



K.R. MANGALAM UNIVERSITY



SCHOOL OF ENGINEERING

AND

TECHNOLOGY

**Department of Computer Science and
Engineering**

B.Sc. (H) (Cyber Security)

Undergraduate Course

2023-26



Preamble

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

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

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

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

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

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

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

Dean,

School of Engineering and Technology,

K. R. Mangalam University



Preface

The ability to secure information within a modern enterprise—large or small—is a growing challenge. Threats to information security are global, persistent, and increasingly sophisticated. Present pandemic situation has led to massive shift in use of digital information system for various purposes. This has fuelled the exponential increase in the cyber-related crimes. Effective information security at the enterprise level requires participation, planning, and practice. It is an ongoing effort that requires management and staff to work together from the same script. The information security community has to constantly evolve itself to develop a variety of resources, methods, and best practices to help modern enterprises address the challenge.

K.R. Mangalam University is in the process of transforming to National Educational Policy 2021. In consultation with Deans, Faculty Members, Industry Experts, and University Alumni, the Academic council constituted department-wise committees to draft the curriculum of B.Sc. (H) Cyber Security. The primary emphasis is to designing a course that addresses mathematical foundation concerning cyber security, legal and ethical perspective and practical aspects concerning enterprise security. The major thrust is on to prepare students for future careers as cyber security professionals.

The B.Sc. (H) Cyber Security course is spread over three years in six semesters. The total numbers of credits are 134. The course is designed as per LOCF guidelines laid by UGC. The core course includes specialized courses pertaining to cyber security along with few cores courses that are taught in B.Sc. (H) Computer Science. The generic electives offered deal with mathematical foundations that necessary to strengthen the



development of computer security algorithms. Departmental Specific Electives addresses the need to familiarize students with emerging areas in computer science. The laboratories, besides supplementing the theory course should also expose the student to the use of the latest software tools.

The present curriculum focuses on effectively applying analytical and critical thinking to plan and execute security measures to shield an organization's computer systems, networks, and networked devices from infiltration and cyber-attacks.

Objectives of the program

After the completion of the degree, students would

- Provide students with a solid foundation in the fundamental concepts, theories, and principles of cyber security. This includes knowledge of computer systems, networks, cryptography, secure programming, risk assessment, and incident response.
- The program focuses on equipping students with hands-on skills in identifying vulnerabilities, securing systems and networks, and implementing effective defense strategies.
- The program emphasizes the development of analytical and problem-solving skills specific to the field of cyber security.
- The program emphasizes the importance of ethical and legal aspects of cyber security.



Career Avenues

A Bachelor of Science (Honors) program in Cyber Security opens up a wide range of career avenues in the rapidly growing field of cybersecurity. Some of the potential career paths and job opportunities include:

- System Security Specialist
- Information Technology Forensic Analyst
- Ethical Hacker
- Compliance officer
- Data Security Analyst
- Cryptanalyst
- Network Security Specialist
- Web Applicant

Prospective Companies

- Amazon
- Flipkart
- Wipro
- IBM
- Infosys
- Deloitte
- Walmart
- Genpact
- Accenture
- Microsoft



Duration

3 Years (Full-Time)

Eligibility Criteria

Candidates must have passed 10+2 examination or equivalent in any stream with mathematics as one subject and with minimum 50% aggregate marks.



Table of Contents

S.N.	CONTENT	Page No.
1.	University Vision & Mission	
2	School Vision & Mission	
2.	About School	
3	Program Objectives (PO)	
4	Program Educational Objectives (PEO)	
5	Program Specific Objectives (PSO)	
6	Program Highlights	
7	Scheme of Studies	
8	Categorization of Courses	
9	Course Template	
10	CO Mapping with Bloom's Taxonomy	
11	CO-PO/PSO Mapping	
12	Mapping with components of Global/National/Regional/Local aspects, employability, skill development, SDG, NEP-20 etc	



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 instil in learners an entrepreneurial mind set 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.

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 and BSc (h) programs are in collaborations with Industries like IBM, Siemens, Samatrix, Xebia, ImaginXP etc. Our students get an opportunity to learn directly by professionals from industry.



Program Outcome (PO)

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 mind set, and leverage innovative ideas to generate value and contribute to the betterment of individuals and society.

Program Educational Objectives (PEO)

PEO1 - Develop a deep understanding of the theoretical and practical aspects of Cyber Security through rigorous education and research.

PEO2 - Pursue a successful career in Cyber Security-related industries, government organizations, or academia.

PEO3 - Apply critical thinking, problem-solving skills, and emerging technologies to develop innovative Cyber Security solutions that address real-world challenges.

PEO4 - Demonstrate leadership, effective communication, ethical behavior, and social responsibility in the practice of Cyber Security with a commitment to lifelong learning and professional development.



Program Specific Outcomes (PSO)

PSO1 - Acquire comprehensive knowledge and skills in theoretical and practical aspects of Cyber Security.

PSO2 - Obtain employment in Cyber Security-related industries, government organizations, or academia.

PSO3 - Apply critical thinking, problem-solving skills, and emerging technologies to develop innovative Cyber Security solutions.

PSO4 - Demonstrate leadership, effective communication, ethical behaviour, and social responsibility in the practice of Cyber Security.

Programme Highlights

- The curriculum is specifically designed in consultation with industry insiders and experts of cyber security
 - Realistic hands-on training for absolute excellence.
 - Consistent mentoring by acclaimed academicians and top industry experts.
 - Highly sophisticated laboratories equipped with cutting-edge tech apparatus.
 - Ensuring absolute preparedness for successful career progression



Program Scheme

Semester I

SN	Category	COURSE CODE	Course Title	L	T	P	C
1	Major	ENBC101	Fundamentals of Web Technologies	4	-	-	4
2	Major	ENBC103	Matlab Programming	4	-	-	4
3	SEC	SEC050	Linux Environment Lab	-	-	2	2
4	Minor	ENSP107	Introduction to Computer Science and Programming in Python	4	0	-	4
5	Major	ENBC151	Fundamentals of Web Technologies Lab	-	-	2	1
6	Major	ENBC153	Matlab Programming Lab	-	-	2	1
7	Minor	ENSP155	Computer Science and Programming in 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	ENSP114	Network Defence Essentials	4	-	-	4
2	Major	ENBC102	Introduction to Discrete Structures	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
6	Minor	ENSP166	Network Defence Essentials Lab	-	-	2	1
7	Major	ENBC152	Basics of Operating Systems Lab	-	-	2	1
8	Major	ENBC154	Concepts of Object Oriented Programming Using C++ Lab	-	-	2	1
9	VAC		Extension Activities(community engagement service)	3	-	-	2
			Open Elective-I	3	-	-	3
			TOTAL	19	3	6	24

**Semester III**

SN	Category	COURSE CODE	Course Title	L	T	P	C
1	Major	ENBC201	Introduction to Data Structures	3	1	-	4
2	Minor	ENSP207	Fundamentals of Cryptography	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	ENSP259	Fundamentals of Cryptography Lab	-	-	2	1
9	VAC		VAC -3	-	-	-	2
10	Summer Internship	SIBC251	Summer Internship/Project-I	-	-	-	2
	TOTAL			17	2	6	26

**Semester IV**

SN	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
6	Major	ENBC254	Fundamentals of Algorithm Design & Analysis Lab	-	-	2	1
7	Major	ENBC256	Introduction to Computer Networks Lab	-	-	2	1
8	AEC	AEC012	Life Skills for Professionals-II	3	-	-	3
9	Proj	SIBC252	Minor Project	-	-	-	2
10	SEC	SEC036	Competitive Coding Lab	-	-	4	2
11	Open Elective		Open Elective-II	3	-	-	3
TOTAL				15	3	10	25

**Semester V**

SN	Category	COURSE CODE	Course Title	L	T	P	C
1	Major	ENBC301	Computer Organization and Architecture	4	-	-	4
2	Minor	ENSP319	Essentials of Ethical Hacking	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	ENSP367	Ethical Hacking Lab	-	-	2	1
10	Summer Internship	SIBC351	Summer Internship/Project				2
11	AEC	AEC013	Life Skills for Professionals-III	3	-	-	3
TOTAL				19	0	6	24



Department Elective-I (Cyber Security)							
1	Minor	ENSP401	Secure Coding and Vulnerabilities	4	0	0	4
	Minor	ENSP451	Secure Coding and Vulnerabilities lab	-	-	2	1
2	Minor	ENSP403	Cyber Crime Investigation & Digital Forensics	4	0	0	4
	Minor	ENSP453	Cyber Crime Investigation & Digital Forensics lab	-	-	2	1
3	Minor	ENSP405	AI in Cyber Security	4	0	0	4
	Minor	ENSP455	AI in Cyber Security Lab	-	-	2	1
4	Minor	ENSP407	Social Media Security	4	0	0	4
	Minor	ENSP457	Social Media Security Lab	-	-	2	1
Department Elective - II (Full Stack Development)							
(i)	Minor	ENSP409	Mobile Application Development using IOS	4	-	-	4
	Minor	ENSP459	Mobile Application Development using IOS Lab	-	-	2	1
(ii)	Minor	ENSP411	DevOps & Automation	4	-	-	4
	Minor	ENSP461	DevOps & Automation Lab	-	-	2	1
(iii)	Minor	ENSP413	.Net FRAMEWORK	4	-	-	4
	Minor	ENSP463	.Net FRAMEWORK Lab	-	-	2	1
(iv)	Minor	ENSP415	New Age Programming languages	4	0	0	4
	Minor	ENSP465	New Age Programming languages Lab	0	0	2	1



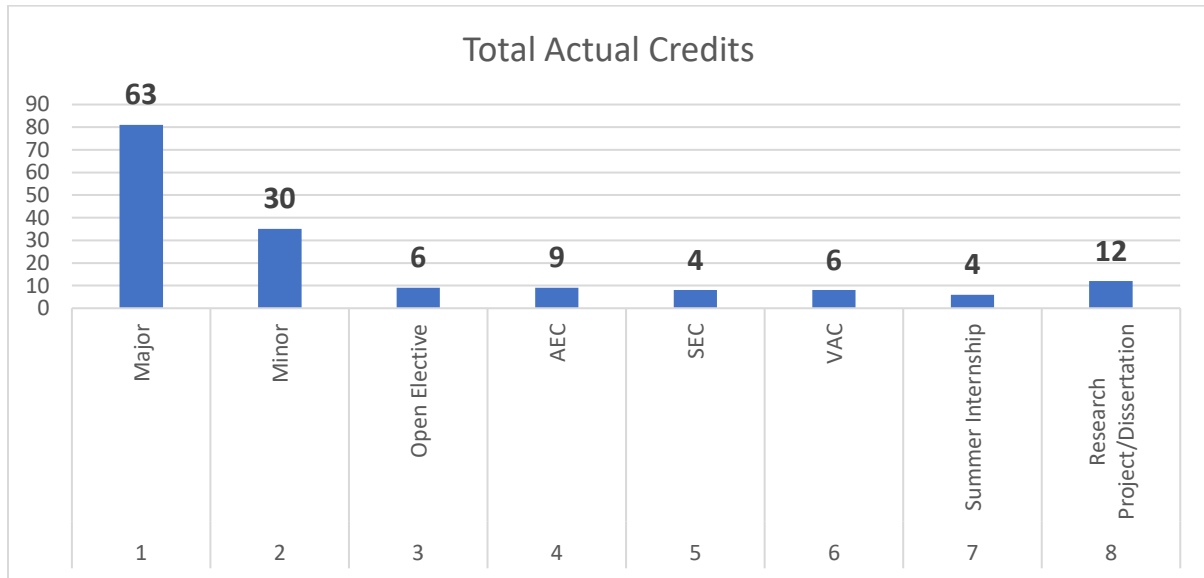
Semester VI

S.No	Category	Course Code	Course Title	L	T	P	C
1	Project		Major Project/Industrial Training/Startup	-	-	-	12
2			TOTAL	16	4	8	12
TOTAL CREDITS							134

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:			
Brief Syllabus: This course is an introduction to Web site development and the technologies behind it. Students will learn how to design and develop Web pages using current technologies and tools. Topics covered will include the World Wide Web, HTML, Cascading Style Sheets (CSS) and XML. The focus of this course is on dynamic HTML, a collection of web technologies such as HTML and scripting languages used together to create interactive and animated Web pages. Students will learn to program client-side scripts using JavaScript and the Document Object Model to transform static Web pages created with HTML and CSS into dynamic Web pages.			
UNIT WISE DETAILS			
Unit Number: 1	Title: Introduction	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	Title: Java Script	No. of hours: 12	
Content Summary: Client-side scripting: JavaScript - 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	Title: Web design Issue	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	Title: XML	No. of hours: 8
Content Summary: XML: 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)



COs	Statements
CO1	Create a well-designed and well-formed, professional Web site utilizing the most current standards and practice
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 markup 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	C1		P1
CO2	C1		P2
CO3	C3		P3
CO4	C4		-
CO5	C5		P3

***Please Note: Map only 1 or 2 Levels in each category. If a higher level is given, no need to mention lower level**

CO-PO Mapping



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

Please Note:

- Refer to POs while mapping each CO.
- Mark “ - ” if not applicable
- If attainment of a CO is strongly mapped with a PO , Mark 3
- If attainment of a CO is moderately mapped with a PO , Mark 2
- If attainment of a CO is weakly mapped with a PO , Mark 1

Justification for mapping must be relevant

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

CO-PSO Mapping

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



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



FUNDAMENTALS OF WEB TECHNOLOGIE 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:			

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.

Proposed Lab 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 Style sheet 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: · 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	CO2,CO3
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: Matlab Programming	Course Code	L-T-P	Credits
	ENBC103	4-0-0	4
Type of Course:	Major		
Pre-requisite(s), if any:			
Brief Syllabus: <p>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.</p>			
UNIT WISE DETAILS			
Unit Number: 1	Title: Introduction to MATLAB	No. of hours: 10	
Content Summary: <p>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.</p>			
Unit Number: 2	Title: Operations & Plots	No. of hours: 12	



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 labeling, curve labeling 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: 10
-----------------------	-------------------------------	-------------------------

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) Introduce the concept of Simulink and its interface. Explain the difference between
- 2) Describe the numerical integration method in MATLAB for solving calculus problems.
- 3) <https://www.mygreatlearning.com/academy/learn-for-free/courses/matlab>
- 4) <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 LawrieBrown "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© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1	-	P1
CO2	C2	A3	-
CO3	C3	-	-
CO4	C4	-	P2
CO5	C5	A4	-

***Please Note: Map only 1 or 2 Levels in each category. If a higher level is given, no need to mention lower level**

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

Please Note:

- Refer to POs while mapping each CO.
- Mark “ - ” if not applicable
- If attainment of a CO is strongly mapped with a PO , Mark 3
- If attainment of a CO is moderately mapped with a PO , Mark 2
- If attainment of a CO is weakly mapped with a PO , Mark 1

Justification for mapping must be relevant



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

CO-PSO Mapping

PSO	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/4 th IR	The principles of the Fourth Industrial Revolution, fostering holistic development, skill development, digital literacy, and addressing emerging challenges in the digital era.

MATLAB PROGRAMMING LAB



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

Defined Course Outcomes

COs	
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.

Proposed Lab 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	CO1, CO2



	numerical integration.	
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:	Course Code	L-T-P	Credits
Linux Environment Lab	SEC050	0-0-2	2
Type of Course:	SEC		
Pre-requisite(s), if any: Basic understanding of computer systems and familiarity with operating systems			

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.

Proposed Lab 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



8. "edX - Introduction to Linux" (Online Course): A free online course provided by The Linux Foundation, introducing Linux fundamentals, command-line usage, file management, and basic system administration. Website: edx.org
9. "Udemy - Linux Administration Bootcamp" (Online Course): A comprehensive course that covers Linux administration, including system configuration, networking, security, and shell scripting. It includes hands-on exercises and practical examples. Website: udemy.com



Department:	Department of Computer Science and Engineering		
Course Name: Introduction To Computer Science And Programming In Python	Course Code	L-T-P	Credits
	ENSP107	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any: -			
Brief Syllabus: "Introduction to Computer Science and Programming in Python" is a foundational course designed to equip students with essential programming skills and concepts for data science. The course covers fundamental components of a computer system, algorithm development, and Python programming basics. Students will learn about control structures, string manipulation, and the concept of abstraction through functions. They will also explore data structures like lists and dictionaries, as well as object-oriented programming and handling exceptions. The syllabus also includes file handling, search, and sorting algorithms. Students will gain proficiency in Python programming, problem-solving, and data manipulation, enabling them to apply these skills in real-world data science scenarios.			
UNIT WISE DETAILS			
Unit Number: 1	Title: Introduction to Programming and Computation	No. of hours: 10	
Content Summary: Components of a computer system: disks, memory, processor, operating system, compilers, etc., Introduction to computation and its principles, Steps to solve logical and numerical problems: idea of algorithms, Algorithm representation: flowcharts, pseudo code, From algorithms to programs: source code, variables, syntax, and logical errors.			
Unit Number: 2	Title: Python Basics and Control Structures	No. of hours: 12	
Content Summary: Introduction to Python programming language, Basic elements of Python, Branching programs: if statements, conditional execution, Iteration: loops while and for statements, String manipulation: operations, methods, formatting, Input and output handling, Introduction to functions and scoping.			
Unit Number: 3	Title: Data Structures and Abstraction	No. of hours: 8	



Content Summary:

Tuples and lists: operations, indexing, slicing, Aliasing, mutability, and cloning, Decomposition and abstraction principles, Functions and their role in abstraction, Recursion: concept and application, Dictionaries: key-value pairs, dictionary methods.

Unit Number: 4	Title: Program Development and Efficiency	No. of hours: 12
--------------------------	--	-------------------------

Content Summary:

Testing and debugging: strategies and techniques, Handling exceptions and assertions, Introduction to object-oriented programming, Python classes and inheritance, File handling: reading and writing files, Search algorithms: linear search, binary search, Sorting algorithms: selection sort, insertion sort, Hash tables and their applications.

Unit Number: 5	Title: Program Efficiency and Complexity Analysis	No. of hours: 10
--------------------------	--	-------------------------

Content Summary:

Understanding program efficiency, Introduction to algorithm analysis, Time complexity and Big O notation, Space complexity, Optimization techniques and best practices.

***Self-Learning Components:**

- **Online Tutorials and Video Lectures:**
 1. **Python for Everybody - University of Michigan Link:**
<https://www.py4e.com/>
 2. **Programming with Python - Coursera (Offered by the University of Michigan) Link:** <https://www.coursera.org/learn/python-programming>
 3. **Python Crash Course - YouTube Playlist by Corey Schafer Link:**
<https://youtube.com/playlist?list=PL-osiE80TeTskrapNbzxhwoFUilCjGgY7>
- **Interactive Coding Platforms:**
 1. **Codecademy Python Course Link:**
<https://www.codecademy.com/learn/learn-python-3>
 2. **DataCamp Python Courses Link:**
<https://www.datacamp.com/courses/intro-to-python-for-data-science>
- **Online Practice and Challenges:**
 1. **HackerRank Python Domain Link:**
<https://www.hackerrank.com/domains/python>
 2. **LeetCode Python Problems Link:**
<https://leetcode.com/problemset/all/?topicSlugs=python>
- **Data Science Case Studies and Projects:**
 1. **Kaggle Datasets and Kernels Link:**
<https://www.kaggle.com/datasets>
 2. **Dataquest Data Science Projects Link:**
<https://www.dataquest.io/projects/>
- **Python Documentation and Libraries:**
 1. **Python Official Documentation Link:** <https://docs.python.org/3/>



2. NumPy User Guide Link: <https://numpy.org/doc/stable/>
3. Pandas User Guide Link:
https://pandas.pydata.org/docs/user_guide/index.html
- Open-Source Textbooks:
 1. "Think Python" by Allen B. Downey Link:
<https://greenteapress.com/wp/think-python-2e/>
 2. "Python for Data Science Handbook" by Jake VanderPlas Link:
<https://jakevdp.github.io/PythonDataScienceHandbook/>

Additional Online Study Sources from the Open-Source Society University (OSSU)

- OSSU Computer Science Curriculum:
Link: <https://github.com/ossu/computer-science>
- OSSU Data Science Curriculum:
Link: <https://github.com/ossu/data-science>
- MIT OpenCourseWare - Introduction to Computer Science and Programming in Python:
Link: <https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2020/>
- edX - Introduction to Computer Science and Programming Using Python (Offered by MIT):
Link: <https://www.edx.org/professional-certificate/introduction-to-computer-science-and-programming-using-python>
- Python Documentation - Official Python Tutorial:
Link: <https://docs.python.org/3/tutorial/>
- Python.org - Beginner's Guide to Python:
Link: <https://www.python.org/about/gettingstarted/>
- OSSU Community Forums:
Link: <https://github.com/ossu/computer-science/discussions>

Reference Books:

1. Guttag, John. Introduction to Computation and Programming Using Python: With Application to Understanding Data Second Edition. MIT Press
2. Python for Data Analysis: Data Wrangling with Pandas, NumPy, and IPython" by Wes McKinney
3. Learning Python" by Mark Lutz
4. Automate the Boring Stuff with Python: Practical Programming for Total Beginners" by Al Sweigart

Define Course Outcomes (CO)



COs	Statements
CO1	Comprehend the fundamental principles of computer programming and computation, including the components of a computer system and the idea of algorithms.
CO2	Demonstrate proficiency in Python programming language and apply it to develop programs using branching, iteration, and string manipulation techniques.
CO3	Utilize data structures such as lists, tuples, and dictionaries to organize and manipulate data efficiently in Python programs.
CO4	Implement object-oriented programming concepts, including classes, inheritance, and file handling, to build complex Python applications.
CO5	Analyze and evaluate the efficiency of algorithms and programs using time and space complexity analysis and apply optimization techniques for improved performance.

