SCHOOL OF ENGINEERING

AND

TECHNOLOGY

B.SC (H) Data Science

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 endeavors, and experiential learning opportunities. Through industry partnerships, internships, and exposure to real-world challenges, our students gain a holistic understanding of their fields, preparing them to make meaningful contributions to society.

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

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

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

Dean,

School of Engineering and Technology,

K. R. Mangalam University.

Preface

Data Science is a distinctive cross-disciplinary field that combines computer science, computational mathematics, statistics, and management. After the theoretical, computational, and empirical paradigms of science, data science is regarded as the fourth paradigm of science. Due to the widespread use of digital devices, our daily lives generate more data than ever before. The amount of data is expanding exponentially with the introduction of IoT (Internet of Things) and Industry 4.0. Insights that can be used to improve our world are hidden in that plethora of data. Due to this, data science has developed into a discipline that entails gathering, visualizing, analysing, and modelling huge and complicated data sets from various domains and sources.

The program's main goal is to provide graduates with the skills necessary to acquire and manage all forms of data in order to conduct data-driven investigations and visual and sophisticated analyses. This 3-year undergraduate program trains students to gather, organize, and extract meaning from data for better business decision-making.

BSc (Honours) Data Science is a three-year undergraduate programme that is a perfect blend of computer science, Business analytics, and Artificial intelligence to understand large data sets and to craft new predictive models using algorithms, prototypes, and customized analysis. Data Science graduates predict features of data using a dedicated algorithm and statistical tools/ models. Students will get a deep insights into the field of big data analytics, data science, and how this industry is growing with the help of Artificial Intelligence.

The course is built around KRMU's basic ideas of offering industry-linked, technology-based, research-driven, and seamless education, just like all of our other flagship programs.

Objectives of the program

After the completion of the degree, students would

- Be prepared with a wide range of knowledge in many data science fields, such as data collecting, visualisation, processing, and modelling of massive data sets.
- Learn how to analyse large data sets coming from a variety of application domains using established data science models based on math and computer science.
- Use the knowledge gained from the curriculum to build models that may be used to tackle current and upcoming difficulties and issues needing extensive data analysis.

- Become better educated professionals to meet the industry's expanding need for data scientists and engineers.

Career Avenues

There is a large scope of B.SC (H) Data Science Jobs for graduates in both the private and public sectors. It is one of the most diverse courses in terms of, not just employment opportunities across various domains, but also the scope of higher education for graduates.

According to the IT industry, there are over 1.5 million jobs that are going unfilled in India right now. This shows that data is the future and so are Data Scientists!

Some of the areas of recruitment are

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

Prospective Companies

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

Duration

3 Years (Full-Time)

Eligibility Criteria

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

Table of Contents

S.N.	Content	Page No.
1	Preamble	2
2	Preface	3
3	Objectives of the program	4
4	Career Avenues	4
5	Duration & Eligibility Criteria	5
6	University Vision & Mission	7
7	School Vision & Mission	8
8	About School	9
9	Program Objectives (PO)	10
10	Program Educational Objectives (PEO)	10
11	Program Specific Objectives (PSO)	11
12	Program Highlights	11
13	Scheme of Studies	13
14	Categorization of Courses	18
15	Course Template	19
16	CO Mapping with Bloom's Taxonomy	20
17	CO-PO/PSO Mapping	23
18	Mapping with components of Global/National/Regional/Local aspects, employability, skill development, SDG, NEP-20 etc	25

Institution Vision & Mission

Vision

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

Mission

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

School Vision & Mission

Vision

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

Mission

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

About School

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

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

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

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

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

Program Outcome (PO)

PO1. Computational Knowledge: Demonstrate understanding and apply foundational mathematical concepts, computing principles, and domain knowledge to conceptualize computing models that address defined problems.

PO2. Problem Analysis: Possess the ability to identify, critically analyze, and formulate complex computing problems, leveraging fundamental principles from computer science and relevant application domains.

PO3. Solution Design and Development: Transform intricate business scenarios and contemporary issues into problem statements, investigate and comprehend them, and propose comprehensive and integrated solutions using cutting-edge technologies.

PO4. Effective Tool Utilization: Select and proficiently employ contemporary computing tools, techniques, and skills necessary for developing innovative software solutions.

PO5. Professional Ethics: Apply and uphold professional ethics and cybersecurity regulations within a global economic environment, ensuring responsible and ethical computing practices.

PO6. Lifelong Learning: Recognize the necessity for continuous learning and cultivate the ability to actively engage in professional development as a computing professional.

PO7. Project Management: Possess the competence to understand and apply management and computing principles in order to successfully manage projects in multidisciplinary environments.

PO8. Proficient Communication: Demonstrate effective communication skills, both within the computing community and with society at large, by comprehending and producing clear and impactful documentation and presentations.

PO9. Individual and Teamwork: Exhibit the ability to effectively collaborate and contribute as a member or leader within diverse teams operating in multidisciplinary environments.

PO10. Innovation and Entrepreneurship: Identify opportunities, demonstrate an entrepreneurial mindset, and leverage innovative ideas to generate value and contribute to the betterment of individuals and society.

Program Educational Objectives (PEO)

PEO1 - Develop a strong foundation in data science theory, concepts, and methodologies, and apply them to solve real-world problems in various domains.

PEO2 - Pursue a career in data science or related fields such as business intelligence, machine learning, artificial intelligence, data analytics, or data engineering, among others.

PEO3 - Continuously upgrade their skills and knowledge in data science through lifelong learning, research, and development of innovative data-driven solutions that contribute to society's sustainable growth.

PEO4 - Demonstrate leadership, teamwork, ethical, and social responsibilities with excellent communication skills in the data science profession, working collaboratively and effectively in diverse and multicultural environments.

Program Specific Outcomes (PSO)

PSO1 - Develop a strong foundation in data science theory, concepts, and methodologies, and apply them to solve real-world problems in various domains.

PSO2 - Pursue a career in data science or related fields such as business intelligence, machine learning, artificial intelligence, data analytics, or data engineering, among others.

PSO3 - Continuously upgrade their skills and knowledge in data science through lifelong learning, research, and development of innovative data-driven solutions that contribute to society's sustainable growth.

PSO4 - Demonstrate leadership, teamwork, ethical, and social responsibilities with excellent communication skills in the data science profession, working collaboratively and effectively in diverse and multicultural environment.

