOS App using Objective-C Language	CO4
	Mini Project 2: Upload your iOS App in Apple AppStore and Publish it	CO4

DevOps & Automation

Department:	**Department of Computer Science and Engineering**		
Course Name: DevOps & Automation	**Course Code**	**L-T-P**	**Credits**
	ENSP411	**4-0-0**	**4**
Type of Course:	Minor		
Pre-requisite(s), if any:			

Brief Syllabus:

- DevOps is basically creating a niche or environment that emphasize bringing both development and operational team together. The key objective is to concentrate on the requirements of the project or the entire business requirement.
- Analysis: Analysis of the entire business requirement and then gathering the necessary information or data.
- Design: Putting all the gathered data into a proper format and then proceed with the development activity.
- The development teams should develop code: Optimized and ready to move codes.
- Compilation: Simultaneous compilation of codes to keep a check on the beauty of the code.
- Test: Without this phase, any software product is not ready for deployment; therefore, it is very much needed to go through testing in each phase.

UNIT WISE DETAILS

Unit Number: 1	**Title:**	**No. of hours: 8**

Content Summary:
Learning Objectives, DevOps Overview, Relationship between Agile and DevOps, DevOps Tool chain, Challenges with the traditional approach, Addressing challenges through DevOps, DevOps approach to the challenges, Overview of the DevOps tools, workflow of DevOps, JIRA
Suggested sources: https://www.atlassian.com/software/jira/guides/use-cases/what-is-jira-used-for.

Unit Number: 2	**Title:**	**No. of hours: 12**

Content Summary:

VERSION CONTROL SYSTEMS: Overview of version control systems – role of version control systems – Types of control systems and their supporting tools – Overview of Git – Overview of Source code and Version Control hosts – Deploy the files to GitHub**.**

Unit Number: 3	**Title:**	**No. of hours: 12**

Content Summary:

CONTINUOUS INTEGRATION AND BUILDING TOOL: Importance of continuous Integration, Overview and Features of Jenkins, set up Jenkins, Overview and Features of Maven, - Setup Maven, Overview and Features of TeamCity, Setup TeamCity

Suggested Source:

1. https://www.jenkins.io/doc/
2. http://maven.apache.org/ 3.
https://www.tutorialspoint.com/continuous_integration/continuous_integration_creating_proj ect_teamcity.htm

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

Content Summary:

SOFTWARE AND AUTOMATION TESTING FRAMEWORKS: Software Testing overview, Testing levels Approach and Automation Tools, Test driven development approaches and JUnit5, Behaviour driven development approach with cucumber.

Suggested Source:

https://howtodoinjava.com/junit-5-tutorial/

https://junit.org/junit5/docs/current/user-guide/

CONFIGURATION MANAGEMENT TOOLS: Overview of configuration management tools, overview of puppet, puppet configuration, overview of Chef, Chef configuration, overview of Ansible, Ansible configuration, containerization and Docker.

Suggested Source:

1. https://www.tutorialspoint.com/puppet/index.htm
2. https://puppet.com/blog/how-get-started-puppet-beginners-guide/
3. https://www.tutorialspoint.com/chef/index.htm
4. https://docs.chef.io/chef_overview/
5. https://www.tutorialspoint.com/ansible/index.htm
6. https://docs.ansible.com/ansible/latest/user_guide/intro_getting_started.html
7. https://docker-curriculum.com/

***Self-Learning Components:** mention 4-5 topics for students in bullet points 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: 1. Jez Humble and David Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Pearson Education, Inc.2011 2. Jennifer Davis, Katherine Daniels, Effective DevOps: Building a Culture of Collaboration, Affinity, and Tooling at Scale, O'Reilly, 2016 REFERENCE BOOKS 3. Gene Kim, Jez Humble, Patrick Debois, and John Willis, THE DEVOPS HANDBOOK How to Create World-Class Agility, Reliability, & Security in Technology Organizations, IT Revolution Press, 2016.

Define Course Outcomes (CO)

COs	Statements
CO1	Identify the difference between Agile and DevOps.
CO2	Practice of GitHub
CO3	Illustrate various Building tools
CO4	Analyze various Testing tools
CO5	Illustrate various Configuration management tools.