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	-	-
CO2	C2	-	-
CO3	C3	A4	P3
CO4	C3	A4	P4
CO5	C4, C5	A5	P5

***Please Note:**

Map only 1 or 2 Levels in each category. If a higher level is given, no need to mention lower level



CO-PO Mapping

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

Please Note:

- Refer to POs while mapping each CO.
- Mark “ - ” if not applicable
- If attainment of a CO is strongly mapped with a PO , Mark 3
- If attainment of a CO is moderately mapped with a PO , Mark 2
- If attainment of a CO is weakly mapped with a PO , Mark 1

Justification for mapping must be relevant

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators



Unit I	Introduction to Programming and Computation
Local	Relevant to local industries and businesses that rely on technology and data processing
Regional	Encourages local innovation and problem-solving through programming
National	Contributes to building a strong national workforce in the technology sector by imparting essential computer science knowledge
Global	Align with global relevance as computer programming and algorithms are essential components of the global tech industry
Employability	Improved employability prospects in local, regional, national, and even global tech companies.
Entrepreneurship	-
Skill Development	Focuses on developing critical skills such as problem-solving, logical thinking, and algorithm design, which are essential for students' overall skill development.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	Python Basics and Control Structures
Local	Relevant to local industries and businesses that rely on technology and data processing
Regional	Encourages local innovation and problem-solving through programming
National	Contributes to building a strong national workforce in the technology sector by imparting essential computer science knowledge
Global	Align with global relevance as computer programming and algorithms are essential components of the global tech industry
Employability	Improved employability prospects in local, regional, national, and even global tech companies.
Entrepreneurship	-
Skill Development	Focuses on developing critical skills such as problem-solving, logical thinking, and algorithm design, which are essential for students' overall skill development.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	Data Structures and Abstraction



Local	Knowledge of data structures and abstraction principles is valuable for local businesses handling large datasets.
Regional	It equips students with skills needed in regional data-driven industries such as finance, healthcare, and marketing.
National	The syllabus contributes to the development of a data-savvy workforce that can contribute to the national data ecosystem.
Global	Align globally as data-driven decision-making is essential in global industries and research.
Employability	Improved employability prospects in local, regional, national, and even global tech companies.
Entrepreneurship	-
Skill Development	Focuses on developing critical skills such as problem-solving, logical thinking, and algorithm design, which are essential for students' overall skill development.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	Program Development and Efficiency
Local	Knowledge of Program Development and Efficiency is valuable for local businesses handling large datasets.
Regional	It equips students with skills needed in regional data-driven industries such as finance, healthcare, and marketing.
National	The syllabus contributes to the development of a data-savvy workforce that can contribute to the national data ecosystem.
Global	Align globally as data-driven decision-making is essential in global industries and research.
Employability	Improved employability prospects in local, regional, national, and even global tech companies.
Entrepreneurship	-
Skill Development	Focuses on developing critical skills such as problem-solving, logical thinking, and algorithm design, which are essential for students' overall skill development.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit V	Program Efficiency and Complexity Analysis
Local	Understanding program efficiency is vital for local industries to optimize their computational resources for improved performance.



Regional	It equips students with skills needed to tackle regional data science challenges in various domains.
National	Aligns with national efforts to promote data-driven decision-making in various sectors for effective policymaking.
Global	The concepts covered in Unit V have global relevance as efficient programming practices are critical for global technology advancement.
Employability	
Entrepreneurship	
Skill Development	
Professional Ethics	
Gender	
Human Values	
Environment & Sustainability	
SDG	Aligns with the UN Sustainable Development Goal 4 (Quality Education) by imparting foundational computer science knowledge to students.
NEP 2020	The syllabus supports the National Education Policy 2020's emphasis on integrating technology and computational thinking in education.
POE/4 th IR	The syllabus prepares students for the Fourth Industrial Revolution by providing them with essential programming and computational skills.



INTRODUCTION TO COMPUTER SCIENCE AND PROGRAMMING IN PYTHON LAB

Department:	Department of Computer Science and Engineering		
Course Name: Introduction to Computer Science and Programming in Python Lab	Course Code	L-T-P	Credits
	ENSP155	0-0-2	1
Type of Course:	Minor		
Pre-requisite(s), if any: Python programming			

Defined Course Outcomes

COs	
CO 1	Apply fundamental programming concepts, including syntax, variables, control structures, functions, and data structures, to solve problems using the Python programming language.
CO 2	Utilize appropriate data structures such as lists, tuples, dictionaries, and understand their properties and operations. Implement basic algorithms like search and sort algorithms to solve problems efficiently.
CO 3	Design and implement object-oriented programs using classes, inheritance, and encapsulation concepts. Apply object-oriented principles to create reusable and modular code.
CO 4	Analyze the efficiency and complexity of algorithms and programs, including time complexity, space complexity, and other factors influencing performance. Make informed decisions to optimize code and improve overall efficiency.
CO 5	Analyze and visualize data using Python: Apply Python libraries and tools to analyze and visualize data, creating plots and charts for effective data presentation. Utilize Python's data manipulation capabilities to extract insights from datasets.

Proposed Lab Experiments



Ex. No	Experiment Title	Mapped CO/COs
1	Introduction to Python: Setting up Python environment	CO1
2	Basic Python Syntax and Variables	CO1
3	Branching Programs: Implementing If Statements	CO1
4	Iteration: Loops and While Statements	CO1
5	String Manipulation: Working with Text	CO1
6	Input and Output: Reading and Writing Files	CO1
7	Functions: Implementing Reusable Code	CO1
8	Scoping and Abstraction: Understanding Function Scope	CO1, CO3
9	Tuples and Lists: Operations and Indexing	CO2
10	Aliasing and Mutability: Working with Data Structures	CO2
11	Recursion: Implementing Recursive Functions	CO1
12	Dictionaries: Working with Key-Value Pairs	CO2
13	Object-Oriented Programming: Creating Classes	CO3
14	Inheritance and Encapsulation: Extending Classes	CO3
15	Exception Handling: Dealing with Errors	CO3
16	File Handling: Reading and Writing Data from Files	CO1
17	Linear Search: Searching for an Element in a List	CO2
18	Binary Search: Implementing Binary Search Algorithm	CO2
19	Selection Sort: Sorting Elements in a List	CO2
20	Insertion Sort: Sorting a List in Ascending Order	CO2
21	Hash Tables: Storing and Retrieving Data	CO2
22	Program Efficiency: Analyzing Time Complexity	CO4
23	Debugging Techniques: Identifying and Fixing Errors	CO1, CO3
24	Data Visualization: Creating Plots and Charts	CO5
25	Project 1: Implementing a Simple Calculator	CO1, CO2
26	Project 2: Creating a To-Do List Application	CO1, CO2, CO3
27	Project 3: Building a Basic Contact Management System	CO1, CO2, CO3
28	Project 4: Analyzing and Visualizing Data using Python	CO4, CO5
29	Final Project: Integrating Concepts into a Comprehensive App	CO1, CO2, CO3, CO4, CO5



Detailed syllabus

1. **Experiment 1:** Introduction to Python: Setting up Python environment.

Session 1:

Topic: Introduction to Python programming language

- Overview of Python and its features
- Installation and setup of Python environment

Exercise: Write a Python program to display "Hello, World!" on the console.

Project: Setting up a Python development environment: Install Python, set up an integrated development environment (IDE), and execute a sample program.

2. **Experiment 2:** Basic Python Syntax and Variables

Session 1: Topic: Basic data types and variables in Python

- Integers, floats, strings, booleans, lists, dictionaries
- Variable declaration and assignment statements

Exercise: Write a Python program to swap the values of two variables.

Project: Create a simple calculator program: Design and implement a calculator program that performs basic arithmetic operations.

3. **Experiment 3:** Branching Programs: Implementing If Statements

Session 1: Topic: Conditional statements and if statements

- Syntax and usage of if, elif, and else statements
- Comparison operators and logical operators

Exercise: Write a Python program to check if a given number is even or odd.

Project: Implement a grading system: Write a program that takes input marks and assigns grades based on predefined criteria.

4. **Experiment 4:** Iteration: Loops and While Statements

Session 1: Topic: Loops and while statements

- Syntax and usage of while loops
- Controlling loop execution using break and continue statements

Exercise: Write a Python program to calculate the factorial of a given number using a while loop.

Project: Implement a number guessing game: Create a program where the user guesses a randomly generated number within a specified range.

5. **Experiment 5:** String Manipulation: Working with Text

Session 1: Topic: String manipulation in Python

- Basic string operations: concatenation, length, indexing, slicing
- String methods for formatting and manipulation

Exercise: Write a Python program to reverse a given string.

Project: Text analysis program: Develop a program that analyzes a text file, counting the occurrence of specific words and characters.

6. **Experiment 6:** Input and Output: Reading and Writing Files

Session 1: Topic: File handling in Python

- Opening, reading, and writing to files
- Different file modes: read, write, append

Exercise: Write a Python program to read data from a text file and display it on the console.



Project: Create a basic address book application: Implement a program that allows users to add, view, and modify contact information stored in a text file.

7. **Experiment 7:** Functions: Implementing Reusable Code

Session 1: Topic: Functions in Python

- Defining and calling functions
- Function arguments and return values

Exercise: Write a Python function to calculate the factorial of a given number.

Project: Building a basic calculator application: Create a calculator program using functions to perform arithmetic operations.

8. **Experiment 8:** Scoping and Abstraction: Understanding Function Scope

Session 1: Topic: Scope of variables in Python

- Local and global variables
- Variable visibility and access

Exercise: Write a Python program to demonstrate variable scoping in nested functions.

Project: Implement a simple quiz game: Develop a program that presents a series of questions to the user and keeps track of their score.

9. **Experiment 9:** Tuples and Lists: Operations and Indexing

Session 1: Topic: Tuples and lists in Python

- Creating and manipulating tuples and lists
- Accessing elements using indexing and slicing

Exercise: Write a Python program to find the largest and smallest elements in a given list.

Project: Implement a grocery list application: Create a program that allows users to add, remove, and view items in a grocery list.

10. **Experiment 10:** Aliasing and Mutability: Working with Data Structures

Session 1: Topic: Aliasing and mutability in Python

- Understanding the concept of aliases and how they affect mutable objects
- Modifying mutable objects in-place

Exercise: Write a Python program to demonstrate the concept of aliasing and its impact on mutable objects.

Project: Building a simple inventory management system: Design and implement a program that tracks the quantity of items in stock and allows for adding and updating inventory.

11. **Experiment 11:** Recursion: Implementing Recursive Functions

Session 1: Topic: Recursive functions in Python

- Understanding recursion and its base case
- Implementing recursive algorithms

Exercise: Write a recursive Python function to calculate the nth Fibonacci number.

Project: Implementing a file system traversal: Create a program that recursively traverses a directory and lists all files and subdirectories.

12. **Experiment 12:** Dictionaries: Working with Key-Value Pairs

Session 1: Topic: Dictionaries in Python

- Creating and manipulating dictionaries
- Accessing and modifying values using keys



Exercise: Write a Python program to count the frequency of each character in a given string using a dictionary.

Project: Creating a simple dictionary application: Build a program that allows users to add, search, and delete word definitions stored in a dictionary.

13. **Experiment 13:** Object-Oriented Programming: Creating Classes

Session 1: Topic: Introduction to object-oriented programming (OOP)

- Understanding the basic principles of OOP
- Creating and using classes in Python

Exercise: Write a Python program to implement a class representing a circle and calculate its area and circumference.

Project: Developing a simple banking system: Design and implement a program that models a bank, with classes for customers, accounts, and transactions.

14. **Experiment 14:** Inheritance and Encapsulation: Extending Classes

Session 1: Topic: Inheritance and encapsulation in Python

- Creating subclasses and inheriting attributes and methods
- Understanding encapsulation and access modifiers

Exercise: Write a Python program to demonstrate inheritance by creating a hierarchy of classes representing different animals.

Project: Designing a simple game using inheritance: Develop a game where players control different characters with unique abilities inherited from a common superclass.

15. **Experiment 15:** Exception Handling: Dealing with Errors

Session 1: Topic: Exception handling in Python

- Handling and raising exceptions
- Using try-except blocks for error handling

Exercise: Write a Python program to handle an exception when dividing a number by zero.

Project: Creating a basic error logging system: Implement a program that logs errors and exceptions encountered during its execution.

16. **Experiment 16:** File Handling: Reading and Writing Data from Files

Session 1: Topic: Advanced file handling in Python

- Reading and writing data to files using different formats (e.g., CSV, JSON)
- Error handling during file operations

Exercise: Write a Python program to read data from a CSV file, perform basic data manipulation, and display the results.

Project: Developing a simple data analysis tool: Create a program that reads data from multiple files, combines and analyzes the data, and generates reports.

17. **Experiment 17:** Linear Search: Searching for an Element in a List

Session 1: Topic: Linear search algorithm

- Implementing and understanding linear search
- Analyzing the time complexity of linear search

Exercise: Write a Python program to search for a specific element in a given list using linear search.



Project: Implementing a basic contact management system: Build a program that allows users to search for contacts stored in a list.

18. **Experiment 18:** Binary Search: Implementing Binary Search Algorithm

Session 1: Topic: Binary search algorithm

- Understanding the concept of binary search
- Implementing and analyzing the efficiency of binary search

Exercise: Write a Python program to search for a specific element in a sorted list using binary search.

Project: Creating a simple phone book application: Design a program that performs efficient search operations on a sorted list of contacts.

19. **Experiment 19:** Selection Sort: Sorting Elements in a List

Session 1: Topic: Introduction to sorting algorithms

- Overview of sorting algorithms and their importance
- Introduction to selection sort algorithm and its principles
- Step-by-step execution of the selection sort algorithm

Exercise: Implement the selection sort algorithm in Python to sort a given list of elements.

Project: Sorting algorithm comparison: Compare the performance of the selection sort algorithm with other sorting algorithms (e.g., insertion sort, merge sort) by analyzing their time complexity and execution time.

20. **Experiment 20:** Insertion Sort: Sorting a List in Ascending Order

Session 1: Topic: Introduction to insertion sort algorithm

- Explanation of the insertion sort algorithm and its working principles
- Step-by-step execution of the insertion sort algorithm

Exercise: Implement the insertion sort algorithm in Python to sort a given list of elements.

Project: Sorting elements with different algorithms: Compare the efficiency and performance of the insertion sort algorithm with other sorting algorithms (e.g., selection sort, quicksort) by analyzing their time complexity and execution time on various datasets.

21. **Experiment 21:** Hash Tables: Implementing a Hash Table

Session 1: Topic: Hash tables and hash functions

- Understanding the concept of hash tables
- Implementing a basic hash table data structure

Exercise: Write a Python program to implement a hash table for storing and retrieving key-value pairs.

Project: Building a simple password manager: Design and implement a program that securely stores and retrieves user passwords using a hash table.

22. **Experiment 22:** Program Efficiency: Analyzing Time Complexity

Session 1: Topic: Analyzing time complexity of algorithms

- Introduction to time complexity analysis
- Big O notation and its significance
- Analyzing time complexity of basic algorithms (e.g., linear search, binary search)



Exercise: Write a Python program to analyze the time complexity of a given algorithm and plot its growth rate.

Project: Optimizing algorithm performance: Evaluate the time complexity of a specific algorithm, identify areas for improvement, and optimize the algorithm for better performance.

23. **Experiment 23:** Debugging Techniques: Identifying and Fixing Errors

Session 1: Topic: Introduction to debugging techniques

- Understanding common types of errors in programming
- Debugging strategies and tools
- Techniques for identifying and fixing errors

Exercise: Write a Python program with intentional errors and practice debugging it using various techniques.

Project: Debugging a complex program: Analyze and fix bugs in a larger Python program, utilizing different debugging techniques and tools.

24. **Experiment 24:** Data Visualization: Creating Plots and Charts

Session 1: Topic: Data visualization with Python libraries

- Overview of data visualization concepts and techniques
- Creating basic plots and charts using Matplotlib and Seaborn libraries

Exercise: Write a Python program to create line plots and bar charts using Matplotlib and Seaborn libraries.

Project: Visualizing real-world data: Select a dataset of interest and create visually appealing and informative plots and charts using appropriate Python libraries.

25. **Experiment 25:** Project 1: Implementing a Simple Calculator

Session 1: Topic: Design and implementation of a simple calculator

- Defining calculator operations and user interface
- Implementing basic arithmetic operations

Exercise: Design and implement a Python program that functions as a basic calculator, supporting operations such as addition, subtraction, multiplication, and division.

Project: Enhancing the calculator: Extend the functionality of the calculator program by adding additional operations, error handling, and user-friendly features.

26. **Experiment 26:** Project 2: Creating a To-Do List Application

Session 1: Topic: Design and implementation of a to-do list application

- Defining requirements and user interface
- Implementing features such as task creation, deletion, and management

Exercise: Design and implement a Python program that functions as a basic to-do list application, allowing users to add, delete, and manage tasks.

Project: Advanced features for the to-do list application: Add additional functionality to the to-do list application, such as task prioritization, due dates, and notifications.

27. **Experiment 27:** Project 3: Building a Basic Contact Management System

Session 1: Topic: Design and implementation of a contact management system

- Defining contact attributes and functionalities
- Implementing features such as contact creation, search, and retrieval



Exercise: Design and implement a Python program that functions as a basic contact management system, allowing users to create, search, and manage contacts.

Project: Advanced features for the contact management system: Enhance the contact management system by adding features such as contact categorization, sorting, and exporting.

28. **Experiment 28:** Project 4: Analyzing and Visualizing Data using Python

Session 1: Topic: Data analysis and visualization project

- Selecting a dataset for analysis
- Performing data manipulation, analysis, and visualization using Python libraries

Exercise: Select a dataset of interest and perform basic data analysis and visualization using Python libraries such as Pandas and Matplotlib.

Project: Comprehensive data analysis and visualization: Conduct an in-depth analysis of a dataset, apply advanced data manipulation techniques, and create meaningful visualizations to present insights.

29. **Experiment 29:** Final Project: Integrating Concepts into a Comprehensive App

Session 1: Topic: Final project development

- Combining concepts and skills learned throughout the course
- Designing and implementing a comprehensive Python application

Exercise: Identify a problem or application domain and outline the requirements and functionality for a comprehensive Python application.

Project: Development of a comprehensive Python application: Design, implement, and test a Python application that addresses a specific problem or serves a particular purpose, incorporating various programming concepts and techniques.

References:

https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/video_galleries/lecture-videos/

https://onlinecourses.nptel.ac.in/noc21_cs45/preview

<https://www.udemy.com/course/learning-python-for-data-analysis-and-visualization/>

<https://www.mygreatlearning.com/academy/learn-for-free/courses/python-for-data-analysis>

<https://www.coursera.org/learn/data-analysis-with-python>



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	A2	P1
CO2	C3	A3	P2
CO3	C3	A4	P3
CO4	C3	A4	P4
CO5	C6	A5	P4

CO-PO Mapping



PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1	2	-	2	2	2		-	1	-
CO2	1	3	2	-	-	2	2	-	-	1
CO3	2	2	3	3	-	2	-	1	-	-
CO4	-	2	2	3	2	2	-	-	-	-
CO5	-	2	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/4 th IR	Emphasizes the responsible and ethical development and deployment of the systems.



NETWORK DEFENSE ESSENTIAL

Department:	Department of Computer Science and Engineering		
Course Name: Network Defense Essential	Course Code	L-T-P	Credits
	ENSP114	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			
Brief Syllabus:			
<p>Network Defense Essentials covers the fundamental concepts of information security and network defense. This course provides a comprehensive understanding of network defense principles and essential skills required to protect computer networks from various security threats. Overview of key components of information security such as identification, authentication and authorization Introduction to Virtualization, Cloud computing, wireless networks, mobile & IOT devices and Data Security. Discusses network level attacks, password cracking and their countermeasures. The course combines theoretical knowledge with hands-on exercises to equip students with practical skills in network defense.</p>			
UNIT WISE DETAILS			
Unit Number: 1	Title: Network Security Fundamentals	No. of hours: 10	
Content Summary:			
Introduction to information security, CIA triad (Confidentiality, integrity, and availability), network defense challenges, Network security controls, administrative controls: frameworks, laws, acts, and security policies, Physical controls: physical security, workplace security, and environmental control			
Unit Number: 2	Title: Network and Web Application Security	No. of hours: 10	
Content Summary:			
Basics of ethical hacking, Penetration testing methodology, Vulnerability assessment tools and technique, Network reconnaissance and scanning techniques, Denial-of-Service (DoS) attacks and prevention, TCP/IP attacks (e.g., TCP/IP hijacking, SYN flooding) and countermeasures, Web application vulnerabilities (e.g., SQL injection, cross-site scripting)			
Unit Number: 3	Title: Wireless and Mobile Security	No. of hours: 10	
Content Summary:			
Wireless network security and attacks (e.g., WEP/WPA/WPA2 cracking), Mobile device security and attacks (e.g., app vulnerabilities, malware), Password cracking techniques and countermeasures, Social engineering techniques and			



countermeasures, Fundamentals of incident response and handling, Penetration testing concepts and methodologies.

Unit Number: 4	Title: IoT, Virtualization and Cloud Computing Security	No. of hours: 10
---------------------------------	--	-------------------------

Content Summary:
Working of IoT devices, application areas, communication models, IoT security principles, IoT framework security considerations, IoT device management, IoT security threats and countermeasures, OT security threats and countermeasures, Cryptography and PKI, Virtualization concept, Security concerns in virtualization.