Programme Highlights

- Professionally qualified, competent and committed teaching faculty.
- Industry enabled curriculum and training from industry experts.
- Consistent interaction with renowned academicians and experts.
- Emphasis on project-based learning, techno-pedagogy, field projects, research projects, internships, continuous and comprehensive evaluation.
- Access to certification courses, ability & skill development programs, value-added courses besides core curriculum.
- Effective career counselling, guidance and mentoring program to excel in professional and personal spheres of life.
- Special programs for advanced and slow learners with focus on inclusion and student diversity.
- Focus on career progression through training, placements and preparation for higher studies.
- Centre of excellence in AI ,Machine Learning & Data Science

Program Scheme

Semester I

SN0	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	-	-	4
5	Major	ENBC151	Fundamentals of Web Technologies Lab	-	-	2	1
6	Major	ENBC153	Matlab Programming Lab	-	-	2	1
7	Minor	ENSP155	Introduction to Computer Science and Programming in Python Lab	-	-	2	1
8	VAC		Environmental Studies and Disaster Management (Online Moodle)	2	-	-	2
9	Major	ENBC105	Fundamentals of Software Engineering	4	-	-	4
TOTAL				**18**	**0**	**8**	**23**

Semester II

SN 0	Category	Course Code	Course Title	L	T	P	C
1	Minor	ENSP110	Essentials of Data Science	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
5	Minor	ENSP162	Data Science Lab	-	-	2	1
6	Major	ENBC152	Concepts of Object Oriented Programming using C++ Lab	-	-	2	1
7	Major	ENBC154	Basics of Operating Systems Lab	-	-	2	1
8	VAC		Extension Activities (Community Engagement Service)	2	-	-	2
9			Open Elective – I	3	-	-	3
TOTAL				**18**	**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	ENSP205	Fundamentals of Machine Learning	4	-	-	4
3	Major	ENBC203	Basics of Probability and Statistics	4	-	-	4
4	Major	ENBC205	Introduction to Java Programming	3	1	-	4
5	AEC	AEC011	Life Skills for Professional-I	3	-	-	3
6	Major	ENBC251	Introduction to Java Programming Lab	-	-	2	1
7	Major	ENBC253	Introduction to Data Structures Lab	-	-	2	1
8	Minor	ENSP257	Machine Learning Lab	-	-	2	1
9	VAC	ETCS257A	VAC – 3	-	-	-	2
10	Summer Internship	SIBC251	Summer Internship/Project-I	-	-	-	2
TOTAL				**17**	**2**	**6**	**26**

Semester IV

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

Semester V

S N	Category	Course Code	Course Title	L	T	P	C
1	Major	ENBC301	Computer Organization and Architecture	4	-	-	4
2	Minor	ENSP321	Natural Language Processing Using Python	4	-	-	4
3	Minor		Department Elective - I	4	-	-	4
4	Minor		Department Elective - I Lab	-	-	2	1
5	Major		Department Elective - II	4	-	-	4
6	Major		Department Elective – II Lab	-	-	2	1
7	Minor	ENSP369	Natural Language Processing Lab	-	-	2	1
8	Summer Internship	SIBC351	Summer Internship/Project	-	-	-	2
9	AEC	AEC013	Life Skills for Professional - III	3	-	-	3
TOTAL				**19**	**0**	**6**	**24**

Departmental Elective – I (Cloud Computing)						
(i)	ENSP401	Computational Services in the Cloud	4	-	-	4
	ENSP451	Computational Services in the Cloud Lab	-	-	2	1
(ii)	ENSP403	Microsoft Azure Cloud Fundamentals	4	-	-	4
	ENSP453	Microsoft Azure Cloud Fundamentals Lab	-	-	2	1
(iii)	ENSP405	Storage and Databases on Cloud	4	-	-	4
	ENSP455	Storage and Databases on Cloud	-	-	2	1

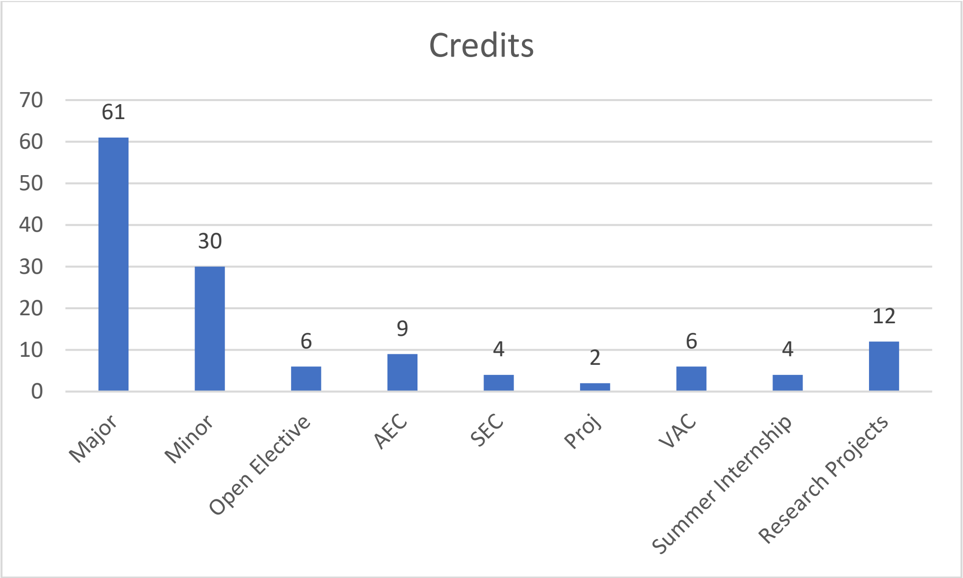
(iv)	ENSP407	Application Development and DevOps on Cloud	4	-	-	4
	ENSP457	Application Development and DevOps on Cloud Lab	-	-	2	1

Departmental Elective – II (Full Stack Development)						
(i)	ENSP409	Mobile Application Development using IOS	4	-	-	4
	ENSP459	Mobile Application Development using IOS Lab	-	-	2	1
(ii)	ENSP411	DevOps and Automation	4	-	-	4
	ENSP461	DevOps and Automation Lab	-	-	2	1
(iii)	ENSP413	.Net Framework	4	-	-	4
	ENSP463	.Net Framework Lab	-	-	2	1
(iv)	ENSP415	New Age Programming Languages	4	-	-	4
	ENSP465	New Age Programming Languages Lab	-	-	2	1

Semester VI

SN	Category	Course Code	Course Title	L	T	P	C
1	PROJECT	SIBC352	Major Project/Industrial Training	-	-	-	12
TOTAL							**12**
TOTAL CREDITS				**134**			

Categorization of Courses



COURSE TEMPLATE

Department:	**Department of Computer Applications**		
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: Basic Introduction HTML**	**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: Javascript**	**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 -

<table>
<tr><td colspan="3">Types of Style Sheets – Internal, inline and External style sheets, customizing common HTML elements, types of CSS selectors</td></tr>
<tr><td><b>Unit Number:</b>
<b>3</b></td><td><b>Title: Web design Issue</b></td><td><b>No. of hours: 12</b></td></tr>
<tr><td colspan="3"><b>Content Summary:</b> 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.</td></tr>
<tr><td><b>Unit Number:</b>
<b>4</b></td><td><b>Title: XML</b></td><td><b>No. of hours: 8</b></td></tr>
<tr><td colspan="3"><b>Content Summary:</b>
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.</td></tr>
<tr><td colspan="3"><b>Self-Learning Components</b>:
Students can additionally take the following courses
<b>https://www.mygreatlearning.com/web-development/free-courses3)</b>
<b>https://www.simplilearn.com/certifications/web-development-courses</b></td></tr>
<tr><td colspan="3">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</td></tr>
</table>

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	C2		P1
C02	C3		P2
CO3	C3		P3
CO4	C1		-
CO5	C1		P1

Program outcomes (POs)

Engineering Graduates will be able to:

PO1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3. **Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4. **Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6. **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10. **Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

CO-PO Mapping

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

1=weakly mapped

2= moderately mapped
3=strongly mapped

CO-PSO Mapping

PO	PO1	PO2	PO3	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/4th IR	Aligns with the concepts of internet telephony, multimedia applications, and SEO

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: **Programming in MATLAB**	**Course Code**	**L-T-P**	**Credits**
	ENBC103	4-0-0	4
Type of Course:	Major		
Pre-requisite(s), if any:			

Brief Syllabus:

MATLAB is a powerful software tool used in engineering, mathematics, and science for numerical computations, data analysis, and simulation. It has a user-friendly interface, supports arrays, matrices, and complex numbers, and allows scripting for automation. MATLAB includes graphing capabilities, Simulink for system modeling, and extensive mathematical functions for integration, solving equations, and transforms. It is widely used by researchers, engineers, and scientists for a variety of applications.