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels©	Affective levels(A)	Psychomotor levels(P)
	1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2		P1

C02	C3		P2
CO3	C3		P3
CO4	C1		-
CO5	C1		P1

*

CO-PO Mapping

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

CO-PSO Mapping

PO	PO1	PO2	PO3	PSO4
CO1	3	1	-	1
CO2	3	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	Introduction
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	
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	
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	

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

DevOps & Automation Lab

Department:	**Department of Computer Science and Engineering**		
Course Name: DevOps & Automation Lab	**Course Code**	**L-T-P**	**Credits**
	ENSP461	0-0-2	1
Type of Course:	Minor (Department Elective IV)		
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	CO1, CO2, CO3, 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.

.NET Framework

Department:	**Department of Computer Science and Engineering**		
Course Name: .NET Framework	**Course Code**	**L-T-P**	**Credits**
	ENSP413	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			
Brief Syllabus: The ".NET Framework" syllabus covers introduction and components of .NET, programming languages, Visual Studio, OOP, exception handling, memory management, Windows Forms/WPF, ASP.NET, web services, .NET Core, Entity Framework, and WCF. Emphasis on practical application and development skills for building robust and secure applications.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Introduction to .NET Framework**	**No. of hours: 8**	
Content Summary: Overview of .NET Framework ,Introduction to the .NET platform, Evolution and history of .NET Framework, Key components and architecture of .NET Framework, Common Language Runtime (CLR) and Just-In-Time (JIT) compilation, Common Intermediate Language (CIL) and Intermediate Language (IL), Programming Languages in .NET (C# as the primary language for .NET development & Visual Basic .NET) ,Introduction to Visual Studio IDE, Installation and configuration of .NET Framework and Visual Studio, NuGet package manager and third-party libraries			
Unit Number: 2	**Title: .NET Framework Fundamentals**	**No. of hours: 8**	
Object-Oriented Programming (OOP) in .NET, Classes, objects, and inheritance, Exception Handling and Debugging, Debugging techniques and tools in Visual Studio, Logging and error reporting in .NET applications, Memory Management and Garbage Collection, Automatic memory management in .NET, Garbage collection concepts and algorithms, Finalizers and the Dispose pattern, Performance considerations and best practices			
Unit Number: 3	**Title: Building Applications with .NET Framework**	**No. of hours: 12**	

Windows Forms and WPF Applications, Introduction to Windows Forms and Windows Presentation Foundation (WPF), Designing user interfaces using WinForms/WPF controls, Event-driven programming and event handling, Data binding and data access in WinForms/WPF applications, ASP.NET Web Development, Data access and validation in ASP.NET applications, Web Services and RESTful APIs, Creating and consuming web services in .NET, Authentication and security considerations in web services.

Unit Number: 4	**Title: Advanced Topics in .NET Framework**	**No. of hours: 12**

Content Summary:
.NET Core and Cross-Platform Development, Introduction to .NET Core and its advantages, Building cross-platform applications with .NET Core, Deploying and hosting .NET Core applications, Entity Framework and Database Connectivity, Overview of Entity Framework and Object-Relational Mapping (ORM), Creating and manipulating databases with Entity Framework, Querying data using LINQ (Language Integrated Query), Handling database migrations and versioning, Windows Communication Foundation (WCF), Introduction to WCF and service-oriented architecture (SOA), Creating and consuming WCF services, Message exchange patterns and bindings in WCF, Security and reliability in WCF applications

***Self-Learning Components:**

1. Online Tutorials and Documentation: Direct students to the official Microsoft documentation for .NET Framework, which provides comprehensive guides and resources. Microsoft .NET Documentation
2. Hands-on Coding Exercises: Assign coding exercises from platforms like LeetCode or HackerRank that focus on implementing concepts of .NET Framework. LeetCode HackerRank
3. Project-Based Learning: Encourage students to work on small projects using different aspects of the .NET Framework. Provide examples of project ideas and resources like GitHub repositories for inspiration. GitHub

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

Define Course Outcomes (CO)

COs	Statements
CO1	Knowledge: Understanding the fundamental concepts and components of the .NET Framework.
CO 2	Application: Applying knowledge to design and develop applications using Windows Forms, WPF, and ASP.NET.
CO 3	Analysis: Analyzing performance considerations and troubleshooting errors in the .NET Framework.
CO 4	Synthesis: Integrating advanced topics like .NET Core, Entity Framework, and WCF for cross-platform development and service creation.
CO 5	Evaluation: Assessing security, reliability, scalability, and performance of applications developed using the .NET Framework.