- *Self-Learning Components:**
- Encryption algorithms
 - Network Security protocols
 - Wireshark: Network protocol analyzer for packet inspection and analysis
 - Nmap: Network mapping and port scanning tool
 - Research and presentations on emerging network security concepts, such as Software-Defined Networking (SDN), Zero Trust Networking, and Threat Intelligence
 - Network Defense essential certification by EC Council

Reference Books & Links:

- <https://www.eccouncil.org/academia/network-defense-essentials-nde/>
- William Stallings, Lawrie Brown, Computer Security: Principles and Practice, 3rd edition, 2014
- Cryptography and Network security, Behrouz A. Forouzan, Debdeep Mukhopadhyay, Mcgraw Hill Education, 2nd Edition, 2011
- Machine Intelligence and Big Data Analytics for Cybersecurity Applications. Studies in Computational Intelligence, vol 919. Springer, Cham, 2021
- Cryptography and Network Security - Principles and Practice | Seventh Edition | By Pearson by Stallings William

Define Course Outcomes (CO)



COs	Statements
CO1	Understand the fundamental concepts of information security and its importance in network defense.
CO2	Gain knowledge of ethical hacking fundamentals and the role of penetration testing in identifying vulnerabilities.
CO3	Identify information security threats and vulnerabilities through effective vulnerability assessment techniques.
CO4	Implement appropriate countermeasures and defensive strategies to mitigate network-level, web application, wireless, mobile, IoT, OT, and cloud computing attacks.
CO5	Conduct basic penetration tests and Evaluate the security of network systems and 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	A1	P1
CO2	C2	A1	P1
CO3	C2	A2	P2
CO4	C3	A3	P3
CO5	C5	A4	P4

CO-PO Mapping



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

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

CO-PSO Mapping

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



Relevance of the Syllabus to various indicators

Unit I	Network Security Fundamentals
Local	Security policies, physical security, and workplace security contribute to local security requirements.
Regional	-
National	-
Global	Network security controls address global challenges in securing information and preventing cyber threats.
Employability	-
Entrepreneurship	-
Skill Development	Develops skills in implementing network security controls and understanding security policies.
Professional Ethics	Emphasizes the importance of confidentiality, integrity, and availability of information.
Gender	-
Human Values	Promotes values of integrity, privacy, and responsible use of technology
Environment & Sustainability	-
Unit II	Network and Web Application Security
Local	-
Regional	-
National	-
Global	Understanding ethical hacking, penetration testing, and countermeasures contribute to global cybersecurity efforts.
Employability	Knowledge of network and web application security enhances employability in cybersecurity roles.
Entrepreneurship	-
Skill Development	Develops skills in identifying vulnerabilities, conducting penetration testing, and implementing countermeasures.
Professional Ethics	Emphasizes ethical hacking and responsible disclosure of vulnerabilities.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	Wireless and Mobile Security
Local	-
Regional	-
National	-
Global	Contributes to global efforts in securing communication technologies.



Employability	-
Entrepreneurship	-
Skill Development	Develops skills in securing wireless networks, mobile devices, and incident response.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	IoT, Virtualization and Cloud Computing Security
Local	-
Regional	-
National	-
Global	Enhancing IoT and cloud computing security contributes to global cybersecurity strategies.
Employability	Knowledge of IoT, virtualization, and cloud computing security enhances employability in cybersecurity roles, particularly in IoT security.
Entrepreneurship	-
Skill Development	Develops skills in securing IoT devices, understanding virtualization security concerns, and addressing cloud computing security threats.
Professional Ethics	Emphasizes responsible use of IoT devices and adherence to cloud computing security best practices.
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	4,9
NEP 2020	Provides knowledge and skills for information security control
POE/4 th IR	The course aligns with the need for network defense and cybersecurity in the context of the fourth industrial revolution, emphasizing the importance of securing networks and protecting against cyber threats.



NETWORK DEFENSE ESSENTIAL LAB

Department:	Department of Computer Science and Engineering		
Course Name: Network Defense Essential lab	Course Code	L-T-P	Credits
	ENSP166	0-0-2	1
Type of Course:	Minor		
Pre-requisite(s), if any:			

Defined Course Outcomes

COs	
CO 1	Understand the foundational concepts of network defense and information security.
CO 2	Apply ethical hacking and penetration testing techniques for vulnerability assessment.
CO 3	Analyze and evaluate security measures for wireless networks, mobile devices, and virtualization.
CO 4	Design and implement security considerations and countermeasures for IoT and OT devices.

Proposed Lab Experiments

Ex. No	Experiment Title	Mapped CO/COs
1	Analyze different network topologies and their implications for security.	CO1
2	Use packet sniffing tools to capture and analyze network traffic for security purposes.	CO4
3	Perform password cracking techniques to understand the importance of strong passwords and password security	CO2
4	Configure and test different types of firewalls to understand their functionalities and effectiveness in protecting a network	CO1
5	Implement an IDS and evaluate its effectiveness in detecting and preventing network intrusions.	CO2



6	Implement encryption and decryption techniques using various algorithms and analyze their effectiveness in ensuring data confidentiality.	CO3
7	Implement digital signature techniques to authenticate and verify the integrity of digital documents.	CO3
8	Configure and test secure email protocols (e.g., PGP and S/MIME) to ensure secure communication	CO1
9	Assess the security vulnerabilities of a web application and propose mitigation strategies.	CO2
10	Implement database security measures, such as access controls and encryption, to protect sensitive data.	CO1
11	Harden the security of an operating system by implementing security configurations and applying updates and patches.	CO2
12	Assess the security risks in wireless networks and implement security measures to protect against unauthorized access.	CO1
13	Analyze malware samples to understand their behavior and develop strategies for detection and removal.	CO2
14	Simulate and respond to a cybersecurity incident to understand the steps involved in incident response and mitigation	CO4
15	Conduct social engineering simulations to raise awareness about the risks associated with social engineering attacks.	CO3
16	Perform vulnerability scanning using automated tools to identify and remediate security vulnerabilities.	CO2
17	Set up a SIEM solution to collect and analyze security logs from various network devices..	CO1
18	Conduct network forensic analysis to investigate and gather evidence of network security incidents.	CO4
19	Implement secure file transfer protocols (e.g., SFTP) to ensure secure data transmissio	CO3
20	Develop and deliver security awareness training materials to educate users about common security threats and best practices.	CO4
21	Implement security measures on mobile devices, such as device encryption and secure app development practices.	CO3
22	Monitor and analyze the tactics used by both red and blue teams to improve overall network defense.	CO4
23	Develop and implement secure coding practices for web applications, including input validation, output encoding, and secure session management.	CO4



Experiments Descriptions:-

1. **Session:**

- Introduction to Network Security Controls:
- Overview of network security controls and their importance.
- Discussion on different types of controls (firewalls, IDS, ACLs).
- Explanation of how to configure and test firewall rules.

2. **Exercise:**

- Firewall Configuration Exercise:
- Provide students with a network scenario and a set of security requirements.
- Instruct them to configure firewall rules to enforce the desired security policies.
- Test and evaluate the effectiveness of the firewall configuration through simulated attacks.

3. **Session:**

- Vulnerability Assessment and Penetration Testing:
 - Introduction to vulnerability assessment and penetration testing methodologies.
 - Explanation of scanning tools, techniques, and their role in identifying vulnerabilities.
 - Discussion on ethical considerations and legal implications.

4. **Exercise:**

- Vulnerability Scanning and Penetration Testing Exercise:
 - Instruct students to conduct a vulnerability assessment on a provided network or application using scanning tools.
 - Identify and document vulnerabilities found.
 - Perform penetration testing to exploit identified vulnerabilities and gain unauthorized access.
 - Prepare a report summarizing the findings, including recommended countermeasures.

5. **Session:**

- Wireless Network Security:
 - Overview of wireless network security principles and common vulnerabilities.
 - Discussion on encryption protocols (WPA2, WEP) and their weaknesses.
 - Explanation of wireless network penetration testing techniques.

6. **Exercise:**

- Wireless Network Security Configuration Exercise:
 - Guide students in setting up a wireless network with appropriate security measures.
 - Instruct them to configure encryption protocols, SSID hiding, and access control.
 - Conduct a wireless penetration test to identify potential vulnerabilities.



- Provide recommendations to enhance the security of the wireless network.

7. Session:

- Web Application Security:
 - Introduction to common web application vulnerabilities (SQL injection, XSS) and their impact.
 - Explanation of secure coding practices and input validation techniques.
 - Overview of tools for web application vulnerability scanning.

8. Exercise:

- Web Application Vulnerability Assessment Exercise:
 - Provide students with a vulnerable web application or simulated environment.
 - Instruct them to perform a comprehensive assessment to identify web application vulnerabilities.
 - Demonstrate various attack scenarios (e.g., SQL injection, XSS) and their potential consequences.
 - Guide students in implementing security measures to mitigate the identified vulnerabilities.

9. Session:

- Incident Response and Handling:
 - Overview of incident response procedures and best practices.
 - Explanation of incident identification, containment, eradication, and recovery processes.
 - Introduction to digital forensics and evidence collection.

10. Exercise:

- Incident Response Simulation Exercise:
 - Simulate a security incident scenario (e.g., network breach, malware infection).
 - Assign roles to students (incident response team, network administrators).
 - Guide them through the incident response process, including identification, containment, and recovery.
 - Provide hands-on experience in conducting forensics analysis and evidence gathering.

Project:

- **Network Security Enhancement Project:**
 - Assign students a network infrastructure to secure and enhance.
 - Instruct them to perform a comprehensive security assessment, including vulnerability scanning, penetration testing, and analysis of existing security controls.
 - Based on the assessment, guide students in developing and implementing a security enhancement plan.

INTRODUCTION TO DISCRETE STRUCTURES



Department:	Department of Computer Science and Engineering		
Course Name: Introduction to Discrete Structures	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 etc.			
UNIT WISE DETAILS			
Unit Number: 1	Propositional Logics & Relations	No. of hours: 12	
Content Summary: Mathematical Logic: Introduction to Mathematical Thinking , Propositional and Predicate Logic, Propositional Equivalences, Sets, Binary Relation, Equivalence Relation, Logical operations, Conditional Statements, Tautologies, Contradictions, Logical Equivalence, The use of Quantifiers, Normal Forms, Predicates and Quantifiers, Nested Quantifiers, Rules of Inference. Sets and Relations: Set Operations, Representation and Properties of Relations & Functions, Equivalence Relations, Partially Ordering.			
Unit Number: 2	Title: Counting, Mathematical Induction and Discrete Probability	No. of hours: 12	
Content Summary: Basics of Counting, Pigeonhole Principle, Permutations and Combinations, Inclusion-Exclusion Principle, Mathematical Induction, Probability, Bayes' Theorem, Discrete Probability Theory, Discrete Structures in Computing, Counting Principles, Permutations and Combinations, Probability Theory, Discrete Random Variables, Discrete Optimization - Optimization Problems and Algorithms, Linear Programming, Integer Programming, Algebraic Structures - Groups (Definition, Properties, Subgroups, Cyclic Groups), Rings (Definition, Properties, Integral Domains, Fields), Isomorphisms and Homomorphisms, Counting and combinatorics.			
Unit Number: 3	Title: Group Theory & Discrete Probability	No. of hours: 8	



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

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

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

***Self-Learning Components:**

Topics (with book references):

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

Online Certification Courses for Discrete Mathematics (With Links):

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



<https://www.coursera.org/specializations/discrete-mathematics>

NPTEL Lecture Links for Discrete Mathematics (With Links):

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

Reference Books of Discrete Mathematics:

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

E-Books of Discrete Mathematics (with Links):

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



Define Course Outcomes (CO)

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

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A1	P1
CO2	C3	A2	P2
CO3	C3	A5	P5
CO4	C6	A5	P5
CO5	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	-	1	3	1	1	-	-	-
CO3	-	-	-	1	3	2	2	-	-	3
CO4	-	2	-	-	3	2	-	-	-	3
CO5	-	2	-	-	3	2	2	-	-	3

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

CO-PSO Mapping

PSO	PSO1	PSO2	PSO3	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/4 th IR	Aligns with the concepts of Design, Efficiency ,Problem Solving



BASIC OF OPERATING SYSTEM

Department:	Department of Computer Science and Engineering		
Course Name: Basic of Operating System	Course Code	L-T-P	Credits
	ENBC104	3-1-0	4
Type of Course:	MAJOR		
Pre-requisite(s), if any: Basics of programming			
Brief Syllabus: The Operating Systems course is intended as a general introduction to the techniques used to implement operating systems and related kinds of systems software. The topics covered will be functions and structure of operating systems, process management (creation, synchronization, and communication); processor scheduling; deadlock prevention, avoidance, and recovery; main-memory management; virtual memory management (swapping, paging, segmentation and page-replacement algorithms); control of disks and other input/output devices; file-system structure an 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, Dining Philosopher Problem etc. Definition of Deadlocks, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker’s algorithm, Deadlock detection and Recovery.		
*Self-Learning Components: 1. Case study on UNIX and WINDOWS Operating System. 2. Practice of System calls 3. Students can refer the following book as well: Operating Systems: Three Easy Pieces by Remzi H. Arpaci-Dusseau and Andrea C. Arpaci-Dusseau https://pages.cs.wisc.edu/~remzi/OSTEP/ 4. Students can refer the following courses as per the Open-Source University Curriculum <ul style="list-style-type: none">• "Operating system courses" on Udemy.• " Introduction to Operating Systems Specialization" Coursera.• "Introduction to Operating Systems" by Udacity.		
Reference Books: 1. Silberschatz and Galvin, "Operating System Concepts", Pearson 2. Tannenbaum, "Operating Systems", PHI, 4 th Edition. 2. William Stallings, "Operating Systems Internals and Design Principles", PHI 3. HallMadnick, J. Donovan, "Operating Systems", Tata McGraw Hill. 4. W. Tomasi, "Electronic Communication Systems" Pearson Education, 5 th Edition		



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© 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
CO2	C4	A2	P3
CO3	C5	A3	P3
CO4	C5	A3	P4
CO5	C5	A4	P4

***Please Note:**

Map only 1 or 2 Levels in each category. If a higher level is given, no need to mention lower level



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

Please Note:

- Refer to POs while mapping each CO.
- Mark " - " if not applicable
- If attainment of a CO is strongly mapped with a PO, Mark 3
- If attainment of a CO is moderately mapped with a PO, Mark 2
- If attainment of a CO is weakly mapped with a PO, Mark 1

Justification for mapping must be relevant

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

CO-PSO Mapping

PSO	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 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 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	It is important for national educational institutions to offer this course to produce skilled graduates who can contribute to the national workforce.
Global	Fundamental concepts are applicable in various global industries and organizations.
Employability	Concepts are crucial for various roles in software development
Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	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	It can be applied globally in various industries 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/4 th 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 SYSTEM LAB

Department:	Department of Computer Science and Engineering		
Course Name: Basic Of Operating System 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.

Proposed Lab Experiments

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	CO1



	CPU Scheduling algorithms: Round Robin	
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 File allocation strategy like sequential file allocation in Python.	CO5



19	Program to demonstrate the use of signals in Python for process management.	CO1
20	Program to create and manipulate threads in Python.	CO3
21	Program to implement memory management techniques (e.g., paging, segmentation) in Python.	CO5
22	Program to simulate file system operations (e.g., open, read, write, close) in Python.	CO1
23	Program to implement process synchronization using mutex locks in Python.	CO4
24	Program to simulate the working of virtual memory in Python.	CO5
25	Program to simulate disk file management operations (e.g., allocation, deallocation) in Python.	CO5
26	Program to implement file locking mechanisms (e.g., advisory, mandatory) in Python.	CO5
27	Write a Python program to simulate the following file organization techniques Two level directories	CO5
28	Write Python programs to simulate the paging in memory management techniques	CO5
29	Write Python programs to simulate the segmentation in memory management techniques	CO5
30	Write a Python program to simulate the following file organization 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	3-1-0	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 various 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: 8	
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 behavior 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: 12	
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:**
Students should explore Platforms like LeetCode, HackerRank for C++.
Students can refer the following courses as per the Open Source University Curriculum
1. Introduction to C++" and "C++ Programming for C Programmers" offered by edX
"C++ Programming for Beginners," and "Learn Advanced C++ Programming." offered by Udemy

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

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



CO-PSO Mapping

PSO	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.



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: Basics of C programming			

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.

Proposed Lab Experiments

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	CO1



	for dynamic memory allocation	
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. (b)Write a program to reverse a string using pointers.	CO5,CO4



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	3-1-0	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			
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	Implementa 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
CO2	C2	A3	P4
CO3	C3,C4	A4	P3
CO4	C5	A2	P2

***Please Note:**

Map only 1 or 2 Levels in each category. If a higher level is given, no need to mention lower level



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	-	2	-	-
CO4	3	3	3	-	-	3	-	-	-	-

Please Note:

- Refer to POs while mapping each CO.
- Mark “ - ” if not applicable
- If attainment of a CO is strongly mapped with a PO , Mark 3
- If attainment of a CO is moderately mapped with a PO , Mark 2
- If attainment of a CO is weakly mapped with a PO , Mark 1

Justification for mapping must be relevant

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

CO-PSO Mapping (BSC)

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 enhance programming skills and helps in developing efficient algorithms to solve real-world problems.
Professional Ethics	Following best practices in algorithm design and implementation promotes code readability, maintainability, and overall software quality.
Gender	-
Human Values	Understanding sorting and searching algorithms enables students to develop efficient solutions that improve productivity, streamline processes, and positively impact human lives.
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4 th 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: Basics of Computer Programming			

Defined Course Outcomes

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

Proposed Lab 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 for best, worst & best cases. Draw the three cases in a single graph to justify its observed time complexities.	CO4
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
	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.	
	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.	
	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 CRYPTOGRAPHY

Department:	Department of Computer Science and Engineering		
Course Name: Fundamentals of Cryptography	Course Code	L-T-P	Credits
	ENSP207	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			
Brief Syllabus: <p>Cryptography is an indispensable tool for protecting information in computer systems. In this course students will learn the inner workings of cryptographic systems and how to correctly use them in real-world applications. The course begins with a detailed discussion of how two parties who have a shared secret key can communicate securely when a powerful adversary eavesdrops and tampers with traffic. Students will examine many deployed protocols and analyze mistakes in existing systems. The second half of the course discusses public-key techniques that let two parties generate a shared secret key. Throughout the course participants will be exposed to many exciting open problems in the field and work on fun (optional) programming projects. In a second course (Crypto II) we will cover more advanced cryptographic tasks such as zero-knowledge, privacy mechanisms, and other forms of encryption.</p>			
UNIT WISE DETAILS			
Unit Number: 1	Title: Introduction to Cryptography	No. of hours: 10	



Content Summary: Computer Security Concepts – The OSI Security Architecture – Security Attacks – Security Services and Mechanisms – A Model for Network Security – Classical encryption techniques: Substitution techniques, Transposition techniques, Steganography – Foundations of modern cryptography: Perfect security – Information Theory – Product Cryptosystem – Cryptanalysis.		
Unit Number: 2	Title: Cryptographic Security, Pseudo Randomness and Symmetric Key Ciphers	No. of hours: 10
Content Summary: Shannon’s theory, Perfect secrecy, Entropy, Spurious keys and unicity distance; Bit generators, Security of pseudorandom bit generators. Substitution-permutation networks, Data encryption standard (DES), Description and analysis of DES; Advanced encryption standard (AES), Description and analysis of AES; Stream ciphers, Tritium		
Unit Number: 3	Title: Basics of Number Theory and Public-Key Cryptography	No. of hours: 10
Content Summary: Basics of number theory; Introduction to public-key cryptography, RSA cryptosystem, Implementing RSA; Primality testing, Legendre and Jacobi symbols, SolovayStrassen algorithm, MillerRabin algorithm; Square roots modulo n , Factoring algorithms, Pollard $p - 1$ algorithm, Pollard rho algorithm, Dixon’s random squares algorithm, Factoring algorithms in practice; Rabin cryptosystem and its security.		
Unit Number: 4	Title: More on Public-Key Cryptography, Hash Functions and Signature Schemes	No. of hours: 10
Content Summary: Basics of finite fields; ElGamal cryptosystem, Algorithms for the discrete logarithm problem, Shanks’ algorithm, Pollard rho discrete logarithm algorithm,		



PohligHellmanalgorithm; Discrete logarithm algorithms in practice, Security of ElGamal systems, Bit security of discrete logarithms.

Hash functions and data integrity, SHA-3; RSA signature scheme, Security requirements for signature schemes, Signatures and Hash functions, ElGamal signature scheme, Security of ElGamal signature scheme, Certificates.

***Self-Learning Components:**

https://seedsecuritylabs.org/Labs_16.04/Crypto/

<https://cse29-iiith.vlabs.ac.in/List%20of%20experiments.html>

<https://cyberlab.pacific.edu/courses/comp178/resources>

Reference Books:

1. Jeffrey Hoffstein, Jill Pipher & Joseph H. Silverman (2014). An Introduction to Mathematical Cryptography (2nd edition). Springer.
2. Neal Koblitz (1994). A Course in Number Theory and Cryptography (2nd edition). Springer- Verlag.
3. Christof Paar & Jan Pelzl (2014). Understanding Cryptography. Springer.
4. Simon Rubinfeld-Salzedo (2018). Cryptography. Springer.
5. Douglas R. Stinson & Maura B. Paterson (2019). Cryptography Theory and Practice



Define Course Outcomes (CO)

COs	Statements
CO1	Understand network security services and mechanisms.
CO2	Understand and apply Symmetrical and Asymmetrical cryptography.
CO3	Analyze Data integrity, Authentication, Digital Signatures
CO4	Understand various network security applications, IPSec, Firewall, IDS, Web security, Email security, and malicious software etc.