UNIT WISE DETAILS

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

Content Summary:

Brief Introduction, Installation of MATLAB, History, Use of MATLAB, Key features, MATLAB Window, Command Window, Workspace, Command history, Setting directory, Working with the MATLAB user interface, Basic commands, Assigning variables, Operations with variables, Data files and Data types: Character and string, Arrays and vectors, Column vectors, Row vectors, Arithmetic operations, Operators and special characters, Mathematical and logical operators, Solving arithmetic equations.

Unit Number: 2	**Title: Operations & Plots**	**No. of hours: 14**
Content Summary: Crating rows and columns Matrix, Matrix operations: Finding transpose, determinant and inverse, Solving matrix, Trigonometric functions, Complex numbers, fractions, Real numbers, Complex numbers, Working with script tools, Writing Script file, Executing script files, The MATLAB Editor, Saving m files Plotting vector and matrix data, Plot labelling, curve labelling and editing, Basic Plotting Functions, Creating a Plot Plotting Multiple Data Sets in One Graph, Specifying Line Styles and Colors, Graphing Imaginary and Complex Data Figure, Windows Displaying, Multiple Plots in One Figure, Controlling the Axes, Creating Mesh and Surface About Mesh and Surface Visualizing Subplots.		
Unit Number: 3	**Title: MATLAB Simulink**	**No. of hours: 8**
Content Summary: Introduction of Simulink, Simulink Environment & Interface, Study of Library, Circuit Oriented Design, Equation Oriented Design, Model Subsystem Design, Connect Call back to subsystem, Application. Automating commands with scripts, writing programs with logic and flow control, Control statement, Programming Conditional Statement, Writing functions, Programming, Examples		
Unit Number: 4	**Title: Symbolic Math in MATLAB**	**No. of hours: 8**
Calculus: Numerical Integration, Linear Algebra, Roots of Polynomials, Algebraic equations, Differential Equations (1st& 2nd order), Transforms (Fourier, Laplace, etc), Ordinary Differential equations, Examples of few ODEs.		
***Self-Learning Components:** **1) 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 Lawrie Brown"Computer Security: Principles and Practice" (2020).

Define Course Outcomes (CO)

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

Cos Mapping with Levels of Bloom’s taxonomy

CO	Cognitive levels© 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

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

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

Skill Development	Learning MATLAB helps individuals develop skills in programming, data manipulation, mathematical analysis, and problem-solving, fostering overall skill development.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	Operations
Local	Local researchers and students can utilize MATLAB's matrix capabilities for their projects and assignments.
Regional	-
National	-
Global	MATLAB is a globally used tool, and its matrix operations have a wide-reaching impact.
Employability	Proficiency in MATLAB's matrix operations is highly valued in various industries.
Entrepreneurship	-
Skill Development	Learning matrix operations in MATLAB helps individuals develop valuable skills in linear algebra, which are essential for data analysis, engineering, and scientific research.
Professional Ethics	-
Gender	-

Human Values	-
Environment & Sustainability	-
Unit III	MATLAB Simulink
Local	Local researchers and students can utilize Simulink for their projects related to control systems, signal processing, and circuit design.
Regional	-
National	-
Global	proficiency in Simulink can be valuable for individuals seeking opportunities in international collaborations and research projects.
Employability	Proficiency in Simulink is highly valued in engineering and technical industries.
Entrepreneurship	-
Skill Development	Learning Simulink helps individuals develop skills in system-level modeling, simulation, and design, which are essential for engineering and research tasks.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	Symbolic Math in MATLAB
Local	Understanding numerical integration techniques and their applications in calculus is relevant at the local level

	for educational institutions, research centers, and industries that deal with data analysis, optimization, and simulations.
Regional	-
National	-
Global	Its relevance extends worldwide, and its applications impact global scientific advancements.
Employability	Proficiency in calculus, numerical integration, and differential equations is highly valued in technical industries such as engineering, data analysis, and scientific research.
Entrepreneurship	-
Skill Development	Learning calculus, numerical integration, and differential equations helps individuals develop strong analytical and problem-solving skills, which are essential for scientific research and engineering tasks.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4,9,16
NEP 2020	Aligns with the objectives and principles of NEP 2020, such as: Holistic Development, Skill Development Digital Literacy
POE/4th IR	The principles of the Fourth Industrial Revolution, fostering holistic development, skill development, digital

	literacy, and addressing emerging challenges in the digital era.

COURSE TEMPLATE

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

Proposed Lab Experiments

Defined Course Outcomes

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

Ex. No	Experiment Title	Mapped CO/COs
1	Installing Linux Operating System	CO1
2	Exploring the System	CO1
3	Working with Directories and Files	CO1
4	File Manipulation and Redirection	CO1
5	Searching for Files	CO1
6	Understanding Display and Window Managers	CO1
7	User and Group Management	CO1, CO3
8	Package Management and Software Building	CO1, CO3
9	Device Management and Mounting	CO1
10	Introduction to Shell Scripting	CO2
11	Shell Scripting: Control Structures	CO2
12	Process Management and Automation	CO2

13	File Permissions and Security	CO3
14	Network Configuration and Troubleshooting	CO4
15	Linux Servers and Web Hosting	CO4
16	Network Security in Linux	CO4
17	Introduction to Emerging Technologies in Linux	CO5
18	Docker and Containerization	CO5
19	Cloud Platforms and Deployment	CO5
20	Linux for IoT and Embedded Systems	CO5
21	Data Processing with Linux Tools	CO5
22	Data Analysis with Linux Tools	CO5
23	Machine Learning with Linux	CO5
24	Big Data Analytics with Linux	CO5
25	Visualization and Reporting	CO5

Details of 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 Applications**		
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:			

Proposed Lab Experiments

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.

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

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

16	Program to generate a random password of a given length.	CO1, CO2
17	Program to implement the Gauss-Seidel method to solve a system of linear equations. .	CO 5
18	Program to implement a simple calculator with basic arithmetic operations.	CO 4
19	Program to simulate a simple dice rolling game	CO 2
20	Mini Project: Develop a MATLAB program to perform basic image processing operations such as image enhancement, filtering, edge detection, and image segmentation. Apply these techniques to analyze and manipulate images.	CO2, CO3, CO5
21	Mini Project: Design a MATLAB program to process and analyze signals, such as audio signals or ECG signals. Implement filtering, noise reduction, Fourier analysis, and plotting of signal waveforms.	CO2, CO3,CO4, CO5
22	Mini Project: Implement various numerical methods in MATLAB, such as solving systems of linear equations, finding roots of nonlinear equations, numerical integration, and solving ordinary differential equations. Apply these methods to solve engineering and scientific problems.	CO2, CO3, CO5
23	Mini Project: Use MATLAB to analyze and visualize data from real-world datasets. Perform statistical analysis, data interpolation, curve fitting, and generate meaningful visualizations such as plots and graphs to present the results.	CO2, CO3,CO4, CO5
24	Mini Project: Build a MATLAB program to simulate control systems. Design and analyze feedback control systems, implement controllers, and simulate the system's response. Evaluate the stability and performance of the control system under various scenarios.	CO2, CO3, CO5

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: Introduction to Computer Science and Programming in Python	**Course Code** :ENSP107	**L-T-P**	**Credits**
		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	-	-
C02	C2	-	-
CO3	C3	A4	P3
CO4	C3	A4	P4
CO5	C4, C5	A5	P5

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	-

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

CO-PSO Mapping

PO	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/4th IR	The syllabus prepares students for the Fourth Industrial Revolution by providing them with essential programming and computational skills.