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	-	-	-
CO5	C5	-	P5

CO-PO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	**Introduction to .NET Framework**
Local	
Regional	
National	Understanding the evolution and history of the .NET Framework provides context specific to the national software development landscape.
Global	
Employability	Understanding the .NET Framework provides valuable skills and knowledge sought by employers in the software development industry.
Entrepreneurship	Understanding the .NET platform can support the development of innovative software products and services.
Skill Development	Studying the .NET Framework helps develop technical skills in application development and programming.
Professional Ethics	Familiarity with the .NET Framework enables professionals to adhere to ethical practices while developing software applications.
Gender	
Human Values	-
Environment & Sustainability	-
Unit II	**NET Framework Fundamentals**
Local	Understanding OOP in .NET can have local relevance in terms of specific programming practices and patterns adopted within the local software development community.
Regional	-
National	Understanding classes, objects, and inheritance in the .NET Framework is relevant at all levels of software development.
Global	Aligns with global trends in internet technologies and network protocols
Employability	Knowledge of classes, objects, and inheritance enhances employability in the software development field.
Entrepreneurship	Knowledge of OOP in .NET supports the development of innovative software products and services by entrepreneurs.
Skill Development	Learning about classes, objects, and inheritance improves technical skills required
Professional Ethics	
Gender	-
Human Values	-

Environment & Sustainability	-
Unit III	**Building Applications with .NET Framework**
Local	Understanding Windows Forms and WPF applications is relevant at a local level as it involves designing user interfaces and developing desktop applications specific to the local context.
Regional	Understanding Windows Forms and WPF applications is relevant at a regional level as these frameworks are commonly used in software development within a specific region.
National	Introduction to Windows Forms and WPF is important at a national level as these frameworks serve as the foundation for developing various types of applications used nationwide.
Global	Introduction to Windows Forms and WPF is important globally as these frameworks are fundamental to developing user interfaces and applications used on a global scale.
Employability	Understanding the basics of Windows Forms and WPF is crucial for employability in software development roles. Proficiency in these frameworks demonstrates competence and versatility, making individuals more desirable to potential employers.
Entrepreneurship	Knowledge of Windows Forms and WPF applications enables entrepreneurs to create innovative software products and services, driving business growth and success.
Skill Development	Understanding Windows Forms and WPF applications enhances technical skills in software development, enabling individuals to design and develop user-friendly and visually appealing applications.
Professional Ethics	Familiarity with Windows Forms and WPF applications ensures adherence to ethical standards in software development, including data privacy, accessibility, and industry best practices for usability and security.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Advanced Topics in .NET Framework**
Local	the content on .NET Core, Entity Framework, and Windows Communication Foundation (WCF) has local relevance as it addresses the specific development and database connectivity needs within the local context.
Regional	-

National	The content on .NET Core, Entity Framework, and Windows Communication Foundation (WCF) is nationally relevant as it addresses specific development and database connectivity needs within the country, considering national requirements and technologies.
Global	addresses development and database connectivity needs on a global scale, considering international requirements and technologies.
Employability	Content equips individuals with the skills and knowledge necessary for software development roles. Proficiency in these technologies enhances employability prospects and opens up opportunities in the job market.
Entrepreneurship	Understanding these technologies enables entrepreneurs to create and scale their own software ventures.
Skill Development	Learning and applying these technologies contribute to the development of practical skills that are in demand in the industry, improving professional capabilities and career prospects
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 9, SDG 4, and SDG 8,
NEP 2020	-
POE/4th IR	the content on .NET Core, Entity Framework, and Windows Communication Foundation (WCF) addresses professional ethics and aligns with the demands and innovations of the Fourth Industrial Revolution (4IR).