COs Mapping with Levels of Bloom’s taxonomy

CO	Cognitive levels© 1. 1Knowledge 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	A1	P1
CO2	C3	A2	P2
CO3	C3	A3	P3
CO4	C3	A3	P3
CO5	C3	A3	P3

***Please Note: Map only 1 or 2 Levels in each category. If a higher level is given, no need to mention lower level**

CO-PO Mapping



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

Please Note:

- Refer to POs while mapping each CO.
- Mark “-” if not applicable
- If attainment of a CO is strongly mapped with a PO , Mark 3
- If attainment of a CO is moderately mapped with a PO, Mark 2
- If attainment of a CO is weakly mapped with a PO , Mark 1

Justification for mapping must be relevant

1=weakly mapped

2= moderately mapped

3=strongly mapped

CO-PSO Mapping

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



Relevance of the Syllabus to various indicators

Unit I	Introduction to Cryptography
Local	
Regional	
National	
Global	The course in Cryptography is relevant at all levels as secure communication and data protection are essential for individuals, businesses, and governments worldwide.
Employability	Knowledge of Cryptography is highly valuable in the digital age, enhancing employability in various sectors like cyber security, IT, and data protection.
Entrepreneurship	
Skill Development	The course equips students with foundational knowledge in cryptography and prepares them to contribute to secure digital communication and data protection across various sectors and industries.
Professional Ethics	The course emphasizes ethical practices in handling sensitive information and respecting privacy rights.
Gender	
Human Values	
Environment & Sustainability	
Unit II	Cryptographic Security, Pseudo Randomness and Symmetric Key Ciphers
Local	
Regional	
National	



Global	As societies become more reliant on digital communication, understanding cryptography becomes essential for protecting sensitive information, ensuring cybersecurity, and fostering trust in online transactions and data exchange at global level.
Employability	By studying this course, students can enhance their employability in industries like IT, cyber security, and government agencies that require data protection and secure communication.
Entrepreneurship	
Skill Development	The course enhances equips students with in-demand skills in cyber security, data protection, and cryptography.
Professional Ethics	
Gender	
Human Values	
Environment & Sustainability	
Unit III	Basics of Number Theory and Public-Key Cryptography
Local	
Regional	
National	
Global	As the world becomes increasingly interconnected, the need for secure communication and data protection is paramount at global levels.
Employability	Knowledge of cryptography can open up opportunities for individuals to work in challenging and well-paying roles.



Entrepreneurship	The course enhances employability and entrepreneurship prospects as it provides learners with specialized skills in cryptography, a field that is in high demand in various sectors, including IT, cybersecurity, finance, and government agencies.
Skill Development	The course equips learners with essential knowledge and skills related to information security and encryption, which are crucial for protecting sensitive data and communication
Professional Ethics	The course addresses professional ethics by emphasizing the responsible use of cryptography to ensure data privacy and protect against cyber threats. It encourages the adoption of ethical practices in handling sensitive information.
Gender	
Human Values	
Environment & Sustainability	Cryptography can also contribute to environmental sustainability by securing online transactions, reducing paper-based processes, and promoting a digital economy.
Unit IV	More on Public-Key Cryptography, Hash Functions and Signature Schemes
Local	
Regional	
National	
Global	It addresses concerns related to data protection, privacy, and secure transactions, making it relevant at the global levels.
Employability	Students can pursue careers as security analysts, cryptographers, or cyber security specialists, contributing to the growth of the digital economy.



Entrepreneurship	
Skill Development	Understanding cryptographic principles and techniques develops critical skills to address digital security challenges.
Professional Ethics	Cryptography has ethical implications related to data privacy, confidentiality, and responsible use of technology.
Gender	
Human Values	
Environment & Sustainability	
SDG	4,9
NEP 2020	It aligns with the NEP's focus on developing practical skills and preparing students for employability.
POE/4 th IR	The course on Cryptography is highly relevant as it addresses the critical need for safeguarding data and maintaining privacy in a digitally connected world



FUNDAMENTALS OF CRYPTOGRAPHY LAB

Department:	Department of Computer Science and Engineering			
Course Name: Fundamentals of Cryptography Lab	Course Code	L-T-P	Credits	
	ENSP259	4-0-0	4	
Type of Course:	Minor			
Pre-requisite(s), if any:				

Defined Course Outcomes

Cos	
CO 1	Students will be able to implement various classical ciphers like Substitution Cipher, Baconian Cipher, Playfair Cipher, and Caesar Cipher.
CO 2	Students will learn to implement both symmetric and asymmetric cryptographic algorithms.
CO 3	Through practical exercises involving encryption and decryption of messages, files, and text between clients and servers, students will learn the application of cryptographic algorithms to ensure data confidentiality, integrity, and authentication.
CO 4	Students will work on a mini project involving Hill Cipher, wherein they will apply their knowledge of encryption and decryption to design a complete cryptographic solution.



Proposed Lab Experiments

Ex. No	Experiment Title	Mapped CO/COs
1	Write a program to perform encryption and decryption using Substitution Cipher	CO1
2	Write a program to perform encryption and decryption using Baconian Cipher	CO1
3	Write a program to perform encryption and decryption using Vigenere Cipher	CO1
4	Write a program to perform encryption and decryption using Play Fair Cipher	CO1
5	Write a program to perform encryption and decryption using Transposition cipher	CO1
6	Write a program to perform encryption and decryption using hill cipher	CO2
7	Write a program to depict the implementation of Diffie-Hellman key exchange algorithm	CO2
8	Write a program to perform encryption and decryption using Rail Fence Cipher.	CO2
9	Write a program to perform message transfer between client and server	CO2
10	Write a program to perform message transfer between client and multiple servers	CO2
11	Write a program to perform encryption and decryption using RSA Cipher	CO3
12	WAP to implement Client Server to send and receive text file from client and server.	CO3
13	Write a program to perform encryption and decryption using CAESAR CIPHER	CO3



14	Program for the Implementation of One time Pad.	CO3
15	Write a program to perform encryption and decryption using Transposition Cipher.	CO3
16	Write a program to perform encryption and decryption using Double Transposition Cipher.	CO4
17	Write a program to perform encryption and decryption using Stream Cipher RC4.	CO4
18	Write a program to perform encryption and decryption using RSA Algorithm.	CO4
19	Write a program to perform encryption and decryption using El-Gamal Cryptographic Algorithm	CO4
20	Write a program to perform encryption and decryption using Hill Cipher. (Mini Project)	CO4



BASICS OF PROBABILITY & STATISTICS

Department:	Department of Computer Science and Engineering		
Course Name: Basics of Probability & 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: <ul style="list-style-type: none">• Probability Simulation: Students can explore and 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. https://www.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		



Reference Books:

1. "Probability and Statistics for Data Science" by Reza Hassanzadeh
2. "Statistics for Data Science" by James D. Miller
3. "Statistics for Data Scientists: 50 Essential Concepts" by Peter Bruce and Andrew Bruce
4. "Statistical Methods for Data Science" by Wenqing Li and Yili Hong
5. "Introduction to Probability and Statistics for Data Science" by Samuel N. Cohen

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	-	-	-
CO2	-	-	-
CO3	-	A4	-
CO4	C4	A4	P4
CO5	C5	A5	P5



***Please Note:**

Map only 1 or 2 Levels in each category. If a higher level is given, no need to mention lower level

CO-PO Mapping

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

Please Note:

- Refer to POs while mapping each CO.
- Mark “ - ” if not applicable
- If attainment of a CO is strongly mapped with a PO , Mark 3
- If attainment of a CO is moderately mapped with a PO , Mark 2
- If attainment of a CO is weakly mapped with a PO , Mark 1

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	-
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
	ENBC205	3-1-0	4
Type of Course:	Major		
Pre-requisite(s), if any: C++ Programming			
Frequency of offering (check one): Odd semester			
9. 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.			
11. 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:**

Students should explore Platforms like LeetCode, HackerRank for JAVA and JAVA IDE like eclipse, Netbeans etc.

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

1. "Java Programming Masterclass for Software Developers" on Udemy by Tim Buchalka
2. "Java Fundamentals: The Java Language" on Pluralsight by Jesse Liberty,

Reference Books:

1. Herbert Schildt, –Java – The Complete Referencell, Oracle Press.
2. Cay S. Horstmann, –Core Java Volume – I Fundamentalsll, 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
CO2	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	1	2	-	-	-
CO2	1	2	-	-	3	2	1	-	-	-
CO3	-	-	-	-	3	1	2	1	-	3
CO4	-	-	-	-	3	2	2	-	-	3

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



CO-PSO Mapping

PSO	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 4
NEP 2020	-
POE/4 th 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: C++ Programming			
Frequency of offering (check one): Odd semester			

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.

Proposed Lab 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.	CO1



	e. WAP to find maximum of 3 integer numbers taken as input from user at runtime.	
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	Credits
	AEC011	3-0-1	3
Type of Course:	AEC		
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.			
Total lecture, Tutorial and Practical Hours for this course:			
	Practice		



Lectures: 30	Doubt session:7	PC Lab: 8
11. Course Outcomes (COs)		
Usefulness of this course after its completion i.e. the learner should build self confidence in their communication abilities the learner should exhibit improved communication skills assesses in communication skills, logical reasoning, quantitative skills along with job-specific domain skills.		
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	
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: 7



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: 7
-----------------------	---------------------------------------	------------------------

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.

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
Communication skills by G.H. Hook



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
	ENCS202	3-1-0	4
Type of Course:	Major		
Pre-requisite(s), if any: - Data Structure			
Brief Syllabus: The analysis and design of algorithm course introduce students to the design of computer algorithms, as well as analysis of sophisticated algorithms. Students will learn how to 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: 10	
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: 10	
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: 10	



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: 10
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
CO2	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	-	-	-	2	1	-	1	1
CO2	-	3		3	2	2	-	2	-	-
CO3	-	-	3	-	-	2	2	1	-	2
CO4	-	-	-	-	2	2	-	-	2	-

Please Note:

- Refer to POs while mapping each CO.
- Mark “-”, not applicable
- If attainment of a CO is strongly mapped with a PO, Mark 3
- If attainment of a CO is moderately mapped with a PO, Mark 2
- If attainment of a CO is weakly mapped with a PO, Mark 1

Justification for mapping must be relevant

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



CO-PSO Mapping

PSO	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/4 th IR	Automation and Efficiency and Data Analysis.



FUNDAMENTALS OF ALGORITHM DESIGN & ANALYSIS LAB

Department:	Department of Computer Science and Engineering		
Course Name: Fundamentals of Algorithm Design & Analysis Lab	Course Code	L-T-P	Credits
	ENCS254	0-0-2	1
Type of Course:	Programme Core: Major		

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.

Proposed Lab Experiments

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:	Course Code	L-T-P	Credits
Introduction to Database Management Systems	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
CO2	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	-	3	2	1	2	-	1
CO2	2	-	2	2	2	1	1	-	2	-
CO3	-	2	-	2	2	1	1	2	-	-
CO4	-	2	3	-	2	1	1	-	2	1
CO5	2	2	2	2	3	1	1	2	-	-

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/4 th 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:	Course Code	L-T-P	Credits
Introduction to Database Management Systems Lab	ENBC252	0-0-2	1
Type of Course:	Major		

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

Proposed Lab 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: 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	CO3, CO4
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—	CO3, CO4



	<ul style="list-style-type: none">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 customervii. 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	
5	<p>Database Schema for a Student Library scenario</p> <p>Student(Stud no : integer, Stud_name: string)</p> <p>Membership(Mem no: integer, Stud_no: integer)</p> <p>Book(book no: integer, book_name: string, author: string)</p> <p>Iss_rec(iss no: integer, iss_date: date, Mem_no: integer, book_no: integer)</p> <p>For the above schema, perform the following—</p> <ul style="list-style-type: none">i. Create the tables with the appropriate integrity constraintsii. Insert around 10 records in each of the tablesiii. List all the student names with their membership numbersiv. List all the issues for the current date with student and Book names	CO3, CO4



	<p>v. List the details of students who borrowed book whose author is CJDATE</p> <p>vi. Give a count of how many books have been bought by each student</p> <p>vii. Give a list of books taken by student with stud_no as 5</p> <p>viii. List the book details which are issued as of today</p> <p>ix. Create a view which lists out the iss_no, iss_date, stud_name, book name</p> <p>x. Create a view which lists the daily issues-date wise for the last one week</p>	
6	<p>Database Schema for a Employee-pay scenario</p> <p>employee(emp_id : integer, emp_name: string)</p> <p>department(dept_id: integer, dept_name: string)</p> <p>paydetails(emp_id : integer, dept_id: integer, basic: integer, deductions: integer, additions: integer, DOJ: date)</p> <p>payroll(emp_id : integer, pay_date: date)</p> <p>For the above schema, perform the following—</p> <p>i. Create the tables with the appropriate integrity constraints</p> <p>ii. Insert around 10 records in each of the tables</p> <p>iii. List the employee details department wise</p> <p>iv. List all the employee names who joined after particular date</p> <p>v. List the details of employees whose basic salary is between 10,000 and 20,000</p> <p>vi. Give a count of how many employees are working in each department</p> <p>vii. Give a names of the employees whose netsalary > 10,000</p> <p>viii. List the details for an employee_id=5</p> <p>ix. Create a view which lists out the emp_name, department,</p>	CO3, CO4



	basic, deductions, netsalary x.Create a view which lists the emp_name and his netsalary	
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 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	CO3, CO4
8	Database Schema for a student-Lab scenario Student(stud_no: integer , stud_name: string, class: string) Class(class: string,descrip: string) Lab(mach_no: integer , Lab_no: integer, description: String)	CO3, CO4



	<p>Allotment(Stud_no: Integer, mach_no: integer, dayofweek: string)</p> <p>For the above schema, perform the following—</p> <ul style="list-style-type: none"> 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". 									
9	<p>Consider the following table:</p> <p style="text-align: center;">Table: CLASS</p> <table border="1" data-bbox="305 1396 1118 1617"> <thead> <tr> <th>Id</th> <th>Name</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Bravo</td> </tr> <tr> <td>2</td> <td>Alex</td> </tr> <tr> <td>4</td> <td>Cheng</td> </tr> </tbody> </table> <p>Give the output of the following SQL script:</p> <ul style="list-style-type: none"> > INSERT INTO class VALUES (5,'Rahul'); > COMMIT; > UPDATE class SET name = 'Abhijeet' WHERE id= '5'; > SAVEPOINT A; 	Id	Name	1	Bravo	2	Alex	4	Cheng	CO3, CO4
Id	Name									
1	Bravo									
2	Alex									
4	Cheng									



	<ul style="list-style-type: none"> > INSERT INTO class VALUES (6, 'Chris'); > SAVEPOINT B; > INSERT INTO class VALUES (7, 'Bravo'); > SAVEPOINT C > SELECT * FROM class; > ROLLBACK TO B; > SELECT * FROM class; > ROLLBACK TO A; 																																																															
10	<p>Consider the following two tables: SHOP and ACCESSORIES</p> <p style="text-align: center;">Table: SHOP</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 10%;">ID</th> <th style="width: 60%;">ShopName</th> <th style="width: 30%;">Area</th> </tr> </thead> <tbody> <tr> <td>S01</td> <td>ABC Computronics</td> <td>CP</td> </tr> <tr> <td>S02</td> <td>All Infotech Media</td> <td>GK II</td> </tr> <tr> <td>S03</td> <td>Tech Shoppe</td> <td>CP</td> </tr> <tr> <td>S04</td> <td>Geek Tenco Soft</td> <td>Nehru Place</td> </tr> <tr> <td>S05</td> <td>Hitech Tech Store</td> <td>Nehru Place</td> </tr> </tbody> </table> <p style="text-align: center;">Table: ACCESSORIES</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 15%;">No</th> <th style="width: 40%;">Name</th> <th style="width: 20%;">Price</th> <th style="width: 25%;">Id</th> </tr> </thead> <tbody> <tr> <td>A01</td> <td>Motherboard</td> <td>12000</td> <td>S01</td> </tr> <tr> <td>A02</td> <td>Hard Disk</td> <td>5000</td> <td>S01</td> </tr> <tr> <td>A03</td> <td>Keyboard</td> <td>500</td> <td>S02</td> </tr> <tr> <td>A04</td> <td>Mouse</td> <td>300</td> <td>S01</td> </tr> <tr> <td>A05</td> <td>Motherboard</td> <td>13000</td> <td>S02</td> </tr> <tr> <td>A06</td> <td>Keyboard</td> <td>400</td> <td>S03</td> </tr> <tr> <td>A07</td> <td>LCD</td> <td>6000</td> <td>S04</td> </tr> <tr> <td>T08</td> <td>LCD</td> <td>5500</td> <td>S05</td> </tr> <tr> <td>T09</td> <td>Mouse</td> <td>350</td> <td>S05</td> </tr> <tr> <td>T10</td> <td>Hard Disk</td> <td>450</td> <td>S03</td> </tr> </tbody> </table> <p>i.Perform Cartesian product or Cross join of both tables.</p>	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	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	CO3, CO4
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																																																														
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																																																													



	<p>ii.To display the Name and Price of all the Accessories in ascending order of their price.</p> <p>iii.To display ID and ShopName of all shops located in Nehru Place.</p> <p>iv.To display minimum and maximum price of all accessories.</p> <p>v.To display Name, Price of all accessories and their respective ShopName where they are available.</p>																																											
11	<p>In continuation with experiment no. 10, find the output of the following SQL queries based on above mentioned tables:</p> <p>i.SELECT DISCTINCT NAME FROM ACCESSORIES WHERE PRICE >= 5000;</p> <p>ii.SELECT AREA, COUNT(*) FROM SHOP GROUP BY AREA;</p> <p>iii.SELECT COUNT(DISTINCT AREA) FROM SHOP;</p> <p>iv.SELECT NAME, PRICE*0.05 DISCOUNT FROM ACCESSORIES WHERE ID IN ('S02', 'S03');</p>	CO3, CO4																																										
12	<p>Consider the following two tables: PRODUCT and CLIENT.</p> <p style="text-align: center;">Table: Product</p> <table border="1" data-bbox="344 1186 1193 1570"> <thead> <tr> <th>P_ID</th> <th>ProdName</th> <th>Manufacturer</th> <th>Price</th> <th>ExpiryDate</th> </tr> </thead> <tbody> <tr> <td>TP01</td> <td>Talcom Powder</td> <td>LAK</td> <td>40</td> <td>2011-06-26</td> </tr> <tr> <td>FW05</td> <td>Face Wash</td> <td>ABC</td> <td>45</td> <td>2010-12-01</td> </tr> <tr> <td>BS01</td> <td>Bath Soap</td> <td>ABC</td> <td>55</td> <td>2010-09-10</td> </tr> <tr> <td>SH06</td> <td>Shampoo</td> <td>XYZ</td> <td>120</td> <td>2012-04-09</td> </tr> <tr> <td>FW12</td> <td>Face Wash</td> <td>XYZ</td> <td>95</td> <td>2010-08-15</td> </tr> </tbody> </table> <p>Note:</p> <ul style="list-style-type: none"> P_ID is the primary key. <p>Table: Client</p> <table border="1" data-bbox="399 1730 1214 1894"> <thead> <tr> <th>C_ID</th> <th>ClientName</th> <th>City</th> <th>P_ID</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Cosmetic Shop</td> <td>Delhi</td> <td>FW05</td> </tr> <tr> <td>6</td> <td>Total Health</td> <td>Mumbai</td> <td>BS01</td> </tr> </tbody> </table>	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	C_ID	ClientName	City	P_ID	1	Cosmetic Shop	Delhi	FW05	6	Total Health	Mumbai	BS01	CO3, CO4
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																																								
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	<p>Consider the following schema for a Library Database:</p> <p>BOOK(Book_id, Title, Publisher_Name, Pub_Year)</p> <p>BOOK_AUTHORS(Book_id, Author_Name) PUBLISHER(Name, Address, Phone)</p> <p>BOOK_COPIES(Book_id, Programme_id, No-of_Copies)</p> <p>BOOK_LENDING(Book_id, Programme_id, Card_No, Date_Out, Due_Date)</p> <p>LIBRARY_PROGRAMME(Programme_id, Programme_Name, Address)</p> <p>Write SQL queries to</p> <p>1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each Programme, etc.</p>	CO3, CO4
----	---	----------



	<p>2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017.</p> <p>3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation.</p> <p>4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.</p> <p>5. Create a view of all books and its number of copies that are currently available in the Library</p>	
14	<p>Consider the following schema for Order Database:</p> <p>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)</p> <p>Write SQL queries to</p> <ol style="list-style-type: none">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	<p>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)</p> <p>Write SQL queries to</p> <ol style="list-style-type: none">1. List the titles of all movies directed by 'Hitchcock'.	CO3, CO4



	<p>2. Find the movie names where one or more actors acted in two or more movies.</p> <p>3. List all actors who acted in a movie before 2000 and in a movie after 2015 (use JOIN operation).</p> <p>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.</p> <p>5. Update rating of all movies directed by 'Steven Spielberg' to 5.</p>	
16	<p>Consider the schema for College Database: STUDENT(USN, SName, Address, Phone, Gender) SEMSEC(SSID, Sem, Sec) CLASS(USN, SSID) COURSE(Subcode, Title, Sem, Credits) IAMARKS(USN, Subcode, SSID, Test1, Test2, Test3, FinalIA)</p> <p>Write SQL queries to</p> <ol style="list-style-type: none">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.	CO3, CO4
17	<p>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)</p>	CO3, CO4



<p>WORKS_ON(SSN, PNo, Hours)</p> <p>Write SQL queries to</p> <ol style="list-style-type: none">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.	
---	--

INTRODUCTION TO COMPUTER NETWORKS

Department:	Department of Computer Science and Engineering
--------------------	---



Course Name: Introduction to Computer Networks	Course Code	L-T-P	Credits
	ENBC206	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: 12
Content Summary: Application Layer: Domain Name Space (DNS), DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls, Basic concepts of Cryptography.		
*Self-Learning Components: 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 <ul style="list-style-type: none">• 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: <ol style="list-style-type: none">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©	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	C1		



C02	C2		P2
C03	C3	A2	
C04	C6		P3
C05			

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

PSO	PSO1	PSO2	PSO3	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:	Course Code	L-T-P	Credits
Introduction to Computer Networks	ENBC256	0-0-2	1



Lab			
Type of Course:	Major		
Pre-requisite(s), if any:			

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.