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

Proposed Lab Experiments

Defined Course Outcomes

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

Ex. No	Experiment Title	Mapped CO/COs
1	Create a webpage that prints the message when was this webpage created and set the title homepage as home	CO1
2	Create a web page that depicts the heading tags	CO2
3	Create a web page that displays the formatted text	CO3
4	Create a web page that display the introduction of student course	CO2
5	Create a web page that uses <pre> tag for displaying the relevant content to history of HTML	CO1
6	Create a web page that displays the content .Use the <br> tag at appropriate places.	C03
7	Create a web page that should display computer science portal and sets the background as green	CO1
8	Create a web page that should display "HTML is used for creating web page" .Set the font size as 2 and color should be blue and the face should be Calibri	CO3

9	Create a web page that include <hr> tag for displaying content	CO3
10	Create a web page that displays different kind of attributes such as image tag and anchor tag	CO2
11	Create a webpage that displays item with their corresponding images in a table	CO1
12	Create a web page that that print H1 level heading followed by horizontal line whose width is 100% below the horizontal line print the three chemical formulas and three mathematical formulas and also use line break	CO3
13	Create a web page that insert an image and align them right,left,centre and each image should have title	CO4
14	Create a webpage that displays student tabbledata in it.	CO2
15	Create a webpage that displays a table with rowspan and colspan attribute	CO1
16	Create a webpage that displays your time table using table tag.	CO2
17	Create a webpage and use hypereferences(<a>) in it.	CO4
18	Create a web page that shows the scrolling text and scrolling image using marquee tag	CO3
19	Create a web page that link to five different websites that should open in a new window	CO4
20	Create a webpage that displays unordered list and customize it.	CO2
21	Create a webpage that displays the use of form and label tag.	CO1
22	Create a webpage that shows all the three ways to include a CSS file to an html file	CO2
23	Change the background colour of a webpage using inline CSS	CO3
24	Create a webpage that displays texts in different colours using CSS	CO4
25	Create a webpage that displays fonts in different sizes using CSS properties	CO2
26	Create a webpage that displays different alignment options on a text using CSS	CO1
27	Add border to a table and an image using CSS	CO3
28	Create a web page that displays ordered and unordered list using CSS	CO2
29	Write a java script program that displays output in 4 different formats.	CO1
30	Write a program to include java script code using embedding style method.	CO3

COURSE TEMPLATE

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

Proposed Lab Experiments

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.

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

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
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© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C2	A3	P1
C02	C3, C4	A4	P2
CO3	C3	A2	P3
CO4	C3	A2	P4
CO5	C6	A5	P3

CO-PO Mapping

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

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	**Introduction**
Local	Can help students understand the local software industry and its specific challenges.
Regional	Can provide insights into the regional software development practices and challenges.
National	Address the broader context of software engineering within a country, including its impact on the economy and society
Global	Explore the global nature of software development and its impact on various industries and sectors worldwide.
Employability	Provide students with a foundational understanding of software engineering concepts and practices, which are valuable skills in the job market.
Entrepreneurship	Provide insights into the software industry, its challenges, and potential opportunities for innovation and business ventures.
Skill Development	Introduce fundamental concepts and techniques used in software engineering.
Professional Ethics	Consideration of ethical issues in software development, such as privacy, security, and responsible use of technology.
Gender	-
Human Values	Impact of software on individuals, societies, and ethical considerations related to human well-being.
Environment & Sustainability	-
Unit II	**Process Models and Program Design Techniques**
Local	Help in assessing the complexity and quality of software developed within the local context.
Regional	Provide insights into the software development practices and trends within a specific region.
National	contribute to evaluating software quality and productivity within a country's software industry.
Global	Provide standardized measures for assessing software complexity and quality, regardless of the geographical location.

Employability	Commonly used in software development organizations to measure productivity, quality, and project estimation.
Entrepreneurship	Evaluating the feasibility, cost estimation, and risks associated with software development projects.
Skill Development	By enhancing the ability to measure, analyze, and improve software quality and productivity.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Verification and Validation**
Local	Provide practical knowledge and techniques for testing software developed within the local context.
Regional	Address common testing challenges and practices in software development within the region.
National	Provide essential knowledge and skills required for testing software developed within the country.
Global	Testing is an integral part of software development across different countries and industries worldwide.
Employability	As software testing skills are in high demand by employers seeking quality assurance in software development projects.
Entrepreneurship	Provide knowledge and techniques for ensuring the quality and reliability of software products developed by entrepreneurs.
Skill Development	Introduce essential concepts, methodologies, and tools used in software testing.
Professional Ethics	Addressing ethical considerations in software testing, such as ensuring impartiality, confidentiality, and integrity in the testing process.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Software Project Management**
Local	Standardized techniques can be employed by local software development teams.

Regional	Provide a common language and methodology for software development, facilitating collaboration and communication among regional software development teams.
National	Provide a standardized framework for software development, promoting consistency and interoperability among national software projects.
Global	Widely adopted internationally, allowing for effective communication and collaboration among software development teams across different countries.
Employability	Commonly used in industry, and proficiency in these techniques is valued by employers.
Entrepreneurship	Aiding entrepreneurs in planning, designing, and communicating their software ideas.
Skill Development	Enhancing students' proficiency in software modeling and design.
Professional Ethics	Address the importance of developing reliable software and adhering to quality standards in the software engineering profession.
Gender	-
Human Values	-
Environment & Sustainability	Development of reliable software that reduces wastage, energy consumption, and potential negative environmental impacts.
SDG	SDG 4
NEP 2020	-
POE/4th IR	Emphasizes the responsible and ethical development and deployment of the systems.

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: Essentials of Data Science	**Course Code** :ENSP110	**L-T-P**	**Credits**
		4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			

Brief Syllabus:
This course introduces the fundamental concepts and techniques of data science. The course focuses on the different stages of the data science lifecycle, including data acquisition, pre-processing, analysis, and interpretation. The course will enable the students to gain hands-on experience with programming languages such as Python, data manipulation libraries like Pandas, and popular machine learning frameworks. By the end of the course, students will be equipped with the skills necessary to tackle real-world data science problems and make data-driven decisions.