.Net FRAMEWORK Lab

Department:	**Department of Computer Science and Engineering**		
Course Name: .Net FRAMEWORK Lab	**Course Code**	**L-T-P**	**Credits**
	ENSP463	**0-0-2**	**1**
Type of Course:	Minor		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Knowledge and Understanding: Gain a thorough understanding of the core concepts and components of the .NET Framework.
CO 2	Application and Problem Solving: Apply .NET Framework knowledge to design and develop applications, solving programming problems effectively.
CO 3	Analyze and troubleshoot .NET applications, using debugging techniques and optimizing performance.
CO 4	Integrate advanced .NET topics like .NET Core, Entity Framework, and WCF to create cross-platform applications, work with databases, and build services.

Ex. No	Experiment Title	Mapped CO/COs
1	Installing and setting up the .NET Framework, Visual Studio IDE, and NuGet package manager	CO1
2	Creating a basic console application in C# or Visual Basic.NET and running it in Visual Studio.	CO1
3	Write a program to display "Hello World" using C#.	CO2
4	Create a Windows Forms application to design a simple calculator.	CO2
5	Develop a console application to perform basic arithmetic operations	CO2
6	Create a class hierarchy to represent different types of vehicles.	CO2
7	Implement inheritance and polymorphism concepts in a C# program.	CO2
8	Design a Windows Forms application to manage student records.	CO3

9	Create a WPF application to build a simple photo gallery.	CO3
10	Develop a web application to display and manage a list of books using ASP.NET..	CO3
11	Implement form validation and data access in an ASP.NET application.	CO3
12	Build a RESTful API using ASP.NET Web API to perform CRUD operations on a database.	CO3
13	Create a client application to consume a web service and display the retrieved data.	CO2
14	Implement a cross-platform application using .NET Core.	CO3
15	Develop a database-driven application using Entity Framework for data manipulation.	CO3
16	Design and implement a WCF service to provide secure communication between client and server.	CO4
17	Connect a .NET application to a database using ADO.NET and retrieve data.	CO3
18	Use LINQ (Language Integrated Query) to perform data querying and manipulation operations.	CO3
19	Deploy a .NET application to a web server or a cloud platform.	CO4
20	Configure and manage the hosting environment for a .NET application.	CO4
21	Use debugging techniques and tools in Visual Studio to identify and fix bugs in a program.	CO2
22	Create a program to demonstrate the automatic memory management feature in .NET.	CO4
23	Implement a program to analyze and optimize memory usage in a .NET application.	CO2
24	Develop a WCF service to perform CRUD operations on a database.	CO4
25	Design a client application to consume the WCF service and display the retrieved data.	CO4

New-Age programming languages (GO, F#, Clojure, Kotlin)

Department:	**Department of Computer Science and Engineering**		
Course Name: New-Age programming languages (GO, F#, Clojure, Kotlin)	**Course Code**	**L-T-P**	**Credits**
	ENSP415	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			

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.

UNIT WISE DETAILS

Unit Number: 1	**Title: GO programming Language**	**No. of hours: 10**

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.

Unit Number: 2	**Title: F# Programming Language**	**No. of hours: 10**

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
CO1	2	2			2					2
CO2	2	2			2					2
CO3	2	2		3						
CO4					3					3
CO5						2				2

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

CO-PSO Mapping

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

CO5		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.

New Age Programming languages Lab

Department:	**Department of Computer Science and Engineering**			
Course Name: New Age Programming languages Lab		**Course Code**	**L-T-P**	**Credits**
		ENSP465	0-0-2	2
Type of Course:		Programme Elective-II		
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.

Proposed Lab Experiments

Ex. No	Experiment Title	Mapped CO/COs
Practicals 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 deleting tasks. The tasks should have attributes such as title, description, due date, and status (e.g., "in progress", "completed").	CO5
Practicals 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).	CO1

	Write pure functions that demonstrate the map, filter, reduce/fold operations.	
13	Implement a program that performs multiple I/O-bound or computationally intensive tasks concurrently using F#'s asynchronous workflows and parallel programming constructs.	CO3
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
Practicals 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 responses, and render dynamic content.	CO5
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
Practicals on Kotlin Programming Language		
25	WAP for print following o/p Hello Kotlin!!!	CO2

		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%	
		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 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.		CO2
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		CO5

	features like a timer, score calculation, and a database of questions.	

MINOR PROJECT-I

Department:	**Department of Computer Science and Engineering**		
Course Name: Minor Project-I	**Course Code**	**L-T-P**	**Credits**
	SIBC252	---	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. 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)	