Proposed Lab 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	CO2,CO3



	for accessing student activity data in the institute.	
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:	Course Code	L-T-P	Credits
Career Development Program-II	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
C01	C2		P1
C02	C3		P2
C03	C3		P3
C04	C1		-
C05	C1		P1

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 (Provided at the [end of Handbook](#)). File needs to be submitted in spiral bind.
- Project will be evaluated on the scale of 100 with following evaluation criteria.
 - Project idea & features (10)
 - Literature review (10)
 - Tools & Techniques employed (10)
 - Methodology (10)
 - Presentation of Results and its usefulness (20)
 - Implementation and its understandability (10)
 - Meetings & comments by guide (20)
 - Research paper (10)

File format for Minor project

1.	Abstract	Page No.
2.	Introduction (description of broad topic)	
3.	Motivation	
4.	Literature Review	
5.	Gap Analysis	
6.	Problem Statement	
7.	Objectives	
8.	Tools/platform used	
9.	Methodology	
10.	Experimental Setup	
11.	Evaluation Metrics	
12.	Results And Discussion	
13.	Conclusion & Future Work	
14.	References	



15.	Annexure I: Responsibility Chart	
16.	Annexure II: Screenshots of all the MS-Team Meetings with links (online)/ handwritten comments(offline) from guide	
17.	Annexure III Complete implementation code	
18.	Annexure IV Research Paper (Published/Submitted)	

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			

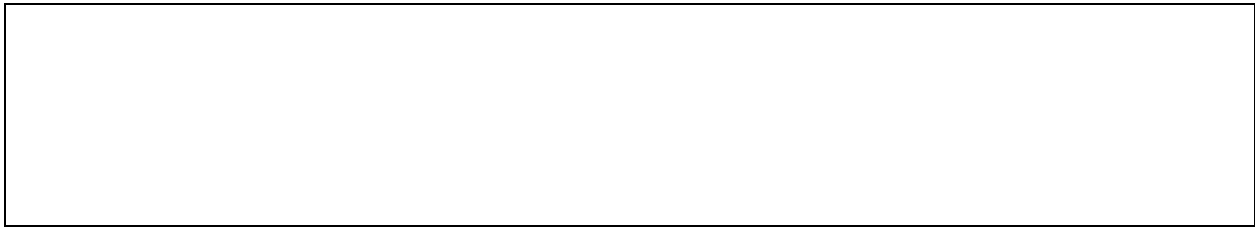


Table of Contents

S.N	Experiment Index	COs
1	Introduction to Competitive Coding <ul style="list-style-type: none">• 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 <ul style="list-style-type: none">• 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 <ul style="list-style-type: none">• Understanding time and space complexity of algorithms.• Analysis of algorithm efficiency and choosing the most optimal solutions.	CO2
4	Problem Solving Techniques <ul style="list-style-type: none">• 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 <ul style="list-style-type: none">• 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 <ul style="list-style-type: none">• 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
	Online Judges and Contest Platforms <ul style="list-style-type: none">• Familiarization with popular online judge platforms like	



7	<p>Codeforces, Topcoder, and LeetCode.</p> <ul style="list-style-type: none">Practice solving problems from online contests and participating in coding competitions. <p>List of suggested links to coding platforms</p> <ul style="list-style-type: none">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/codejamKick Start by Google: https://codingcompetitions.withgoogle.com/kickstartACM ICPC Live Archive: https://icpcarchive.ecs.baylor.edu/A2 Online Judge: https://a2oj.com/CodeSignal: https://codesignal.com/	CO5
8	<p>Tips and Tricks for Competitive Coding</p> <ul style="list-style-type: none">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	<p>Mock Contests and Practice Sessions</p> <ul style="list-style-type: none">Conducting mock contests and practice sessions to simulate real coding competitions.Solving a wide range of problems to enhance coding skills and adaptability to different problem types.	CO5
10	<p>Self-Learning Component:</p> <p>List of Suggested Competitive programming Courses:</p> <ul style="list-style-type: none">"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:	CO5



<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. Reville: 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 n th 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:	Course Code	L-T-P	Credits



Computer Organization & Architecture	ENBC301	4-0-0	4
Type of Course:	Major		
Pre-requisite(s), if any: Concepts of Digital Electronics			

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

UNIT WISE DETAILS

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

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

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

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



Unit Number: 3	Title: The Processor	No. of hours: 10
Content Summary: Revisiting clocking methodology, Amdahl’s law, Building a data path and control, single cycle processor, multi-cycle processor, instruction pipelining, Notion of ILP, data and control hazards and their mitigations.		
Unit Number: 4	Title: Memory hierarchy, Storage and I/O	No. of hours: 10
Content Summary: SRAM/DRAM, locality of reference, Caching: different indexing mechanisms, Trade-offs related to block size, associativity, and cache size, Processor-cache interactions for a read/write request, basic optimizations like writethrough/write-back caches, Average memory access time, Cache replacement policies (LRU), Memory interleaving. Introduction to magnetic disks (notion of tracks, sectors), flash memory. I/O mapped, and memory mapped I/O. I/O data transfer techniques: programmed I/O, Interrupt-driven I/O, and DMA.		
*Self-Learning Components: 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. RIZES: <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
CO2	C3	A2	P1
CO3	C5	A2	P2
CO4	C6	A3	P3
CO5	C6	A4	P4

CO-PO Mapping

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

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/4 th IR	Aligns with the concepts of parallel computing, advanced processors, and memory architectures.

ESSENTIALS OF ETHICAL HACKING

Department:	Department of Computer Science and Engineering		
Course Name: Essentials of Ethical hacking	Course Code	L-T-P	Credits
	ENSP319	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			



Brief Syllabus:

The subject "Essentials of Ethical Hacking" covers an introduction to ethical hacking, including its concept, objectives, and importance in cybersecurity. It explores different types of hacker attacks, such as Footprinting, Reconnaissance, and Network Scanning, along with legal and ethical considerations. The course delves into enumeration techniques, including active and passive enumeration, sniffing, and social engineering, and their countermeasures. It further covers password cracking methods, malware threats, system hacking vulnerabilities, and network security principles, including firewalls, IDS/IPS, VPNs, wireless security, and web application security.

UNIT WISE DETAILS

Unit Number: 1	Title: Introduction to Ethical Hacking	No. of hours: 10
--------------------------	---	-------------------------

Content Summary:

Introduction to Ethical Hacking: Understanding the concept, objectives, and importance of ethical hacking in cybersecurity; different types of hacker attacks, Foot printing and Reconnaissance, Scanning Networks, TCP flag types, types of port scans, scanning countermeasures; Legal and Ethical Considerations: Overview of laws, regulations, and ethical guidelines governing ethical hacking practices.

Unit Number: 2	Title: Enumeration	No. of hours: 10
--------------------------	---------------------------	-------------------------

Content Summary:

Enumeration: Role and enumeration techniques recognize how to establish a sessions, Identify enumeration countermeasures, Perform active and passive enumeration. Sniffers, types of sniffing and protocols vulnerable to sniffing, Recognize types of sniffing attacks, methods for detecting sniffing, social engineering: Different types of social engineering, and social engineering countermeasures

Unit Number: 3	Title: System Hacking and Vulnerability Analysis	No. of hours: 10
--------------------------	---	-------------------------

Content Summary:

Password Cracking: Identify different types of password attacks, Use a password cracking tool, Identify various password cracking countermeasures; Malware Threats: Overview of various types of malware, including viruses, worms, Trojans, and ransomware; System Hacking: Exploiting vulnerabilities to gain unauthorized access to systems, Identify different ways to hide files, Recognize how to detect a rootkit, Identify tools that can be used to cover attacker tracks

Unit Number: 4	Title: Network Security	No. of hours: 10
--------------------------	--------------------------------	-------------------------

Content Summary:

Network Security Principles: Understanding network security concepts, including firewalls, IDS/IPS, and VPNs; Wireless Network Security: Securing wireless networks and mitigating potential risks; Web Application Security: Identifying and



addressing security vulnerabilities in web applications.

***Self-Learning Components:**

- Learning to use ethical hacking tools: Metasploit, Wireshark, Nmap, Burp Suite.
- Exploring rootkit detection tools and techniques.
- Understanding wireless security protocols (WPA2, WPA3) and encryption.
- Researching social engineering attack vectors and case studies.
- Reading articles on recent cybersecurity breaches and their impact.

Reference Books:

- "Ethical Hacking and Penetration Testing Guide" by Rafay Baloch
- "The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws" by Dafydd Stuttard and Marcus Pinto

Reference Links:

- OWASP (Open Web Application Security Project): <https://owasp.org/>
- NIST (National Institute of Standards and Technology) Cybersecurity Framework: <https://www.nist.gov/cyberframework>

Define Course Outcomes (CO)

COs	Statements
CO1	Recall and understand the principles and objectives of ethical hacking and the different types of hacker attacks.
CO2	Apply enumeration techniques, scanning methods, and countermeasures in network security assessments.
CO3	Analyze and evaluate different password cracking and malware threat techniques and implement countermeasures.
CO4	Synthesize and propose network security solutions, wireless network security measures, and web application security enhancements.
CO5	Design and create system hacking strategies to exploit vulnerabilities and detect rootkits.

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
CO2	C3	A2	P2
CO3	C3	A3	P3
CO4	C6	A4	-
CO5	C6	A5	P1

CO-PO Mapping

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

Please Note:

- Refer to POs while mapping each CO.
- Mark “ - ” if not applicable
- If attainment of a CO is strongly mapped with a PO , Mark 3
- If attainment of a CO is moderately mapped with a PO , Mark 2
- If attainment of a CO is weakly mapped with a PO , Mark 1

CO-PSO Mapping

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



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

Relevance of the Syllabus to various indicators

Unit I	Introduction to Ethical Hacking
Local	-
Regional	-
National	In the context of national security, having professionals well-versed in ethical hacking is essential to safeguard critical infrastructure and sensitive government data from cyber attacks.
Global	With the increasing global interconnectedness, cybersecurity is a shared concern. Ethical hacking helps in addressing cyber threats that can have a global impact on organizations and individuals.
Employability	Individuals trained in ethical hacking have increased employability prospects as organizations seek cybersecurity professionals to protect their systems from cyber threats.
Entrepreneurship	-
Skill Development	Technical skills, critical thinking, and problem-solving abilities related to cybersecurity and ethical hacking.
Professional Ethics	Understanding the legal and ethical aspects of ethical hacking promotes responsible and ethical computing practices.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	Enumeration
Local	-
Regional	-
National	-
Global	Enumeration is a universal cybersecurity practice applicable to organizations worldwide, contributing to the global efforts to combat cyber threats.
Employability	Individuals proficient in enumeration techniques are sought after by organizations to identify network vulnerabilities and strengthen their cybersecurity infrastructure.
Entrepreneurship	Enumeration skills can be applied to offer cybersecurity auditing and vulnerability assessment services to businesses and organizations.
Skill Development	Students develop skills in assessing network security, identifying weaknesses, and proposing countermeasures through enumeration techniques.



Professional Ethics	Ethical consideration is emphasized when using enumeration for cybersecurity assessments.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	System Hacking and Vulnerability Analysis
Local	-
Regional	-
National	-
Global	Cyber threats can transcend borders, and a global understanding of system hacking is essential to collectively address and respond to cyber incidents worldwide.
Employability	Knowledge of system hacking and vulnerability analysis increases employability in cybersecurity roles responsible for securing systems and detecting potential threats.
Entrepreneurship	-
Skill Development	Students acquire skills in identifying and addressing system vulnerabilities, ensuring secure system configurations.
Professional Ethics	Understanding the ethical implications of system hacking is crucial to maintain integrity in cybersecurity practices.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	Network Security
Local	Local businesses can enhance their network security using principles like firewalls, IDS/IPS, and VPNs to protect sensitive data and communications.
Regional	-
National	-
Global	Network security practices are vital for securing international communications and protecting global networks.
Employability	-
Entrepreneurship	-
Skill Development	Enhances students' abilities to implement network security measures, including firewalls and VPNs, to safeguard data transmission.
Professional Ethics	Practicing network security principles aligns with ethical responsibilities to protect user data and maintain network integrity.
Gender	-
Human Values	-
Environment &	-



Sustainability	
SDG	SDG 9
NEP 2020	Emphasis on skill development and vocational training
POE/4 th IR	Emphasize responsible and legal use of hacking techniques for defensive purposes. Ethical hacking skills are essential to protect the interconnected systems and data in the 4th IR era.

ESSENTIALS OF ETHICAL HACKING LAB

Department:	Department of Computer Science and Engineering
--------------------	---



Course Name: Essentials of Ethical hacking Lab	Course Code	L-T-P	Credits
	ENSP367	0-0-2	1
Type of Course:	Minor		

Defined Course Outcomes

COs	
CO 1	Comprehend the fundamental principles and theories of ethical hacking and how they apply to different scenarios.
CO 2	Apply ethical hacking techniques, tools, and methodologies to assess and identify vulnerabilities in computer systems and networks.
CO 3	Analyze the results of ethical hacking activities, identify security weaknesses, and evaluate the impact of potential vulnerabilities.
CO 4	Evaluate the effectiveness of countermeasures and create defensive strategies to mitigate security risks and vulnerabilities.

Proposed Lab Experiments

Ex. No	Experiment Title	Mapped CO/COs
1	Learn how to use the latest techniques and tools to perform footprinting and reconnaissance. Use Google and Whois for Reconnaissance	CO1
2	Perform WHOIS lookup and identify the domain registrar and other relevant information.	CO1
3	Conduct DNS enumeration to discover subdomains associated with the target domain	CO1
4	Use Nmap or similar tools to perform port scanning on a target system and identify open ports and services.	CO2
5	Using TraceRoute, ping, ifconfig, netstat Command	CO2
6	Conduct a network scan using tools like Nmap to identify hosts, services, and vulnerabilities	CO2
7	Case study on Honeypots	CO2
8	Use CryptTool to encrypt and decrypt passwords using RC4 algorithm	CO3
9	Use password cracking tools like John the Ripper or Hashcat to crack passwords from a given set of encrypted passwords.	CO3
10	Implement dictionary attacks and brute force attacks to crack passwords of test accounts.	CO3
11	Perform SQL injection attacks on a vulnerable web application to extract sensitive data	CO3



12	Use WireShark sniffer to capture network traffic and analyze packets to identify potential security issues.	CO3
13	Identify and analyze suspicious network traffic, such as port scanning or unauthorized access attempts.	CO3
14	Exploit known vulnerabilities in a target system using Metasploit or other exploitation frameworks.	CO4
15	Gain unauthorized access to a target system and demonstrate post-exploitation activities like privilege escalation and data exfiltration.	CO4
16	Perform a wireless network assessment using tools like Aircrack-ng to crack WEP or WPA/WPA2 encryption keys.	CO4
17	Demonstrate the use of rogue access points and man-in-the-middle attacks on a wireless network.	CO4
18	Case Study on Windows linux system security	CO1
19	Install jcrypt tool (or any other equivalent) and demonstrate Asymmetric, Symmetric Crypto algorithm	CO4
20	Install tool for Hash and Digital/PKI signatures studied in theory Network Security And Management	CO4
21	Implement Passive scanning, active scanning using Burp suit tool	CO4
22	Implement session hijacking using any network tool	CO4
23	Implement cookies extraction using Burp suit tool	CO4
24	Conduct a phishing attack to obtain sensitive information from users	CO4
25	Execute a spear-phishing attack to target specific individuals in the organization.	CO4

Projects

1. Web Application Security Assessment:

Description: Build a web application or use a vulnerable web application. Perform a comprehensive security assessment using ethical hacking techniques, such as SQL injection, cross-site scripting (XSS), and other OWASP Top 10 vulnerabilities. Generate a detailed report with identified vulnerabilities and proposed countermeasures to enhance the application's security.

2. Network Penetration Testing:

Description: Set up a simulated network environment with multiple devices and services. Conduct a penetration test using tools like Nmap, Metasploit, and Wireshark to identify weaknesses in the network infrastructure and devices. Develop a report outlining the potential risks and recommendations for securing the network.

3. Wireless Network Security Assessment:

Description: Create a wireless network environment with different security protocols (WEP, WPA/WPA2, etc.). Conduct a wireless network security assessment using tools like Aircrack-ng and Wireshark to identify vulnerabilities and assess the



effectiveness of encryption methods. Provide recommendations to improve wireless network security.

4. **Malware Analysis and Reverse Engineering:**

Description: Obtain malware samples (with proper authorization) and analyze their behavior in a controlled environment using virtual machines or sandboxes. Reverse engineer the malware to understand its code and functionalities. Prepare a detailed analysis report with insights into the malware's intent and potential mitigation strategies.

5. Red Team vs. Blue Team Exercise:

Description: Divide students into red and blue teams. The red team's objective is to simulate an attack on a target system/network using ethical hacking techniques. The blue team's objective is to defend against the red team's attacks and identify and mitigate security breaches. The exercise aims to simulate real-world scenarios and foster a competitive yet cooperative learning environment.

DSE- I

SECURE CODING & VULNERABILITIES

Department:	Department of Computer Science and Engineering
--------------------	---



Course Name: Secure Coding & Vulnerabilities		Course Code	L-T-P	Credits
		ENSP401	4-0-0	4
Type of Course:	Minor			
Pre-requisite(s), if any:				
Brief Syllabus:				
<p>Secure Coding and Vulnerabilities is a comprehensive course that focuses on understanding and mitigating application security threats and attacks. The course covers various aspects such as security requirements gathering, secure application design and architecture, and secure coding practices for input validation, authentication, cryptography, session management, and error handling. Students will learn about common application vulnerabilities and the potential consequences of security breaches. They will also gain knowledge and skills in static and dynamic application security testing methods, as well as secure deployment and maintenance practices. The course aims to equip students with the necessary tools and techniques to develop robust and secure applications while adhering to best practices in the field of application security.</p>				
UNIT WISE DETAILS				
Unit Number: 1	Title: Introduction to coding and Security		No. of hours: 12	
Content Summary:				
Introduction-security concepts-CIA Triad, Viruses, Trojans, and Worms, threat, vulnerability, risk, attack. Coding Standards: Dirty Code and Dirty Compiler, Dynamic Memory Management functions, Common memory management Errors (Initialization Errors, Forget to Check Return Values, accessing already freed memory, Freeing the same memory multiple times, Forget to free the allocated memory), Integer Security –Introduction to integer types: Integer Data Types, data type conversions, Integer vulnerabilities and mitigation strategies				
Unit Number: 2	Title: Secure Application Design and Architecture		No. of hours: 8	
Content Summary:				
Security requirements gathering and analysis, Secure software development life cycle (SSDLC) , Security issues while writing SRS, Design phase security, Development Phase, Test Phase, Maintenance Phase, Writing Secure Code – Best Practices SD3 (Secure by design, default and deployment), Security principles and Secure Product Development Timeline.				
Unit Number: 3	Title: Secure Coding Practices and Vulnerabilities		No. of hours: 8	
Content Summary:				
Input validation Techniques-whitelist validation, regular expressions, authentication and authorization, Cryptography, buffer overflows, Session management and protection against session-related attacks, Secure error handling and logging practices, SQL Injection Techniques and Remedies, Race conditions				



Unit Number: 4	Title: Application Security Testing and Deployment	No. of hours: 12
--------------------------	---	-------------------------

Content Summary:
Security code overview, Secure software installation. The Role of the Security Tester, Building the Security Test Plan. Testing HTTP-Based Applications, Testing File-Based Applications, Testing Clients with Rogue Servers, Static and Dynamic Application Security Testing (SAST & DAST), Secure Deployment and Maintenance, Patch management and software updates, Vulnerability scanning and penetration testing.

- *Self-Learning Components:**
- Code Review Tools:** Students can explore open-source code review tools such as SonarQube, ESLint, or FindBugs to understand how these tools can help identify security vulnerabilities in code.
 - Security Frameworks** such as OWASP (Open Web Application Security Project) and their associated resources.
 - Secure Development Tools:** Students can explore tools like Burp Suite, ZAP (Zed Attack Proxy), or WebInspect to understand how these tools can be used for dynamic application security testing (DAST) and penetration testing.
 - Secure Coding in Web Applications:** Students can dive deeper into web application security topics, such as Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), or security measures like Content Security Policy (CSP) and HTTP security headers.

Reference Books:

- Writing Secure Code, Michael Howard and David LeBlanc, Microsoft Press, 2nd Edition, 2004
- Buffer Overflow Attacks: Detect, Exploit, Prevent by Jason Deckard ,Syngress, 1st Edition, 2005
- Threat Modeling, Frank Swiderski and Window Snyder, Microsoft Professional, 1st Edition ,2004
- Secure Coding: Principles and Practices by Mark G. Graff, Kenneth R. van Wyk, Publisher(s): O'Reilly Media, Inc., 2003
- The Software Vulnerability Guide (Programming Series) by H. Thompson (Author), Scott G. Chase, 2005

Reference Links:

- "Secure Coding Practices" on Udemy - Offered by The App Brewery. Link: <https://www.udemy.com/course/secure-coding-practices/>
- "Secure Coding: Preventing Software Vulnerabilities" on Pluralsight - Offered by Pluralsight. Link: [Secure Coding: Preventing Software Vulnerabilities](#)
- "Software Security" on edX - Offered by University of Maryland, College Park. Link: [Software Security](#)
- [Identifying Security Vulnerabilities in C/C++ Programming | Coursera](#)
- [Principles of Secure Coding | Coursera](#)
- [Identifying Security Vulnerabilities | Coursera](#)

Define Course Outcomes (CO)

COs	Statements
CO1	Understand different types of application security threats and their potential impact.