UNIT WISE DETAILS

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

Content Summary:

- Chapter - 1

➢ Data Science Landscape

- ■ Data Science Overview
- ■ Data Science Domains
- ■ Data Science Roles

Chapter - 2

➢ Data Science Methodology

- ■ Data Analytics in Practice
- ■ Data Analytics Methodologies
- ■ Data Science Method

Chapter - 3

➢ Data Science on the Cloud

- ■ Integrated environment for Data Science projects
- ■ Cloud-based Data Science Lifecycle
- ■ Data Science capabilities on the cloud

Unit Number: 2	Title: Data Exploration and Visualization	No. of hours: 8
Content Summary: Chapter - 1 ➢ Explore and Prepare Data ■ Business understanding ■ Explore data ■ Prepare data ■ Understanding data Chapter - 2 ➢ Represent and Transform Data ■ Statistics and representation techniques ■ Data transformation ■ Represent and transform unstructured data ■ Data transformation tools Chapter - 3 ➢ Data Visualisation and Presentation ■ Decision-centered Visualization ■ Fundamentals of Visualizations ■ Common Graphs ■ Common Tools		
Unit Number: 3	**Title: Machine Learning Algorithms**	**No. of hours: 8**
Content Summary: Chapter - 1 ➢ Data Modelling ■ Overview of modeling techniques ■ Machine learning techniques ■ Accuracy, precision and recall ■ Model Deployment ❖ Chapter - 2 ➢ Machine Learning Algorithms ■ About Machine Learning ■ From Regression to Neural Nets ■ Decision Tree Classifier ■ Machine Learning Framework		

***Self-Learning Components**:*

- Big Data
- Distributed Computing
- Text Classification
- Sentimental Analysis

Reference Books:

1. Introduction to Machine Learning with Python: A Guide for Data Scientists by Andreas C. Mueller
2. Think Stats: Probability and Statistics for Programmers by Author: Allen B. Downey
3. Data Science for Beginners, by Andrew Park
4. The Art of Data Science — A Guide for Anyone Who Works With Data, by Roger D. Peng and Elizabeth Matsui

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** the fundamental concepts and principles of data science.
CO2	**Understand** the fundamental concepts and principles of data analytics.
CO3	**Apply** data exploration and visualization techniques.
CO4	**Implement** machine learning 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	Receiving	P1

CO2	C2	Receiving	P1
CO3	C3	Responding	P2
CO4	C4	Organizing	P2

CO-PO Mapping

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

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	Introduction
Local	Yes
Regional	Yes
National	Yes
Global	Yes
Employability	Yes
Entrepreneurship	Yes
Skill Development	Yes
Professional Ethics	-

Gender	-
Human Values	-
Environment & Sustainability	Yes
Unit II	Data Exploration and Visualization
Local	-
Regional	-
National	-
Global	Yes
Employability	Yes
Entrepreneurship	Yes
Skill Development	Yes
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	Machine Learning Algorithms
Local	-
Regional	-
National	-
Global	Yes
Employability	Yes
Entrepreneurship	Yes
Skill Development	Yes
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	The National Education Policy (NEP) 2020 emphasizes the integration of multidisciplinary and flexible approaches to education. In the context of a Data Science course, compliance with NEP 2020 can be achieved by incorporating interdisciplinary elements that bridge the gap between data science and various domains such as business, social sciences, healthcare,

	or engineering. The course focuses on developing critical thinking, problem-solving, and analytical skills, along with a strong foundation in statistical analysis, machine learning, and data visualization techniques. Additionally, the course also promotes hands-on experiential learning through real-world projects, internships, or industry collaborations, providing students with practical exposure to data science applications.
POE/4th IR	Aligns with the concepts of 4th IR.

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

Define Course Outcomes (CO)

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

COs Mapping with Levels of Bloom's taxonomy

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

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

Relevance of the Syllabus to various indicators

Unit I	Introduction
Local	-
Regional	-
National	-
Global	It lays a solid foundation for further studies in mathematics, computer science, and related fields while fostering critical thinking and analytical skills.
Employability	Equips with problem-solving techniques to analyse and process data, design algorithms, and make informed decisions.
Entrepreneurship	-
Skill Development	Discrete mathematics allows students to think abstractly, develop formal mathematical arguments, and engage in rigorous problem-solving.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	Counting, Mathematical Induction and Discrete Probability
Local	-
Regional	-
National	-
Global	Probability, Bayes' theorem, and statistical analysis provide a framework for understanding and interpreting real-world phenomena that involve uncertainty and data.
Employability	It is beneficial in areas such as probability theory, statistics, optimization, cryptography, and network analysis
Entrepreneurship	skills obtained are valuable in various fields, including computer science, mathematics, law, and philosophy.
Skill Development	Enhances your ability to analyze problems logically, identify patterns, and draw logical conclusions. These skills are valuable in various fields, including computer science, mathematics, law, and philosophy.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	Group Theory

Local	-
Regional	-
National	-
Global	Group theory is widely used in physics, chemistry, crystallography, and other fields where symmetry is a fundamental concept.
Employability	This develops ability to think conceptually, make connections between different mathematical structures, and develop a broader perspective on mathematics as a whole.
Entrepreneurship	-
Skill Development	Group theory, in particular, is essential for studying symmetry and transformations. It provides a framework for analysing the symmetries of objects, understanding transformational properties, and solving problems related to symmetry.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	Graph Theory
Local	-
Regional	-
National	-
Global	By studying these topics, the students will gain the ability to model and analyse various real-world scenarios, including social networks, transportation networks, communication networks, and data dependencies.
Employability	Understanding concepts such as shortest paths, network connectivity, and digraphs allows students to design efficient and reliable routing algorithms, analyze network performance, and ensure optimal data
Entrepreneurship	-
Skill Development	Graph theory provides a powerful framework for representing and analyzing relationships between objects or entities.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 9
NEP 2020	-
POE/4th IR	Aligns with the concepts of Design, Efficiency ,Problem Solving

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: Basics of Operating System	**Course Code**	**L-T-P**	**Credits**
	ENBC104	3-1-0	4
Type of Course:	Major		

Pre-requisite(s), if any: Basics of programming

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

UNIT WISE DETAILS

Unit Number: 1	**Title: Introduction to OS**	**No. of hours: 6**

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

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

Content Summary:

Processes: Definition, Process Relationship, Different states of a Process, Process State transitions, Process Control Block (PCB), Context switching.
Thread: Definition, Various states, Benefits of threads, Types of threads, Concept of multithreads.
Process Scheduling: Basic Concept, Type of Scheduling (Preemptive Scheduling, Non-preemptive Scheduling), Scheduling criteria: CPU utilization, Throughput, Turnaround Time, Waiting Time, Response Time.

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

Content Summary:

Memory Management: Address Binding, Dynamic Loading and Linking Concepts, Logical and Physical Addresses, Contiguous Allocation, Fragmentation, Paging, Segmentation, Combined Systems, Virtual Memory, Demand Paging, Page fault, Page replacement algorithms, Global Vs Local Allocation, Thrashing, Working Set Model.

Unit Number: 4	Title: Process-Synchronization & Deadlocks	No. of hours: 10

Content Summary:

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

Definition of Deadlocks, Necessary and sufficient conditions for Deadlock, Deadlock Prevention, Deadlock Avoidance: Banker's algorithm, Deadlock detection and Recovery.

***Self-Learning Components:**

1. Case study on UNIX and WINDOWS Operating System.
2. Practice of System calls
3. Students can refer the following book as well:

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

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

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

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

Reference Books:

1. Silbersachatz and Galvin, "Operating System Concepts", Pearson
2. Tannenbaum, "Operating Systems", PHI, 4th 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, 5th 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	C3	A2	P2
CO5	C5	A4	P4

CO-PO Mapping

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

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	**Introduction to OS**
Local	Can help students to build a strong foundation in computer science.
Regional	
National	Widely used across industries and organizations

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

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: Basic of Operating System Lab	**Course Code** : ENBC154	**L-T-P**	**Credits**
		0-0-2	1
Type of Course:	Major		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

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

List of Programs

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

14	Write a Python program to implement the Producer-Consumer problem using semaphores using UNIX/Linux system calls.	CO3
15	Write Python programs to illustrate the following IPC (Inter-Process Communication) mechanisms: a) Pipes	CO3
16	Write Python programs to illustrate the following IPC (Inter-Process Communication) mechanisms: a) FIFOs (Named Pipes)	CO3
17	Program to implement process synchronization using semaphores in Python.	CO4
18	Program to implement a basic Fo5ile allocation strategy like sequential file allocation in Python.	CO5
19	Program to demonstrate the use of signals in Python for process management.	CO1
20	Program to create and manipulate threads in Python.	CO3
21	Program to implement memory management techniques (e.g., paging, segmentation) in Python.	CO5
22	Program to simulate file system operations (e.g., open, read, write, close) in Python.	CO1
23	Program to implement process synchronization using mutex locks in Python.	CO4
24	Program to simulate the working of virtual memory in Python.	CO5
25	Program to simulate disk file management operations (e.g., allocation, deallocation) in Python.	CO5
26	Program to implement file locking mechanisms (e.g., advisory, mandatory) in Python.	CO5
27	Write a Python program to simulate the following file organization techniques Two level directories	CO5
28	Write Python programs to simulate the paging in memory management techniques	CO5
29	Write Python programs to simulate the segmentation in memory management techniques	CO5
30	Write a Python program to simulate the following file organization techniques Single level directory	CO5

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: **Concepts of Object Oriented Programming using C++**	**Course Code** :ENBC106	**L-T-P**	**Credits**
		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: 10**
Content Summary: Procedure Oriented and Object Oriented Approach. Basic Concepts: Objects, classes, Principals like Abstraction, Encapsulation, Inheritance and Polymorphism. Dynamic Binding, Message Passing. Characteristics of Object-Oriented Languages, Functions, Returning values from functions, Data Types		
Unit Number: 2	**Title: CLASSES AND OBJECTS**	**No. of hours: 10**
Content Summary: Abstract data types, Object & classes, attributes, methods, C++ class declaration, Local Class and Global Class, State identity and behaviour of an object, Local Object and Global Object, Scope resolution operator, Friend Functions, Inline functions, Constructors and destructors, instantiation of objects, Types of Constructors, Static Class Data, Array of Objects, Constant member functions and Objects, Memory management Operators.		
Unit Number: 3	**Title: INHERITANCE & POLYMORPHISM**	**No. of hours: 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
C02	C3	A2	P2
CO3	C3	A5	P5
CO4	C6	A5	P5

CO-PO Mapping

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

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

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

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

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

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: **Concepts of Object Oriented Programming using C++** Lab	**Course Code** : ENBC152	**L-T-P**	**Credits**
		0-0-2	1
Type of Course:	Major		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

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

Ex. No	Experiment Title	Mapped CO/COs
1	Write a program for Functions with default arguments	CO1
2	Simple Classes for understanding objects, member functions and Constructors Classes with primitive data members	CO2
3	Write a program for Classes with constant data members, Classes with static member functions	CO2
4	Write a program for Classes with pointers as data members – String Class	CO3
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	CO4
7	Write a Program to illustrate New and Delete Keywords for dynamic memory allocation	CO3

8	Write a Program Containing a Possible Exception. Use a Try Block to Throw it and a Catch Block to Handle it Properly.	CO4
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	
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.	CO3
12	Write a program to illustrate the concept of one class with two objects by taking student data.	CO4
13	Write a program to show the relationship of class and object to display roll no., grade and fee paid by student.	CO2
14	Write a program to define the member function outside and inside the class.	CO3
15	Write a program to read and display the information of N persons to illustrate the concept of array of objects.	CO1
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.	CO3
18	Write a program to show the order of constructor and destructor.	CO1
19	Write a program to add two numbers using binary operator overloading.	CO2
20	Write a program to illustrate the assignment operator overloading.	CO3
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.	CO4
22	Write a program to find the factorial of a number using inheritance	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.	CO4

24	Write a program to explain the relationship of inheritance and virtual function.	CO2
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.	CO1

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: Data Science Lab	**Course Code** : ENSP162	**L-T-P**	**Credits**
		0-0-2	1
Type of Course:	Minor		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Understand & learn numbers of Data Science related Phases on cloud
CO 2	Implementing Data Exploration and Understanding on any Data.
CO 3	Implementing Data Preparation and Conversion on any data.
CO 4	Understand various ML Algorithm/Estimator.

Ex. No	Experiment Title	Mapped CO/COs
1	Introduction to IBM Watson Studio.	CO1
2	To Study About Data Refinery Tool in IBM Watson Studio.	CO2
3	Explore and Understand Data using IBM Watson Studio.	CO 2
4	Data Preparation and Conversion using IBM Watson Studio.	CO3
5	To Study about different type of Visualization graph using Data Refinery Visualization features.	CO3
6	To Study About Different ML Algorithm/Estimator using scikit-learn.	CO4
7	Introduction to AutoAI.	CO4
8	Build Models using AutoAI and study about their different Hyperparameters.	CO4
9	Deploying AutoAI Model.	CO4

10	Do Fraud Analysis on Auto Insurance Datasets.	CO1
11	Guided Projects: **https://tinyurl.com/guided-project**	CO4

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
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	**Implement** a given Search problem (Linear Search and Binary Search).
CO3	**Demonstrate** an understanding of how data structures are implemented and their logical organization.
CO4	**Design & implement** the algorithm for Selection Sort, Bubble Sort, Insertion Sort, Quick Sort, Merge Sort, Heap sort. Compare their performance in term of Space and time complexity

COs Mapping with Levels of Bloom's taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C1	A3	P5

C02	C2	A3	P4
CO3	C3,C4	A4	P3
CO4	C5	A2	P2

***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	-	-	-	-
CO4	3	3	3	-	-	3	-	-	-	-

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	**Introduction to Data Structure**
Local	-
Regional	-

National	It provides foundational knowledge in data structures and algorithm analysis, which are fundamental concepts in computer science and software engineering.
Global	The principles taught in this course are applicable worldwide and form the basis of software engineering practices globally.
Employability	Understanding these concepts is crucial for technical interviews and can enhance job prospects in various technology companies.
Entrepreneurship	It helps in developing efficient and scalable software solutions, which are essential for building successful tech startups or innovative ventures.
Skill Development	Develop skills that are fundamental to computer science and software development and can be applied in various programming languages and contexts.
Professional Ethics	Applying good coding practices and software engineering principles align with professional ethics in the field.
Gender	-
Human Values	It encourages students to approach problems analytically and develop efficient solutions that can positively impact human lives.
Environment & Sustainability	-
Unit II	**Stacks, Queues and Linked List**
Local	-
Regional	-
National	These data structures are used extensively in computer science and software engineering, and the skills learned in this course can be applicable to various industries and sectors across the country.
Global	The principles taught in this course are applicable worldwide and form the basis of software engineering practices globally.
Employability	Understanding these data structures and their operations is important for solving problems efficiently and implementing optimized algorithms, which are highly sought after skills in the job market.
Entrepreneurship	Knowledge of data structures like stacks, queues, and linked lists is valuable for entrepreneurship in the technology sector.