CO2	Apply secure design principles and architectures to develop robust and secure applications.
CO3	Implement secure coding practices for input validation, authentication, cryptography, session management, and error handling.
CO4	Conduct static and dynamic application security testing to identify vulnerabilities and implement secure deployment and maintenance practices.

COs Mapping with Levels of Bloom’s taxonomy

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

CO-PO Mapping

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

1=weakly mapped



2= moderately mapped
3=strongly mapped

CO-PSO Mapping

CO-PSO	PSO1	PSO2	PSO3	PSO4
CO1	2	2	1	2
CO2	3	2	2	2
CO3	2	3	3	1
CO4	2	3	2	3

Relevance of the Syllabus to various indicators

Unit I	Introduction to coding and Security
Local	-
Regional	-
National	-
Global	Enhancing cybersecurity capabilities and promoting secure software development.
Employability	-
Entrepreneurship	-
Skill Development	Developing coding skills with a focus on security.
Professional Ethics	Develop applications that prioritize data security and user privacy.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	Secure Application Design and Architecture
Local	Applying security requirements and following secure software development life cycle (SSDLC) practices aligns with local needs to protect sensitive data and ensure secure application design.
Regional	-
National	-
Global	Promotes global standards in application security, fostering a global culture of secure software development.
Employability	Proficiency in secure application design and adherence to security principles enhances students' employability in organizations seeking professionals with secure development skills
Entrepreneurship	-
Skill Development	Enhances students' skills in secure software design,



	architecture, and implementation, preparing them to tackle application security challenges.
Professional Ethics	Ethical considerations of data privacy and security, fostering responsible development practices.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	Secure Coding Practices and Vulnerabilities
Local	Understanding input validation techniques, cryptography, and secure error handling directly addresses local needs for developing secure applications and mitigating common vulnerabilities.
Regional	-
National	Addressing vulnerabilities such as SQL injection and race conditions through secure coding practices aligns with national objectives of securing critical applications and preventing cyber attacks.
Global	Knowledge of secure coding practices and vulnerability mitigation strategies helps establish global standards for secure software development and promotes a secure digital environment worldwide.
Employability	-
Entrepreneurship	-
Skill Development	Hones students' skills in secure coding, vulnerability identification, and remediation, enhancing their technical capabilities in application security.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	Application Security Testing and Deployment
Local	-
Regional	-
National	-
Global	Understanding static and dynamic application security testing methods and secure deployment practices promotes global standards for secure software deployment and mitigating application vulnerabilities.
Employability	Expertise in application security testing and secure deployment enhances students' employability in roles focused on ensuring application security and secure software deployment.



Entrepreneurship	-
Skill Development	Develops students' skills in application security testing, penetration testing, and secure deployment, equipping them with practical expertise in securing applications.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG-4,9,16
NEP 2020	Skill development, employability, and entrepreneurship
POE/4 th IR	Emphasizes the importance of cybersecurity in the digital era.

SECURE CODING & VULNERABILITIES LAB

Department:	Department of Computer Science and Engineering
--------------------	---



Course Name: Secure Coding & Vulnerabilities Lab	Course Code	L-T-P	Credits
	ENSP451	0-0-2	1
Type of Course:	Minor		

Defined Course Outcomes

COs	
CO 1	Apply Secure Coding Techniques for enhancing application security.
CO 2	Analyze and Evaluate Security Vulnerabilities
CO 3	Evaluate and Communicate Importance of Secure Coding by analyzing potential vulnerabilities
CO 4	Design and Implement Secure Applications using secure cryptographic libraries.

Proposed Lab Experiments

Ex. No	Experiment Title	Mapped CO/COs
1	Write code to convert between different data types	CO1
2	Implement dynamic memory allocation and deallocation operations, and analyze potential errors and vulnerabilities.	CO2
3	Write code snippets with initialization errors, memory leaks, and double free issues, and use tools like Valgrind to detect and fix these errors.	CO2
4	Analyze a given code snippet with dirty code practices	CO2
5	Perform static code analysis on a sample codebase using a secure coding tool SonarQube	CO3
6	Conduct dynamic code analysis on a web application using OWASP	CO2
7	Configure the tool to intercept and analyze HTTP requests and responses.	CO2
8	Implement encryption algorithms (e.g., AES) using secure cryptographic libraries or frameworks.	CO4
9	Implement and test whitelist validation techniques to ensure secure input handling.	CO1
10	Develop a simple web application that requires user authentication.	CO4
11	Implement cryptographic functions for secure data protection	CO4
12	Implement input sanitization and validation techniques to prevent SQL injection attacks.	CO1
13	Conduct a security audit and penetration testing on a	CO2



	provided application to identify SQL injection vulnerabilities.	
14	Identify common memory management errors such as forgetting to check return values or accessing already freed memory.	CO2
15	Write a sample code that requires input validation, such as user input or data from external sources.	CO1
16	Write a sample code that involves cryptographic operations, such as encryption or hashing.	CO4
17	Conduct security testing on an HTTP-based application to identify vulnerabilities and security weaknesses.	CO2
18	Set up a local or web-based application that operates over HTTP. Perform security testing using appropriate tools and techniques, such as vulnerability scanners and penetration testing	CO2
19	Perform security testing on a file-based application to assess its security posture and identify potential vulnerabilities.	CO2
20	Utilize appropriate tools and techniques to conduct static analysis on the application's source code to identify potential vulnerabilities	CO3
21	Identify and configure important HTTP security headers, such as Strict-Transport-Security (HSTS), X-Frame-Options, X-XSS-Protection, and X-Content-Type-Options.	CO1
22	Develop a sample web application that includes error handling and logging functionality.	CO4
23	Implement secure error handling techniques, such as displaying generic error messages to users and logging detailed errors only to authorized personnel.	CO4
24	Apply secure coding best practices, such as input validation, output encoding, proper error handling, and secure use of APIs and libraries.	CO1
25	Test the code for vulnerabilities and discuss the importance of writing secure code to prevent potential exploitation.	CO3

Projects

- Implement a secure software development lifecycle
http://www.owasp.org/index.php/Category:OWASP_CLASP_Project
- Establish secure coding standards
http://www.owasp.org/index.php/Category:OWASP_Guide_Project
- Build a re-usable object library
http://www.owasp.org/index.php/Category:OWASP_Enterprise_Security_API
- Verify the effectiveness of security controls



[http://www.owasp.org/index.php/Category:OWASP Application Security Verification Standard Project](http://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project)

- Establish secure outsourced development practices including defining security requirements and verification methodologies in both the request for proposal (RFP) and contract.

[http://www.owasp.org/index.php/Category:OWASP Legal Project](http://www.owasp.org/index.php/Category:OWASP_Legal_Project)

CYBER CRIME INVESTIGATION & DIGITAL FORENSICS

Department:	Department of Computer Science and Engineering
--------------------	---



Course Name: Cyber Crime Investigation & Digital Forensics	Course Code	L-T-P	Credits
	ENSP403	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			
Brief Syllabus: Introduces the principles and practices of digital forensics including digital investigations, data and file recovery methods, and digital forensics analysis and invalidation. Topics include data acquisition, digital forensics tools, virtual machines, network, mobile devices and cloud forensics.			
UNIT WISE DETAILS			
Unit Number: 1	Title: Introduction	No. of hours: 8	
Content Summary: Introduction to Digital Forensics, Definition and types of cybercrimes, electronic evidence and handling, electronic media, collection, searching and storage of electronic media, introduction to internet crimes, hacking and cracking, credit card and ATM frauds, web technology, cryptography, emerging digital crimes and modules.			
Unit Number: 2	Title: Types of Cyber Crimes	No. of hours: 10	
Content Summary: Crimes targeting Computers: Unauthorized Access Packet Sniffing Malicious Codes including Trojans, Viruses, Logic Bombs, etc. Online based Cyber Crimes: Phishing and its variants Web Spoofing and E-mail Spoofing Cyber Stalking Web defacement Financial crimes, ATM and Card Crimes etc Spamming Commercial espionage and Commercial Extortion online Software and Hardware Piracy Money Laundering Fraud & Cheating Other Cyber Crimes.			
Unit Number: 3	Title: Investigation of Cyber Crimes	No. of hours: 12	
Content Summary: Investigation of malicious applications Agencies for investigation in India, their powers and their constitution as per Indian Laws Procedures followed by First Responders; Evidence Collection and Seizure Procedure of Digital mediums Securing the Scene, Documenting the Scene, Evidence Collection and Transportation Data Acquisition Data Analysis Reporting			
Unit Number: 4	Title : Forensic Tools and Processing of Electronic Evidence	No. of hours: 10	



Content Summary:

Introduction to Forensic Tools, Usage of Slack space, tools for Disk Imaging, Data Recovery, Vulnerability Assessment Tools, Encase and FTK tools, Anti Forensics and probable counters, retrieving information, process of computer forensics and digital investigations, processing of digital evidence, digital images, damaged SIM and data recovery, multimedia evidence, retrieving deleted data: desktops, laptops and mobiles, retrieving data from slack space, renamed file, ghosting, compressed files.

***Self-Learning Components:**

1. **Open-Source Digital Forensics Tools: Introduction to popular open-source digital forensics tools such as Autopsy, Sleuth Kit, and Volatility.**
2. **Exploring open-source threat intelligence platforms like MISP and AlienVault OTX.**
3. **Discussing the benefits of information sharing and collaborative efforts in combating cyber threats.**
4. **Digital Forensics and Cyber-Crime Investigation,**<https://www.udemy.com/course/digital-forensics-and-cyber-crime-investigation/>

Reference Books:

- Moore, Robert, (2011). Cybercrime, investigating high-technology computer crime(2nd Ed.). Elsevier
- C. Altheide& H. Carvey Digital Forensics with Open Source Tools, Syngress, 2011.
- Majid Yar, "Cybercrime and Society", SAGE Publications Ltd, Hardcover, 2nd Edition, 2013.
- Robert M Slade, "Software Forensics: Collecting Evidence from the Scene of a Digital Crime", Tata McGraw Hill, Paperback, 1st Edition, 2004.

Web references:

- <https://www.coursera.org/learn/digital-forensics-concepts>
- <https://www.udemy.com/course/computer-forensics-and-digital-forensics-for-everyone/>

Define Course Outcomes (CO)

COs	Statements
CO1	Understand the nature and classification of conventional and cyber-crimes.
CO2	Analyze and identify various types of cyber-crimes and their modes of operation.



CO3	Evaluate the impact of cyber-crimes on individuals, organizations, and society.
CO4	Develop an understanding of digital forensics and the investigative procedures used in cyber-crime cases.
CO5	Apply forensic tools and techniques to retrieve and analyze digital evidence.

COs Mapping with Levels of Bloom’s taxonomy

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

***Please Note:**

Map only 1 or 2 Levels in each category. If a higher level is given, no need to mention lower level

CO-PO Mapping

COs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------



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

Please Note:

- Refer to POs while mapping each CO.
- Mark " - " if not applicable
- If attainment of a CO is strongly mapped with a PO , Mark 3
- If attainment of a CO is moderately mapped with a PO , Mark 2
- If attainment of a CO is weakly mapped with a PO , Mark 1

Justification for mapping must be relevant

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	Introduction
Local	The knowledge and understanding of cybercrime and computer crime can help local communities and law enforcement agencies address and prevent such crimes in their area.
Regional	Cybercrime is a regional concern, and understanding its types and emerging trends can help in regional collaboration for combating cyber threats.
National	Cybercrime is a significant concern at the national level. Developing expertise in digital forensics and cybercrime investigation can enhance national security and protect critical infrastructure.
Global	Cybercrime has a global impact, and knowledge in this area can contribute to international efforts in combating cyber threats and promoting cybersecurity.
Employability	The skills and knowledge gained in this unit can enhance



	employability in the field of cybersecurity, law enforcement, digital forensics, and related industries.
Entrepreneurship	-
Skill Development	Developing skills in digital forensics, evidence handling, and understanding emerging digital crimes, contributing to skill development in the field.
Professional Ethics	Studying cybercrime and computer crime can raise awareness of ethical issues related to information security, privacy, and responsible use of technology.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	Types of Cyber Crimes
Local	-
Regional	-
National	Cyber crimes pose significant challenges to national security and the economy. Understanding different types of cyber crimes allows governments and law enforcement agencies to develop robust policies, laws, and strategies to address cyber threats at the national level.
Global	Cyber crimes have a global reach and impact. By studying the types of cyber crimes, individuals and organizations can contribute to global efforts in promoting cybersecurity, sharing threat intelligence, and developing international frameworks to combat cyber threats.
Employability	Acquiring knowledge about various types of cybercrimes enhances employability in the field of cybersecurity.
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in internet technologies and network protocols
Professional Ethics	Awareness of different types of cybercrimes raises ethical considerations surrounding privacy, data protection, and responsible use of technology.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	Investigation of Cyber Crimes
Local	-
Regional	Collaboration among regional investigation agencies can be improved through the knowledge of investigation procedures and digital evidence handling.
National	Investigating cyber crimes is a critical aspect of national security, and this unit's content can enhance the investigation



	capabilities of agencies at the national level.
Global	Aligns with global Cooperation and sharing of best practices in cybercrime investigation.
Employability	Proficiency in cybercrime investigation and evidence handling is in high demand, offering employment opportunities in the field of digital forensics and cybersecurity.
Entrepreneurship	Knowledge in cybercrime investigation can inspire entrepreneurs to develop innovative tools and services for digital forensics and incident response.
Skill Development	Developing skills in evidence collection, data analysis, and reporting, contributing to skill development in the field of cybercrime investigation.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	Forensic Tools and Processing of Electronic Evidence
Local	The knowledge and skills gained in this unit are relevant at the local level as local law enforcement agencies and forensic professionals need to be equipped with the tools and techniques to effectively process electronic evidence in cybercrime investigations within their jurisdiction
Regional	-
National	Protecting national security and upholding the rule of law in the digital realm requires a strong capability in digital forensics. The knowledge and proficiency in forensic tools and processing of electronic evidence contribute to national efforts in preventing and investigating cybercrimes.
Global	Cybercrimes are a global concern, and international cooperation is vital in addressing them.
Employability	Proficiency in forensic tools and processing of electronic evidence enhances employability in the field of digital forensics and cybersecurity.
Entrepreneurship	Knowledge of forensic tools and techniques can inspire entrepreneurs to develop innovative solutions, tools, and services in the field of digital forensics.
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4,9,16
NEP 2020	Its aligns with the policy's objective of developing skills



	relevant to the current and future job market, particularly in the field of cyber security.
POE/4 th IR	The Fourth Industrial Revolution by providing knowledge and skills necessary to combat cyber threats and protect digital assets in an increasingly interconnected and digital world

CYBER CRIME INVESTIGATION & DIGITAL FORENSICS LAB

Department:	Department of Computer Science and Engineering
--------------------	---



Course Name: Cyber Crime Investigation & Digital Forensics Lab	Course Code	L-T-P	Credits
	ENSP453	0-0-2	1
Type of Course:	Minor		
Pre-requisite(s), if any:			

Defined Course Outcomes

COs	
CO 1	Understand the fundamental concepts and principles of digital forensics and cybercrimes.
CO 2	Apply the knowledge of digital forensics techniques and procedures to collect, analyse, and preserve electronic evidence in various types of cybercrimes.
CO 3	Evaluate and utilize forensic tools and technologies for data acquisition, analysis, and recovery in the investigation of cybercrimes.
CO 4	Analyse and interpret digital evidence obtained from different sources, such as electronic media, internet crimes, malicious applications, and various forms of cybercrimes.

Proposed Lab Experiments

Ex. No	Experiment Title	Mapped CO/COs
1	Experiment on collecting and preserving electronic media for forensic analysis.	CO 2
2	Experiment on searching and retrieving digital evidence from various storage devices.	CO 2
3	Experiment on handling and analyzing malicious codes, such as Trojans, viruses, and logic bombs.	CO 2
4	Experiment on investigating unauthorized access to computer systems.	CO 2
5	Experiment on packet sniffing and analyzing network traffic for evidence.	CO 2
6	Experiment on identifying and investigating phishing attacks and their variants.	CO 2
7	Experiment on detecting and investigating web spoofing and email spoofing incidents.	CO 2
8	Experiment on cyber stalking investigation techniques.	CO 2
9	Experiment on investigating web defacement incidents and identifying the perpetrators.	CO 2
10	Experiment on investigating financial crimes, including	CO 2



	ATM and credit card frauds.	
11	Experiment on tracing and investigating spamming activities.	CO 2
12	Experiment on investigating cases related to software and hardware piracy.	CO 2
13	Experiment on tracing and investigating money laundering activities.	CO 2
14	Experiment on investigating fraud and cheating cases in the digital realm.	CO 2
15	Experiment on analyzing malicious applications and their impact on digital devices.	CO 4
16	Experiment on understanding the role and capabilities of investigation agencies in India.	CO 1
17	Experiment on following proper evidence collection and seizure procedures in digital investigations.	CO 2
18	Experiment on securing and documenting the crime scene in digital forensics.	CO 2
19	Experiment on acquiring and analyzing data from digital devices.	CO 3
20	Experiment on using forensic tools like EnCase and FTK for digital investigations.	CO 3
21	Experiment on countering anti-forensics techniques and retrieving hidden information.	CO 3
22	Experiment on recovering data from damaged SIM cards and other multimedia evidence.	CO 2
23	Experiment on recovering deleted data from desktops, laptops, and mobile devices.	CO 2
24	Experiment on analyzing data from slack space and renamed files.	CO 4
25	Experiment on forensic imaging, including ghosting and analysis of compressed files.	CO 3

Description of experiments:

Session 1:

Topic: Experiment on collecting and preserving electronic media for forensic analysis

- Introduction to electronic media collection and preservation in digital forensics
- Techniques for ensuring the integrity and authenticity of collected data
- Chain of custody and documentation procedures

Exercise: Practice collecting electronic media and preserving it for forensic analysis.

Project: Create a comprehensive report on the collection and preservation of electronic media: Document the process of collecting electronic media, maintain a chain of custody, and ensure the integrity of the collected data.



Session 2:

Topic: Experiment on searching and retrieving digital evidence from various storage devices

- Different types of storage devices and their characteristics
- Techniques for searching and retrieving digital evidence from storage devices
- File systems analysis and data carving

Exercise: Search for and retrieve digital evidence from different storage devices.

Project: Analyze and document the process of searching and retrieving digital evidence: Perform data recovery and analysis on different storage devices, document the findings, and present a comprehensive report.

Session 3:

Topic: Experiment on handling and analyzing malicious codes, such as Trojans, viruses, and logic bombs

- Introduction to different types of malicious codes
- Techniques for analyzing and understanding malicious code behavior
- Anti-malware tools and techniques

Exercise: Analyze and dissect different types of malicious codes to understand their behavior.

Project: Develop a comprehensive report on the analysis of malicious codes: Analyze and document the behavior of various malicious codes, identify their impact, and propose countermeasures.

Session 4:

Topic: Experiment on investigating unauthorized access to computer systems

- Understanding the concept of unauthorized access
- Techniques for investigating unauthorized access incidents
- Log analysis and intrusion detection systems

Exercise: Investigate and analyze unauthorized access incidents in computer systems.

Project: Create a detailed investigation report on unauthorized access incidents: Analyze log files, identify the extent of unauthorized access, determine the entry points, and propose preventive measures.

Session 5:

Topic: Experiment on packet sniffing and analyzing network traffic for evidence

- Introduction to packet sniffing and network traffic analysis
- Tools and techniques for capturing and analyzing network packets
- Identifying and extracting relevant evidence from network traffic

Exercise: Capture and analyze network packets to extract evidence.

Project: Prepare a comprehensive report on network traffic analysis for a given scenario: Analyze captured network packets, extract relevant evidence, and present the findings in a structured report.

Session 6:

Topic: Experiment on identifying and investigating phishing attacks and their variants

- Understanding phishing attacks and their impact
- Techniques for identifying and investigating phishing incidents
- Analyzing phishing emails and websites



Exercise: Identify and investigate phishing attacks by analyzing phishing emails and websites.

Project: Perform a comprehensive analysis of a phishing attack: Analyze phishing emails and websites, identify the modus operandi, and propose countermeasures to prevent future attacks.

Session 7:

Topic: Experiment on detecting and investigating web spoofing and email spoofing incidents

- Understanding web spoofing and email spoofing techniques
- Techniques for detecting and investigating web and email spoofing incidents
- Analyzing spoofed web pages and email headers

Exercise: Detect and investigate web spoofing and email spoofing incidents by analyzing spoofed web pages and email headers.

Project: Prepare a detailed investigation report on web and email spoofing incidents: Analyze spoofed web pages and email headers, identify the perpetrators, and suggest preventive measures.

Session 8:

Topic: Experiment on cyber stalking investigation techniques

- Understanding cyber stalking and its implications
- Techniques for investigating cyber stalking incidents
- Gathering digital evidence and documenting the case

Exercise: Investigate and gather digital evidence for a cyber stalking case.

Project: Create a comprehensive investigation report on a cyber stalking incident: Analyze the digital evidence, document the case details, and propose measures to protect the victim.

Session 9:

Topic: Experiment on investigating web defacement incidents and identifying the perpetrators

- Understanding web defacement and its impact
- Techniques for investigating web defacement incidents
- Analyzing web defaced pages and server logs

Exercise: Investigate web defacement incidents and analyze defaced web pages and server logs.