Skill Development	Understanding and applying concepts related to stacks, queues, and linked lists enhances programming skills and helps in developing efficient algorithms to solve real-world problems.
Professional Ethics	-
Gender	-
Human Values	Fostering critical thinking, problem-solving skills, and logical reasoning, which are important qualities in a technology-driven society which can improve productivity and streamline processes, thus positively impacting human lives.
Environment & Sustainability	-
Unit III	**Trees and Graphs**
Local	-
Regional	-
National	The skills learned in this course can be applicable to various industries and sectors across the country.
Global	The principles taught in this course are applicable worldwide and form the basis of software engineering practices globally.
Employability	Knowledge of data structures such as trees and graphs is highly relevant to employability in the field of software development and computer science.
Entrepreneurship	These data structures are commonly used in designing and developing software solutions, and understanding their implementation and applications can help entrepreneurs build innovative and scalable products.
Skill Development	Understanding and applying concepts related to trees and graphs enhances programming skills and helps in developing efficient algorithms to solve real-world problems.
Professional Ethics	Following best practices in data structure implementation and algorithm design promotes code readability, maintainability, and overall software quality.
Gender	-
Human Values	Understanding data structures like trees and graphs enables students to develop efficient algorithms that

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

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: **Fundamentals of Machine Learning**	**Course Code** :ENSP205	**L-T-P**	**Credits**
		4-0-0	4
Type of Course:	Minor		
Pre-requisite(s), if any:			
Brief Syllabus: Help student understand what machine learning is. How business can use machine learning in different domains to gain competitive advantage. Student is able to differentiate between different learning algorithms. To understand different data science processes, tools and techniques. Gain a fundamental understanding of the concepts and techniques that underpin machine learning algorithms			
Unit Number: 1	**Title: : Introduction to Machine Learning**	**No. of hours: 10**	
Content Summary: Learning systems, real world applications of machine learning, why machine learning, variable types and terminology, function approximation, **Types of machine learning:** Supervised learning, unsupervised learning, Reinforcement learning **Linear Regression:** Weights and Features, Applications, Cost Functions, Finding best fit line, Gradient Descent Algorithm: Learning Algorithm, First order derivatives, Linear regression using gradient descent, Learning rate, Logistic Regression, Sigmoid Function, Cost Function for Logistic Regression, Multi-class classification, Probability Distribution, SoftMax Function, Polynomial Regression. **Performance Metrics:** Classification (Confusion Matrix, Accuracy, Precision, Recall, F1-score, ROC-AUC), Regression (MSE, MAE, RMSE, R2 Score).			
Unit Number: 2	**Title: Supervised Learning**	**No. of hours: 10**	
Content Summary: Decision Tree, Selecting Best Splitting Attribute, CART (Gini Index). ID3 (Entropy, Information Gain), Hyperparameters in Decision tree, Issues in Decision tree learning. Overfitting and Underfitting, Bias and Variance, Cross Validation.			

Bootstrap and Aggregation, Random Forest. Feature Engineering, Feature Selection, Feature Extraction. Artificial Neural Network, Neural network representation, Perceptron model, Stepwise v/s Sigmoid function, Multilayer perceptron model, Matrix Calculus (Jacobian, Hessian Matrix), Computation Graph, Backpropagation Algorithm, Activation Functions, Stochastic Gradient Descent, Batch Gradient Descent, Overfitting Problem, Regularization (Ridge, Lasso, Elastic). Bayesian Learning: Bayes theorem and concept learning, Naïve Bayes classifier, Gibbs Algorithm, Support Vector Machines, Hyperplane, Support Vectors, Kernels, K-nearest neighbour.		
Unit Number: 3	**Title: Unsupervised Learning**	**No. of hours: 10**
Content Summary: **Unsupervised learning** (clustering, Association rule learning, Dimensionality reduction), Common distance Measures, k-means clustering, Elbow method, Hierarchical Clustering – agglomerative and divisive, Dendogram, Similarity measures for hierarchical clustering, DBSCAN, Cluster Quality (R index, Silhouette Coefficient), Dimensionality Reduction, Principal Component Analysis, T-distributed Stochastic Neighbour Embedding.		
Unit Number: 4	**Title: : Evolutionary Algorithms and Deep Learning**	**No. of hours: 10**
Content Summary: **Population Based Algorithms:** Genetic Algorithm, Fitness Function, Selection, Crossover, Mutation, Swarm Optimization, Particle Swarm Optimization, Ant-Colony Optimization, Reinforcement Learning, Actors, State, Reward Policy, Actions, Convolutional Neural Networks, Deep Learning for Sequential Data, Recurrent Neural Network, LSTM.		
Self-Learning Component: - **Note: -**It is expected that the students will design, analyse, and implement the programs that learn from experience. **Self-Learning Component: -** The students are expected to work on a project based on supervised learning/unsupervised learning/Evolutionary algorithms and present the progress at the end of the semester.		
Reference Books: 1. Machine Learning by Tom M. Mitchell - McGraw Hill Education; First edition. 2. E. Alpaydin, Introduction to Machine Learning (4 ed.), Phi, 2020. ISBN 978-8120350786		

Define Course Outcomes (CO)

COs	**Statements**
CO1	Define the fundamental concepts and principles of artificial intelligence (AI) and machine learning (ML).

CO2	Apply supervised learning algorithms such as linear regression, logistic regression, decision trees, and support vector machines.
CO3	Analyze the strengths and limitations of different ML algorithms and approaches.
CO4	Design ML pipelines for solving real-world problems.

COs Mapping with Levels of Bloom's taxonomy

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

CO-PO Mapping

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

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

Relevance of the Syllabus to various indicators

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

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

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

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: Basics of Probability And Statistics	**Course Code**	**L-T-P**	**Credits**
	ENBC203	4-0-0	4
Type of Course:	Major		
Pre-requisite(s), if any: Basics of Probability and Statistics			
Brief Syllabus: The Probability and Statistics course is designed to provide students with a strong foundation in the principles and applications of probability and statistics in the context of data science. The course will cover various topics, including probability functions, random variables, discrete and continuous distributions, correlation and regression analysis, central limit theorem, and modeling uncertainty. Students will also explore real-world examples and utilize programming languages for statistical analysis and data visualization.			
UNIT WISE DETAILS			
Unit Number: 1	**Title: Basic Probability**	**No. of hours: 8**	
Content Summary: Definition of probability, conditional probability, independent events, Bayes' theorem, Bernoulli trials, Random variables, discrete random variable, probability mass function, continuous random variable, probability density function, cumulative distribution function, properties of cumulative distribution function, Two dimensional random variables and their distribution functions, Marginal probability function, Independent random variables.			
Unit Number: 2	**Title: Probability Distributions for Data Science**	**No. of hours: 8**	
Content Summary: Binomial distribution, Poisson distribution, Poisson approximation to the binomial distribution, Normal distribution and its properties, Exponential distribution, Gamma distribution, Evaluation of statistical parameters for these distributions.			