Project: Prepare a detailed investigation report on web defacement incidents: Analyze defaced web pages and server logs, identify the perpetrators, and suggest measures to enhance website security.

Session 10:

Topic: Experiment on investigating financial crimes, including ATM and credit card frauds

- Understanding financial crimes in the digital realm
- Techniques for investigating ATM and credit card frauds
- Analyzing financial transaction records and digital evidence

Exercise: Investigate financial crimes related to ATM and credit card frauds by analyzing financial transaction records and digital evidence.

Project: Create a comprehensive report on the investigation of financial crimes: Analyze financial transaction records, identify fraudulent activities, and propose preventive measures.



Session 11:

Topic: Experiment on tracing and investigating spamming activities

- Understanding spamming activities and their impact
- Techniques for tracing and investigating spamming incidents
- Analyzing spam emails and tracking email senders

Exercise: Trace and investigate spamming activities by analyzing spam emails and tracking email senders.

Project: Prepare a detailed investigation report on spamming activities: Analyze spam emails, trace email senders, identify the source of spamming, and propose measures to mitigate spamming incidents.

Session 12:

Topic: Experiment on investigating cases related to software and hardware piracy

- Understanding software and hardware piracy and its consequences
- Techniques for investigating piracy cases
- Analyzing pirated software and counterfeit hardware

Exercise: Investigate cases related to software and hardware piracy by analyzing pirated software and counterfeit hardware.

Project: Develop a comprehensive report on software and hardware piracy investigations: Analyze pirated software, identify counterfeit hardware, determine the extent of piracy, and propose measures to combat piracy.

Session 13:

Topic: Experiment on tracing and investigating money laundering activities

- Understanding money laundering in the digital realm
- Techniques for tracing and investigating money laundering incidents
- Analyzing financial transaction records and blockchain data

Exercise: Trace and investigate money laundering activities by analyzing financial transaction records and blockchain data.

Project: Prepare a detailed investigation report on money laundering activities: Analyze financial transaction records, track money flow, identify money laundering techniques, and propose measures to prevent money laundering.

Session 14:

Topic: Experiment on investigating fraud and cheating cases in the digital realm

- Understanding fraud and cheating in the digital realm
- Techniques for investigating fraud and cheating cases
- Analyzing digital evidence and transaction records

Exercise: Investigate fraud and cheating cases in the digital realm by analyzing digital evidence and transaction records.

Project: Create a comprehensive investigation report on fraud and cheating cases: Analyze digital evidence, identify fraudulent activities, document the case details, and propose preventive measures.

Session 15:

Topic: Experiment on analyzing malicious applications and their impact on digital devices

- Understanding malicious applications and their impact
- Techniques for analyzing and identifying malicious applications
- Analyzing malware behavior and reverse engineering techniques



Exercise: Analyze and identify malicious applications and study their impact on digital devices.

Project: Prepare a detailed analysis report on malicious applications: Analyze the behavior of different types of malicious applications, identify their impact on digital devices, and propose measures to prevent malware infections.

Session 16:

Topic: Experiment on understanding the role and capabilities of investigation agencies in India

- Introduction to investigation agencies in India
- Understanding the roles and responsibilities of investigation agencies
- Case studies and examples of investigations conducted by Indian agencies

Exercise: Study and understand the roles and capabilities of investigation agencies in India through case studies and examples.

Project: Prepare a report highlighting the role and capabilities of investigation agencies in India: Discuss the functions, powers, and responsibilities of key investigation agencies, and analyze their notable investigations.

Session 17:

Topic: Experiment on following proper evidence collection and seizure procedures in digital investigations

- Understanding the importance of proper evidence collection and seizure
- Techniques and procedures for collecting and preserving digital evidence
- Documentation and chain of custody requirements

Exercise: Practice following proper evidence collection and seizure procedures in digital investigations.

Project: Create a comprehensive report on evidence collection and seizure procedures: Document the process of evidence collection, maintain the chain of custody, and ensure compliance with legal and procedural requirements.

Session 18:

Topic: Experiment on securing and documenting the crime scene in digital forensics

- Importance of securing the crime scene in digital forensics
- Techniques for securing and documenting the crime scene
- Best practices for maintaining the integrity of digital evidence

Exercise: Secure and document the crime scene in a simulated digital forensics case.

Project: Prepare a detailed report on securing and documenting the crime scene: Describe the steps taken to secure the crime scene, document the process, and provide recommendations for improving crime scene management.

Session 19:

Topic: Experiment on acquiring and analyzing data from digital devices

- Techniques for acquiring data from digital devices
- Best practices for preserving the integrity of acquired data
- Analyzing acquired data using forensic tools and techniques

Exercise: Acquire and analyze data from different digital devices using forensic tools and techniques.



Project: Analyze and document the process of acquiring and analyzing data from digital devices: Perform data acquisition, analyze the acquired data, and present the findings in a structured report.

Session 20:

Topic: Experiment on using forensic tools like EnCase and FTK for digital investigations

- Introduction to popular forensic tools like EnCase and FTK
- Familiarization with the features and capabilities of forensic tools
- Hands-on practice with forensic tool usage in digital investigations

Exercise: Use forensic tools like EnCase and FTK to conduct digital investigations on simulated cases.

Project: Prepare a comprehensive report on the usage of forensic tools in digital investigations: Describe the features and capabilities of EnCase and FTK, document the usage in specific investigations, and evaluate their effectiveness.

Session 21:

Topic: Experiment on countering anti-forensics techniques and retrieving hidden information

- Understanding anti-forensics techniques used to hide digital evidence
- Techniques for countering anti-forensics and retrieving hidden information
- Analysis of steganography, encryption, and file obfuscation methods

Exercise: Counter anti-forensics techniques and retrieve hidden information from digital evidence.

Project: Develop a comprehensive report on countering anti-forensics techniques: Analyze different anti-forensics methods, propose countermeasures, and demonstrate the retrieval of hidden information.

Session 22:

Topic: Experiment on recovering data from damaged SIM cards and other multimedia evidence

- Techniques for recovering data from damaged SIM cards
- Recovering data from damaged multimedia evidence like CCTV footage and audio recordings
- Best practices for data recovery from different types of damaged media

Exercise: Recover data from damaged SIM cards and analyze multimedia evidence from various sources.

Project: Prepare a detailed report on data recovery from damaged media: Document the process of recovering data from damaged SIM cards and analyze recovered multimedia evidence.

Session 23:

Topic: Experiment on recovering deleted data from desktops, laptops, and mobile devices

- Techniques for recovering deleted data from different devices
- Understanding file systems and data storage mechanisms
- Analyzing recovered deleted data for evidence

Exercise: Recover deleted data from desktops, laptops, and mobile devices and analyze the recovered data for evidence.



Project: Analyze and document the process of recovering deleted data: Recover deleted data from different devices, analyze the recovered data, and present the findings in a comprehensive report.

Session 24:

Topic: Experiment on analyzing data from slack space and renamed files

- Understanding slack space and its significance in digital forensics
- Techniques for analyzing data from slack space and renamed files
- Extracting hidden information and evidence from slack space and renamed files

Exercise: Analyze data from slack space and renamed files to extract hidden information and evidence.

Project: Prepare a detailed report on the analysis of data from slack space and renamed files: Analyze the data, extract hidden information, and present the findings in a structured report.

Session 25:

Topic: Experiment on forensic imaging, including ghosting and analysis of compressed files

- Understanding forensic imaging and its importance in digital forensics
- Techniques for creating forensic images and conducting analysis
- Analyzing ghost images and compressed files for evidence

Exercise: Create forensic images, analyze ghost images, and conduct analysis on compressed files.

Project: Develop a comprehensive report on forensic imaging and analysis: Describe the process of creating forensic images, analyze ghost images, and analyze compressed files for evidence. Present the findings in a structured report.



Department:	Department of Computer Science and Engineering		
Course Name: AI in Cyber Security	Course Code	L-T-P	Credits
	ENSP405	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any: basic understanding of web development technologies such as HTML, CSS, and JavaScript. Additionally, students should have some familiarity with networking concepts, operating systems, and databases.			
Brief Syllabus: This syllabus covers essential topics in web application security, including injection attacks, authentication and access control, cryptography, testing, security standards, best practices, and risk management. It is divided into four units and may be completed in a semester-long course. Students will gain an understanding of common web application vulnerabilities and how to prevent and mitigate them. They will also learn about authentication and access control mechanisms, cryptography techniques, and web application security testing. Finally, students will explore best practices for secure web application development and incident response and disaster recovery planning.			
UNIT WISE DETAILS			
Unit Number: 1	Title: Introduction to AI and Cyber Security	No. of hours: 10	
Content Summary: Overview of Artificial Intelligence and its applications in Cyber Security, History and evolution of AI in cyber security, Understanding of the Cyber Security threats landscape, Familiarization with the latest trends and techniques of AI in Cyber Security, Basic principles of Machine Learning and Deep Learning in Cyber Security, Ethical considerations and challenges of using AI in cyber security.			
Unit Number: 2	Title: Machine Learning Techniques for Cyber Security	No. of hours: 10	
Content Summary: An introduction to Machine Learning techniques, Supervised and unsupervised Machine Learning models in Cyber Security, Feature engineering and data preparation for Machine Learning models, Case studies demonstrating the application of Machine Learning to Cyber Security problems.			
Unit Number: 3	Title: Deep Learning Techniques for Cyber	No. of hours: 10	
Content Summary: Introduction to Deep Learning techniques, Convolutional Neural Networks (CNNs) and their application in Cyber Security, Recurrent Neural Networks (RNNs) and their application in Cyber Security, GANs and their application in Cyber Security, Case studies demonstrating the application of Deep Learning to Cyber Security problems.			



Unit Number: 4	Title: AI for Cyber Security: Threat Detection and Prevention	No. of hours: 10
Content Summary: Introduction to AI and its applications in threat detection and prevention ,Overview of different types of threats in cyber security and their characteristics ,Understanding the limitations of traditional threat detection and prevention methods ,Fundamentals of machine learning and deep learning for threat detection and prevention ,Supervised machine learning algorithms for threat detection, such as decision trees, support vector machines, and random forests ,Unsupervised machine learning algorithms for anomaly detection, such as clustering and outlier detection ,Deep learning techniques for threat detection, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) ,Feature selection and feature engineering for machine learning in threat detection, Emerging trends and challenges in AI for threat detection and prevention, including adversarial machine learning, explainable AI, and privacy concerns.		
*Self-Learning Components: <ol style="list-style-type: none"> 1. Anomaly Detection 2. Malware Detection 3. Adaptive Access Control 4. Network Traffic Analysis 		
Reference Books: <ol style="list-style-type: none"> 1. Artificial Intelligence for Cybersecurity" by Bhaskar Sinha (Auerbach Publications) 2. Machine Learning and Security: Protecting Systems with Data and Algorithms" by Clarence Chio and David Freeman (O'Reilly Media) 		

Define Course Outcomes (CO)

COs	Statements
CO1	Understand Understand the concepts and applications of AI in the field of cyber security.
CO2	Express the ethical and legal considerations associated with the use of AI in cyber security.
CO3	Determine emerging trends and technologies in AI for cyber security, and their potential impact on the field.
CO4	Identify strategies for integrating AI-driven solutions into existing cyber security frameworks, policies, and practices.
CO5	Articulate critical thinking and problem-solving skills to address real-world cyber security challenges using AI techniques.
CO6	Design machine learning techniques for threat detection and prevention in cyber security, including supervised and unsupervised algorithms.



COs Mapping with Levels of Bloom’s taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A1	P1
CO2	C3	A3	P2
CO3	C3	A3	P3
CO4	C4	A3	-
CO5	C4	A3	P5
CO6	C6	A4	-

***Please Note:**

Map only 1 or 2 Levels in each category. If a higher level is given, no need to mention lower level

CO-PO Mapping

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



CO3	3	3	3	3	3	2	2	1	2	1
CO4	2	2	3	3	3	1	-	-	1	2
CO5	3	2	3	3	3	2	-	1	2	1
CO6	3	3	2	3	3	2	1	1	2	1

Please Note:

- Refer to POs while mapping each CO.
- Mark " - " if not applicable
- If attainment of a CO is strongly mapped with a PO , Mark 3
- If attainment of a CO is moderately mapped with a PO , Mark 2
- If attainment of a CO is weakly mapped with a PO , Mark 1

Justification for mapping must be relevant

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	Introduction to AI and Cyber Security
Local	Addresses local understanding of the Cyber Security and its impact on society
Regional	Addresses regional Cyber Security infrastructure requirements.
National	Contributes to national Cyber Security literacy and its impact to the nation.
Global	Aligns with global trends in Cyber Security technologies and network protocols
Employability	Develops skills in using Cyber Security and its tools for



	network protocols
Entrepreneurship	Build entrepreneurship
Skill Development	Develops basic knowledge and skills in Cyber Security technologies and network protocols
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	Machine Learning Techniques for Cyber Security
Local	Addresses local understanding of the Internet and its impact on society
Regional	-
National	Contributes to national digital literacy and internet connectivity strategies
Global	Aligns with global trends in internet technologies and network protocols
Employability	Develops skills in using Machine learning techniques and understanding network protocols
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in Machine learning techniques technologies and network protocols
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	Deep Learning Techniques for Cyber
Local	Addresses local network security needs and practices
Regional	-
National	Contributes to national network security strategies and protocols
Global	Aligns with global trends in network security techniques and protocols
Employability	Develops skills in Deep learning techniques and network security techniques
Entrepreneurship	-
Skill Development	Develops knowledge and skills in Deep learning techniques and network security
Professional Ethics	-
Gender	-



Human Values	-
Environment & Sustainability	-
Unit IV	AI for Cyber Security: Threat Detection and Prevention
Local	Addresses local understanding of Threat Detection and Prevention. and implementation of internet-based services
Regional	-
National	Contributes to national digital communication strategies and multimedia applications
Global	Aligns with global trends in internet telephony, multimedia applications, and SEO
Employability	Develops skills in Threat Detection and Prevention.
Entrepreneurship	-
Skill Development	Develops knowledge and skills in Threat Detection and Prevention
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	Aligns with the concepts of internet telephony, multimedia applications, and SEO

AI IN CYBER SECURITY LAB

Department:	Department of Computer Science and Engineering		
Course Name: AI in Cyber Security LAB	Course Code	L-T-P	Credits
	ENSP455	0-0-2	1



Type of Course:	Minor
Pre-requisite(s), if any: basic understanding of web development technologies such as HTML, CSS, and JavaScript. Additionally, students should have some familiarity with networking concepts, operating systems, and databases.	

Defined Course Outcomes

COs	Comprehensive Understanding of AI in Cyber Security:
CO 1	Practical Experience with AI Tools and Techniques
CO 2	Enhanced Malware Detection and Classification Skills
CO 3	Critical Thinking and Problem-Solving Abilities
CO 4	Research and Innovation in AI Cyber Security

Proposed Lab Experiments

Ex. No	Experiment Title	Mapped CO/COs
1	Malware detection: Develop an AI model to detect and classify different types of malware.	CO 2
2	Intrusion detection: Build an AI system to identify and alert on network intrusions and suspicious activities.	CO 1
3	Phishing detection: Train an AI algorithm to recognize and flag phishing emails or websites.	CO 1, CO 3
4	Vulnerability assessment: Use AI techniques to identify potential vulnerabilities in software or systems.	CO 1, CO 3
5	Botnet detection: Develop an AI model to detect and track botnet activities on a network.	CO 1
6	Password cracking: Build an AI system to analyze and crack weak passwords.	CO 1, CO 3
7	Network traffic analysis: Use AI algorithms to analyze network traffic and identify patterns or anomalies.	CO 1, CO 3
8	Behavioral authentication: Develop an AI model to authenticate users based on their behavioral patterns.	CO 1, CO 3
9	Anomaly detection: Train an AI system to detect anomalous behavior in user activities or system logs.	CO 1
10	Zero-day vulnerability detection: Use AI techniques to identify unknown or previously undiscovered vulnerabilities.	CO 1, CO 3
11	Social engineering detection: Build an AI system to recognize and alert on social engineering attempts.	CO 1, CO 3
12	Web application security: Develop an AI model to identify and mitigate web application vulnerabilities.	CO 3, CO 4



13	Data exfiltration detection: Train an AI algorithm to detect and prevent unauthorized data exfiltration attempts.	CO 3, CO 4
14	Ransomware detection: Use AI techniques to identify and block ransomware attacks in real-time.	CO 3, CO 4
15	Firewall optimization: Employ AI algorithms to optimize firewall rules and configurations for better security.	CO 3, CO 4
16	Network anomaly prediction: Build an AI system to predict network anomalies before they occur.	CO 1, CO 3
17	Security log analysis: Use AI techniques to automatically analyze and correlate security logs for identifying threats.	CO 3, CO 4
18	DDoS attack detection: Develop an AI model to detect and mitigate Distributed Denial of Service (DDoS) attacks.	CO 3, CO 4
19	Mobile application security: Train an AI algorithm to identify security vulnerabilities in mobile applications.	CO 3, CO 4
20	Network segmentation optimization: Employ AI techniques to optimize network segmentation for enhanced security.	CO 1, CO 3
21	Threat intelligence analysis: Use AI algorithms to analyze and extract insights from threat intelligence feeds.	CO 1, CO 3
22	Security incident response automation: Develop an AI system to automate and streamline security incident response processes.	CO 3, CO 4
23	Deepfake detection: Train an AI model to identify and flag manipulated or forged media content.	CO 1, CO 3
24	Network forensics: Use AI techniques to analyze network traffic and digital artifacts for forensic investigations.	CO 3, CO 4
25	Security policy compliance: Develop an AI system to assess and ensure compliance with security policies and regulations.	CO 3, CO 4

SOCIAL MEDIA SECURITY

Department:	Department of Computer Science and Engineering		
Course Name: Social Media Security	Course Code	L-T-P	Credits
	ENSP407	4-0-0	4



Type of Course:	Programme Elective-I (Cyber Security)	
Pre-requisite(s), if any:		
Brief Syllabus: Social media has become an integral part of our lives, shaping our online behaviors and interactions in numerous ways. People join social media platforms to share information, connect with friends, and engage in online communities. While social media offers these advantages, it also brings forth concerns regarding privacy and security. The constant flow of personal information shared on these platforms makes individuals vulnerable to various risks. Therefore, it is crucial for all of us to understand and address the issues surrounding privacy and security in the realm of social media. By acquiring knowledge about these challenges, we can adopt safer practices and protect ourselves from potential threats while enjoying the benefits of social media platforms. Being aware and proactive about social media security empowers us to navigate the digital landscape responsibly and ensure our online safety.		
UNIT WISE DETAILS		
Unit Number: 1	Title: Social Media Overview	No. of hours: 8
Content Summary: Introduction to Social media. Types of Social media, Social media platforms, Social media monitoring, Hashtag, Viral content, Social media marketing, challenges, opportunities, and pitfalls in online social networks, APIs, Collecting data from Online Social Media, Social Media Content Analysis - BoW Model, TF-IDF; Network Analysis - Node Centrality Measures, Degree Distribution, Average Path Length, Clustering Coefficient, Power Law; Synthetic Networks - Random Graphs, Preferential Attachment Model.		
Unit Number: 2	Title: Security Issues in Social Media	No. of hours: 11
Content Summary: Overview, Review of Machine Learning, The evolution of privacy and security concerns with networked technologies, Contextual influences on privacy attitudes and behaviours, Anonymity in a networked world, Identity Theft - Profile Cloning, Social Phishing, Fake, Compromised, Sybil accounts and their behaviour, Spamming, Rumour or Misinformation, Cyberbullying, Collective Misbehaviours, Flagging and reporting of inappropriate content.		
Unit Number: 3	Title: Privacy Issues in Social Media	No. of hours: 11
Content Summary: Overview, Privacy Settings, PII Leakage, Identity vs Attribute Disclosure Attacks, Inference Attacks, De-anonymization Attacks, Privacy Metrics - k-anonymity, I-diversity, Personalization vs Privacy, Differential Privacy, Social Media and User Trust.		



Unit Number: 4	Title: Social Media Security: Laws, Best Practices, and Case Studies	No. of hours: 10
Content Summary: Laws regarding posting of inappropriate content, Best practices for the use of Social media, Content Moderation and Removal Policies, User Authentication and Access Control, Security Awareness and Education, Social media Case studies-Facebook, Twitter, Instagram, YouTube, LinkedIn, StackOverflow, GitHub, Quora, SnapChat, Reddit, FourSquare, Yelp.		
*Self-Learning Components: <ol style="list-style-type: none"> 1. Social Media Security 101 - Stop The Hackers! 2. Privacy and Security in Online Social Media 3. CompTIA Social Media Security 		
References: <ol style="list-style-type: none"> 1. https://www.udemy.com/course/social-media-security-101-stop-the-hackers/ 2. https://onlinecourses.nptel.ac.in/noc20_cs31/preview 3. https://niccs.cisa.gov/education-training/catalog/certfirst/comptia-social-media-security 		
Please Note: At least 5-10 % syllabus will be asked in end term exams from self-learning components		
Reference Books: <ol style="list-style-type: none"> 1. Mastering Social Media Mining, Bonzanini Marco, Packt Publishing Limited 2. Mining the Social Web, Mikhail Klassen and Matthew A. Russell, O'Reilly Media, Inc 3. Social media mining: an introduction, Zafarani, Reza, Mohammad Ali Abbasi, and Huan Liu, Cambridge University Press 4. Social Media Security: Leveraging Social Networking While Mitigating Risk, Michael Cross, Syngress 5. Social Media and the Law: A Guidebook for Communication Students and Professionals, Daxton R. Stewart, Taylor & Francis Ltd 6. Security in the Digital Age: Social Media Security Threats and Vulnerabilities by Henry A. Oliver, Create Space Independent Publishing Platform. 		
Online References: <ol style="list-style-type: none"> 1. https://media.defense.gov/2021/Sep/16/2002855950/-1/-1/0/CSI_KEEPING_SAFE_ON_SOCIAL_MEDIA_20210806.PDF 2. https://www.technology.pitt.edu/security/best-practices-safe-social-networking 3. https://www.mdpi.com/1999-5903/10/12/114 		

Course Outcomes (CO)

COs	Statements
-----	------------



CO1	Demonstrate an understanding of the different types of social media platforms, their features, and their impact on communication, marketing, and society.
CO2	Acquire knowledge and skills in social media monitoring techniques, including data collection, analysis, and the use of relevant tools and technologies.
CO3	Develop the ability to analyze and evaluate viral content on social media, understand the factors contributing to its spread, and recognize its implications for marketing and online engagement.
CO4	Identify and analyze the challenges, opportunities, and pitfalls associated with social media marketing, and formulate strategies for effective audience targeting, engagement, and brand promotion.
CO5	Develop strategies to safeguard personal information, foster user trust, and mitigate associated risks.

COs Mapping with Levels of Bloom’s taxonomy

CO	Cognitive levels(C) 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A3	P1
CO2	C2	A2	P2
CO3	C4	A5	-
CO4	C4	A3	P4
CO5	C6	A4	P5

CO-PO Mapping

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

1=weakly mapped



2= moderately mapped

3=strongly mapped

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	Social Media Overview
Local	-
Regional	-
National	Provides essential knowledge and skills related to social media platforms, social media marketing, and data collection from online social media.
Global	Covers key aspects of social media platforms, social media marketing, and data analysis techniques that have global applicability
Employability	Highly valued in the job market.
Entrepreneurship	Explore entrepreneurial opportunities in the digital marketing and social media industry.
Skill Development	Enhances students' technical skills in understanding and utilizing social media effectively.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	Security Issues in Social Media
Local	Addresses local indicators by covering security issues in social media.
Regional	-
National	Provides essential knowledge and skills related to security issues in social media, which are relevant at the national level for ensuring online safety.
Global	Covers key aspects of security issues in social media that have global relevance.
Employability	Highly valued in the job market, particularly in roles related to cybersecurity, digital risk management, and social media governance.



Entrepreneurship	Equips with knowledge of security issues in social media, allowing to identify entrepreneurial opportunities in developing innovative solutions for securing social media platforms.
Skill Development	Enhances technical skills in identifying, analyzing, and addressing security threats.
Professional Ethics	Indirectly promotes professional ethics by emphasizing the importance of protecting user privacy, preventing cyberbullying, and addressing collective misbehaviors in social media platforms.
Gender	-
Human Values	Indirectly supports human values by fostering a safe and inclusive online environment.
Environment & Sustainability	-
Unit III	Privacy Issues in Social Media
Local	Addresses local indicators by covering privacy issues in social media that are relevant to the local context.
Regional	-
National	Provides essential knowledge and skills related to privacy issues in social media, which are relevant at the national level for ensuring data protection and privacy rights.
Global	Covers key aspects of privacy issues in social media that have global relevance, such as identity and attribute disclosure attacks.
Employability	Highly valued in the job market, particularly in roles related to data privacy, information security, and compliance with privacy regulations.
Entrepreneurship	Equips with knowledge of privacy issues in social media, allowing them to identify entrepreneurial opportunities in developing privacy-enhancing solutions for social media platforms.
Skill Development	Enhances technical skills in privacy settings, data protection, and privacy metrics.
Professional Ethics	Indirectly promotes professional ethics by emphasizing the importance of respecting user privacy, protecting personal information, and ensuring transparency.
Gender	-
Human Values	Indirectly supports human values by fostering a culture of privacy and user trust in social media platforms.
Environment & Sustainability	-
Unit IV	Social Media Security: Laws, Best Practices, and Case Studies
Local	Addresses local indicators by covering laws regarding posting of inappropriate content that are relevant to local jurisdictions



	and regulations.
Regional	Provides regional relevance by including case studies of popular social media platforms that are widely used in the regional context, such as Facebook, Twitter, Instagram, and LinkedIn.
National	Covering laws related to social media and best practices for the use of social media platforms
Global	Includes case studies of various global social media platforms.
Employability	Highly valued in roles related to social media management, digital marketing, content moderation, and information security..
Entrepreneurship	Equips with knowledge of social media security laws, best practices, and case studies, allowing them to identify entrepreneurial opportunities in providing social media security services
Skill Development	Enhances students' skills in content moderation, user authentication, access control, security awareness, and education.
Professional Ethics	Emphasizing the importance of adhering to social media laws.
Gender	-
Human Values	Indirectly supports human values by promoting responsible use of social media, ensuring user privacy and safety, and addressing ethical considerations.
Environment & Sustainability	-
SDG	-
NEP 2020	Digital literacy, Critical thinking, Ethical use of technology
POE/4 th IR	Technological advancements, innovation, adaptability, digital fluency, problem-solving, collaboration, and lifelong learning.

SOCIAL MEDIA SECURITY LAB

Department:	Department of Computer Science and Engineering		
Course Name: Social Media Security Lab	Course Code	L-T-P	Credits
	ENSP457	0-0-2	1



Type of Course:	Programme Elective-I (Cyber Security)
Pre-requisite(s), if any:	

Course Outcomes (CO)

COs	Statements
CO1	Understand the risks and vulnerabilities associated with social media platforms.
CO2	Understand the social and ethical implications of social media security.
CO3	Develop practical skills to secure social media accounts and data.
CO4	Analyze and respond to social media security incidents.
CO5	Evaluate the effectiveness of social media security controls.

Proposed Lab Experiments

Ex. No	Experiment Title	Mapped CO/COs
1	Exploring Different Social Media Platforms a. Research and analyze various social media platforms. b. Identify their key features, target audiences, and unique characteristics. c. Compare and contrast their usage, advantages, and challenges.	CO1, CO2, CO5
2	Monitoring Social Media Trends a. Use social media monitoring tools to track popular hashtags and viral content. b. Analyze the patterns and trends in social media conversations. c. Identify the factors contributing to the popularity of certain content.	CO1, CO5
3	Social Media Marketing Analysis a. Study real-world social media marketing campaigns. b. Analyze their strategies, target audience engagement, and impact. c. Evaluate the challenges and opportunities in social media marketing.	CO2, CO3
4	Collecting and Analyzing Social Media Data a. Utilize APIs to collect data from online social	CO3



	<p>media platforms.</p> <p>b. Perform content analysis using techniques like Bag-of-Words (BoW) model and TF-IDF.</p> <p>c. Extract insights and patterns from the collected data.</p>	
5	<p>Social Network Analysis</p> <p>a. Perform network analysis on social media data.</p> <p>b. Calculate node centrality measures, degree distribution, average path length, and clustering coefficient.</p> <p>c. Identify key influencers and community structures within the social network.</p>	CO3, CO5
6	<p>Creating Synthetic Networks</p> <p>a. Generate random graphs and preferential attachment models to simulate social networks.</p> <p>b. Analyze the characteristics of the synthetic networks.</p> <p>c. Compare and contrast them with real-world social networks.</p>	CO1, CO5
7	<p>Profile Cloning and Identity Theft</p> <p>a. Study different types of identity theft in social media.</p> <p>b. Analyze profile cloning, social phishing, and compromised accounts.</p> <p>c. Understand the behavioral patterns and impacts of these attacks.</p>	CO1, CO2
8	<p>Dealing with Spam and Misinformation</p> <p>a. Analyze the spread of spam and misinformation in social media.</p> <p>b. Identify techniques to detect and mitigate spamming activities.</p> <p>c. Evaluate the effectiveness of flagging and reporting mechanisms.</p>	CO4
9	<p>Privacy Settings Evaluation</p> <p>a. Evaluate the privacy settings of popular social media platforms.</p> <p>b. Assess the level of protection they provide for Personally Identifiable Information (PII).</p> <p>c. Propose recommendations for enhancing user privacy.</p>	CO2, CO5
10	<p>Privacy Attacks and Anonymity</p> <p>a. Study different privacy attacks in social media,</p>	CO2



	<p>such as inference attacks and de-anonymization attacks.</p> <p>b. Analyze the impact of identity disclosure and attribute disclosure attacks.</p> <p>c. Explore techniques like differential privacy for preserving user privacy.</p>	
11	<p>Privacy Metrics Analysis</p> <p>a. Investigate privacy metrics like k-anonymity and l-diversity.</p> <p>b. Apply these metrics to analyze the privacy risks in social media datasets.</p> <p>c. Discuss the trade-offs between personalization and privacy in social media.</p>	CO2
12	<p>Understanding Social Media Laws and Regulations</p> <p>a. Study the laws and regulations related to social media usage.</p> <p>b. Analyze the legal implications of posting inappropriate content.</p> <p>c. Explore content moderation policies and user responsibilities.</p>	CO2
13	<p>User Authentication and Access Control</p> <p>a. Evaluate user authentication mechanisms in popular social media platforms.</p> <p>b. Analyze access control policies and user permissions.</p> <p>c. Discuss best practices for ensuring secure user authentication.</p>	CO3
14	<p>Security Awareness and Education</p> <p>a. Develop security awareness campaigns for social media users.</p> <p>b. Design educational materials to raise awareness about social media security risks.</p> <p>c. Evaluate the effectiveness of these campaigns through surveys or assessments.</p>	CO2
15	<p>Case Study Analysis - Facebook</p> <p>a. Analyze the security and privacy practices of Facebook.</p> <p>b. Explore the challenges faced by Facebook in maintaining user data privacy.</p> <p>c. Discuss notable security incidents and their impact on user trust.</p>	CO1, CO2, CO4
16	<p>Case Study Analysis - Twitter</p> <p>a. Investigate the security measures</p>	CO1, CO2, CO4



	<p>implemented by Twitter.</p> <p>b. Analyze the response to cybersecurity incidents on the platform.</p> <p>c. Discuss the role of Twitter in addressing misinformation and cyberbullying.</p>	
17	<p>Case Study Analysis - Instagram</p> <p>a. Analyze the privacy and security features of Instagram.</p> <p>b. Investigate the effectiveness of content moderation policies.</p> <p>c. Discuss the impact of influencer marketing and brand safety on Instagram.</p>	CO1, CO2, CO4
18	<p>Case Study Analysis - YouTube</p> <p>a. Evaluate the security controls and privacy settings of YouTube.</p> <p>b. Analyze the challenges of content moderation and copyright infringement.</p> <p>c. Discuss the role of YouTube in combating hate speech and harmful content.</p>	CO1, CO2, CO4
19	<p>Case Study Analysis - LinkedIn</p> <p>a. Study the security and privacy considerations on LinkedIn.</p> <p>b. Analyze the protection of professional user data and connections.</p> <p>c. Discuss the impact of LinkedIn in job search and professional networking.</p>	CO1, CO2
20	<p>Case Study Analysis - StackOverflow</p> <p>a. Investigate the security practices implemented on StackOverflow.</p> <p>b. Analyze the trust and reputation systems within the community.</p> <p>c. Discuss the role of StackOverflow in knowledge sharing and code collaboration.</p>	CO1, CO2
21	<p>Case Study Analysis - GitHub</p> <p>a. Analyze the security measures adopted by GitHub for source code repositories.</p> <p>b. Investigate the role of vulnerability reporting and code review processes.</p> <p>c. Discuss the importance of secure coding practices in open-source projects.</p>	CO1, CO2
22	<p>Case Study Analysis - Quora</p> <p>a. Evaluate the privacy controls and content moderation on Quora.</p>	CO1, CO2



	<p>b. Analyze the impact of user-generated content and knowledge sharing.</p> <p>c. Discuss the challenges of maintaining a respectful and inclusive community.</p>	
23	<p>Case Study Analysis - SnapChat</p> <p>a. Study the privacy and security features of SnapChat.</p> <p>b. Analyze the ephemeral messaging and privacy-by-design approach.</p> <p>c. Discuss the challenges of preventing data leaks and unauthorized access.</p>	CO1, CO2
24	<p>Case Study Analysis - Reddit</p> <p>a. Analyze the security and privacy considerations on Reddit.</p> <p>b. Investigate the moderation policies and community-driven content curation.</p> <p>c. Discuss the challenges of maintaining a balance between free speech and harmful content.</p>	CO1, CO2

DEPARTMENT ELECTIVE – II

**MOBILE APPLICATION DEVELOPMENT
USING IOS**

Department:	Department of Computer Science and Engineering
--------------------	---



Course Name: Mobile Application Development using iOS		Course Code ENSP409	L-T-P 4-0-0	Credits 4
Type of Course:	Departmental Elective			
Pre-requisite(s), if any: NA				
Brief Syllabus: The Android operating system (OS) has the highest market share worldwide on mobile devices. Android held 71.93 percent of the market. It is therefore necessary for students to know that how to build mobile apps for android operating system. This course covers the necessary concepts which are required to understand mobile communication and to develop Android Applications				
UNIT WISE DETAILS				
Unit Number: 1	Title: Introduction to Mobile Computing:	No. of hours: 10		
Content Summary: Concept of Mobile Communication, Different generations of wireless technology, Basics of cell, cluster and frequency reuse concept, Noise and its effects on mobile, Understanding GSM and CDMA, Basics of GSM architecture and services like voice call, SMS, MMS, LBS, VAS, Different modes used for Mobile Communication, Architecture of Mobile Computing(3 tier), Design considerations for mobile computing, Characteristics of Mobile Communication, Application of Mobile Communication, Security Concern Related to Mobile Computing, Middleware and Gateway required for mobile Computing, Making Existing Application Mobile Enable, Mobile IP, Basic Mobile Computing Protocol.				
Unit Number: 2	Title: Introduction to Android Programming	No. of hours: 10		
Content Summary: Overview of Android, Android Internals, Android for mobile apps development, Environment setup for Android apps Development, Framework -Android-SDK, Emulators - Android AVD, Android Emulation – Creation and set up, First Android Application				
Unit Number: 3	Title: Android Activities and GUI Design:	No. of hours: 10		
Content Summary: Activity Lifecycle of Android, Design criteria for Android Application : Hardware Design Consideration, Design Demands For Android application, Intent, Activity, Activity Lifecycle and Manifest, Creating Application and new Activities, Simple UI - Layouts and Layout properties: Introduction to Android UI Design, Introducing Layouts, Fragments, Push Button , Text / Labels , Edit Text, Toggle Button , Padding				
Unit Number: 4	Title: Background Tasks	No. of hours: 10		



Content Summary:

Customizations: Floating hints and Auto Complete, Create Custom Layout, Create Custom Toast.

Save Data Locally on Phone: Save User Preferences, Save data using text files, Making use of Async Task class: Intro to Async Task Loader, load In Background() , Async Task Loader callbacks , Benefits of loaders . Connecting to data by SQL Lite Database: Overview of SQLite, Open Helper Android class, Querying (dev) Searching (user) databases, Best practices for using databases in Android, Best practices for testing your database

Permissions:The permissions model, Libraries: Using libraries, Widgets: What are widgets?, When to use them and how to implement them, Publishing your App: Different ways to monetize your app, Making and publishing APKs: Guidelines for publishing in Google Play , Make and sign the APK, Beta test your app , Publish your app to Google Play

***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/>

Please Note:

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

Text Book:

1. Reto Meier, "Professional Android Application Development", Wiley India Pvt Ltd

Reference Books:

1. Mark L Murphy, "Beginning Android", Wiley India Pvt Ltd
2. Sayed Y Hashimi and SatyaKomatineni, "Pro Android", Wiley India Pvt Ltd

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/~srsarang/archbooksoft.html>
7. XILINX(VHDL/Verilog tools):
<https://www.xilinx.com/support/university/students.html>

Course Outcomes (CO)

COs	Statements
CO1	Understand functioning of different mobile technology
CO2	Demonstrate Android Activities Life Cycles
CO3	Execute Operations on GUI objects
CO4	Perform Event Driven Programming
CO5	Apply various techniques on working with the menu

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
CO2	C3	A2	P1
CO3	C5	A2	P2
CO4	C6	A3	P4
CO5	C6	A3	P4

CO-PO Mapping

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

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

CO-PSO Mapping

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

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/4 th 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:	Departmental Elective IV (Minor)		
Pre-requisite(s), if any: Basics of Android			

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.



Proposed Lab Experiments

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:			
<p>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.</p> <ul style="list-style-type: none"> • 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: Introduction		No. of hours: 8
Content Summary:			
<p>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</p>			
Unit Number: 2	Title: : VERSION CONTROL SYSTEMS		No. of hours: 12
Content Summary:			
<p>VERSION CONTROL SYSTEMS: Overview of version control systems – role of versiocontrol systems – Types of control systems and their supporting tools – Overview of Git – Overview of Source code and Version Control hosts – Deploy</p>			



the files to GitHub.		
Unit Number: 3	Title: CONTINUOUS INTEGRATION AND BUILDING TOOL	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_project_teamcity.htm		
Unit Number: 4	Title: software testing	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. 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.		
*Self-Learning Components: Suggested Source: 2. https://www.tutorialspoint.com/puppet/index.html 3. https://puppet.com/blog/how-get-started-puppet-beginners-guide 4. https://www.tutorialspoint.com/chef/index.html 5. https://docs.chef.io/chef_overview 6. https://www.tutorialspoint.com/ansible/index.html 7. 8. https://docker-curriculum.com 9. https://howtodoinjava.com/junit-5-tutorial 10. https://junit.org/junit5/docs/current/user-guide		
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	Analyse various Testing tools
CO5	Illustrate various Configuration management tools.

COs Mapping with Levels of Bloom’s taxonomy

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

Please Note:

- Refer to POs while mapping each CO.
- Mark “ - ” if not applicable
- If attainment of a CO is strongly mapped with a PO , Mark 3
- If attainment of a CO is moderately mapped with a PO , Mark 2
- If attainment of a CO is weakly mapped with a PO , Mark 1

Justification for mapping must be relevant

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

CO-PSO Mapping

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

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/4 th 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	4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			

Defined Course Outcomes

COs	Course Outcomes (COs)
CO 1	Manage Software Version Control using GitHub
CO 2	Configure Management tools
CO 3	Analyze various Testing tools

Proposed Lab Experiments

Ex. No	Experiment Title	hours
1	Deploy three servers	2 lab hours
2	Set up static websites on two servers using Nginx. Make a small change in the index.html file of one of the websites to differentiate between two servers.	2 lab hours
3	Configure Nginx to load and balance traffic between two static websites.	2 lab hours
4	Add the Nginx Load balancer IP to the DNS A record.	2 lab hours
5	Set up Nginx on the third server. It will act as a load balancer.	2 lab hours
6	Try different Nginx load-balancing algorithms and options.	2 lab hours
7	Understand L7 load balancing	2 lab hours
8	Try accessing the website. Every time you reload the website you should see a different index.html.	2 lab hours
9	Access the DNS and validate the WordPress website setup.	2 lab hours
10	Set up the LAMP stack on the server. Configure WordPress Application	2 lab hours



.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		
<p>*Self-Learning Components:</p> <ol style="list-style-type: none"> 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. LeetCodeHackerRank 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 <p><i>*students will demonstrate the self-learning components through classroom presentations</i></p>		
Reference/Text Books:		
<ol style="list-style-type: none"> 1. "Mastering C# and .NET Framework" by Jayantha Dhanapala 2. "Pro C# and .NET Framework" by Andrew Troelsen 3. ".NET Framework Programming with C#" by G. Shankar 4. ".NET Programming: Concepts and Practice" by Atul Kumar 		

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
CO2	C3	A2	P2
CO3	C2	A3	P3
CO4	-	-	-
CO5	C5	-	P5

***Please Note:**
Map only 1 or 2 Levels in each category. If a higher level is given, no need to mention lower level

CO-PO Mapping

PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10
CO1	1	-	3	-	2	2	-	1	-	1
CO2	-	1	2	3	1	2	-	2	2	-
CO3	2	2	-	2	-	2	2	-	-	2
CO4	-	-	2	-	-	2	-	-	1	1
CO5	1	1	-	3	2	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/4 th 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:			

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.

Proposed Lab Experiments

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	CO2



	C# program.	
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

Department:	Department of Computer Science and Engineering		
Course Name: New-Age programming languages	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. 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	-
CO2	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	1	-	2	2	2	1	1	2
CO2	2	2	-	-	2	2	-	1	-	2
CO3	2	2	-	3	-	2	1	-	1	-
CO4	-	-	3	-	3	2	-	-	-	3
CO5	1	-	-	-	-	2	1	2	2	2

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

CO-PSO Mapping

PSO	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/4 th IR	Technological advancements, digital transformation, and future-ready skills.



NEW AGE PROGRAMMING LANGUAGES LAB

Department:	Department of Computer Science and Engineering		
New Age Programming languages Lab	Course Code	L-T-P	Credits
	ENSP465	0-0-2	1
Type of Course:	Minor		
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	CO1



	demonstrate their functionality and reusability.	
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	CO1



	using higher-order functions (define a list of integers or strings).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.	C03
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.	C03
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.	C04
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.	C05
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.	C01
18	Write functions that manipulate and transform sequences using operations such as map, filter, reduce, and take.	C02
19	Implement a program that showcases asynchronous programming using the core.async library.	C03
20	Write code that calls Java methods, creates Java objects, and works with Java collections and objects from Clojure.	C04
21	Develop a web application using Clojure and the Ring library. Set up routes, handle HTTP requests and responses, and render dynamic content.	C05
22	Write functions that interact with the database, perform CRUD operations, and handle transactions.	C05
23	Implement error handling mechanisms, such as exception handling and error management, in Clojure.	C04
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.	C05
Practicals on Kotlin Programming Language		
25	19 WAP for print following o/p	C02



	20	Hello Kotlin!!! 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%	
	21	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 questions.	CO5