Unit Number: 3	**Title: Descriptive Statistics for Data Science**	**No. of hours: 8**
Content Summary: Measures of central tendency: mean, median, mode, Measures of dispersion: variance, standard deviation, range, Skewness and kurtosis, Moments, Expectation, Linear correlation and correlation coefficient, Rank correlation coefficient.		
Unit Number: 4	**Title: Statistical Inference for Data Science**	**No. of hours: 8**
Content Summary: Hypothesis formation and testing, large sample tests for proportions, means, and standard deviations, Small sample tests: t-test, F-test, chi-square test, Test of significance for correlation coefficients, Goodness of fit tests, Independence of attributes tests.		
Unit Number: 5	**Title: Curve Fitting and Regression for Data Science**	**No. of hours: 8**
Content Summary: Curve fitting using the method of least squares, Fitting straight lines, parabolas, and general curves, Correlation analysis: coefficient of correlation, rank correlation, Simple linear regression: regression coefficients, lines of regression, Multiple linear regression: coefficient of multiple correlation, multiple regression equations.		

***Self-Learning Components:**

- **Probability Simulation**: 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.
  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	C1	-	-

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

CO-PO Mapping

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

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).

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
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):			
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.			
Total lecture, Tutorial and Practical Hours for this course:			
Lectures:42	**Practice**		
	Tutorials:	**Lab Work:**	
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 & Event Handling**	**No. of hours: 8**
Basics Streams, Byte and Character Stream, predefined streams, Reading and writing from console and files. Event handling Mechanism, Event Model, Event Classes, Sources of Events, Event Listener Interfaces, Java GUI Programming: Introduction to Swing, Swings components, Generics and Collections: Generics and type parameters, Collections framework (List, Set, Map)		
***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 Reference||, Oracle Press. 2. Cay S. Horstmann, —Core Java Volume – I Fundamentals||, Pearson.		

Define Course Outcomes (CO)

COs	Statements
CO1	Recognize features of object-oriented design such as encapsulation, polymorphism inheritance and composition of systems based on object identity.

CO2	Articulate re-usable programming components using Abstract Class, Interfaces and other permitted ways in packages.
CO3	Apply access control mechanism to safeguard the data and functions that can be applied by the object.
CO4	Design GUI applications using pre-built frameworks available in Java.

COs Mapping with Levels of Bloom's taxonomy

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

CO-PO Mapping

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

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

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4

CO1	3	2	2	1
CO2	2	3	2	1
CO3	2	3	2	1
CO4	2	3	2	1

Relevance of the Syllabus to various indicators

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

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

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

COURSE TEMPLATE

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

Proposed Lab Experiments

Defined Course Outcomes

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

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.	CO1

	c. WAP to find sum of two float numbers taken as command line arguments d. WAP to find changed case of entered character. e. WAP to find maximum of 3 integer numbers taken as input from user at runtime.	
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	CO4

	features like a timer, question randomization, and a scoring system.	
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

COURSE TEMPLATE

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

Proposed Lab Experiments

Defined Course Outcomes

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

List of Programs

Ex No	Experiment Title	Mapped CO/COs
1	Prediction using simple linear regression	CO1
2	Prediction using multiple linear regression	CO1
3	Classification using Logistics regression	CO1

4	Classification using linear discriminant analysis	CO1
5	Classification using support vector machine.	CO2
6	Classification using Guassian Naïve Bayes	CO2
7	Classification using decision Tree	CO2
8	Classification using Random Forest.	CO1
9	Classification using K nearest neighbour.	CO4
10	Write a program to Retrieve Data for a machine Learning project.	CO3
11	Write a program to Conduct Exploratory Data Analysis using Python	CO3
12	Write a program to Clean the Data using Python	CO4
13	Write a program for Data Modeling using Python	CO4
14	Write a program to implement multiple linear regression.	CO2
15	Write a program to scale the data and implement linear regression using sklearn.	CO2
16	Write a program to implement multiple logistic regression.	CO2
17	Write a program for graphical representation of data.	CO1
18	Write a program to implement genetic algorithms.	CO4
19	Write a program to implement CNN.	CO3
20	Write a program to implement LSTM.	CO3

COURSE TEMPLATE

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

Proposed Lab Experiments

Defined Course Outcomes

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

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

	in a single graph to justify its observed time complexities.	
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.	

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

Unit Number: 3	**Title: Basic number system**	**No. of hours: 6**
Content Summary: Divisibility, Unit digit, Last two digit, Remainder, Number of zero, Factor, LCM & HCF, Simplification, Mixture, Average, Ratio, and Partnership.		
Unit Number: 4	**Title: Number system**	**No. of hours: 6**
Content Summary: Factor, LCM & HCF, Simplification, Mixture, Average, Ratio, and Partnership.		
Unit Number: 5	**Title: Time Management**	**No. of hours: 6**
Content Summary: Time management strategies, setting goals, organizing, and planning ahead, Making the most of your time Deal with distractions, Procrastination and Avoiding distractions		
***Self-Learning Components:** https://onlinecourses.nptel.ac.in/noc21_hs02/preview **Please Note:** 1)Students are supposed to learn the components on self-basis 2) At least 5-10 % syllabus will be asked in end term exams from self-learning components		
Reference Books: Aggarwal, R. S. (2014). Quantitative aptitude (Revised edition). Gladwell, M. (2021). Talking to strangers. Scott, S. (2004). Fierce conversations.		

Define Course Outcomes (CO)

COs	Statements
CO1	**Perform** calculations related to number systems, percentages and averages, quickly and accurately.
CO2	**Exhibit** confidence in tackling multiple-choice questions, time-constrained tests and competitive examinations.
CO3	**Demonstrate** active listening techniques, including attentive listening and reflection
CO4	**Articulate** and speak with confidence and express ideas clearly and coherently.
CO5	**Improve** confidence and display open and positive non-verbal communication.

COs Mapping with Levels of Bloom's taxonomy

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

CO3	C2	-	P4
CO4	C6	-	-
CO5	C6	-	P5

CO-PO Mapping

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

Justification for mapping must be relevant.

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

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

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

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

Department:	**Department of Computer Applications**		
Course Name: Summer Summer Internship / Project-I	**Course Code**	**L-T-P**	**Credits**
	SIBC251	0-0-0	2
Type of Course:	Summer Internship		
Pre-requisite(s), if any: NA			

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

The following options can be opted by the students:

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

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

Report Submission and Evaluation Guidelines:

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

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

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

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

Course Outcomes:

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

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

COURSE TEMPLATE

<table>
<tr><td><b>Department:</b></td><td colspan="3"><b>Department of Computer Applications</b></td></tr>
<tr><td rowspan="2"><b>Course Name:</b>
Database Management System</td><td><b>Course Code</b></td><td><b>L-T-P</b></td><td><b>Credits</b></td></tr>
<tr><td>ENBC204</td><td>3-1-0</td><td>4</td></tr>
<tr><td><b>Type of Course:</b></td><td colspan="3">Major</td></tr>
<tr><td colspan="4"><b>Pre-requisite(s), if any:</b></td></tr>
<tr><td colspan="4"><b>Brief Syllabus:</b>
This course introduces the basic concept of database, Database modelling languages, E-R modelling and Transaction Processing. Draw ER diagram to design a</td></tr>
<tr><td colspan="4"><b>UNIT WISE DETAILS</b></td></tr>
<tr><td><b>Unit Number: 1</b></td><td colspan="2"><b>Title:</b> Introduction to database</td><td><b>No. of hours: 8</b></td></tr>
<tr><td colspan="4"><b>Content Summary:</b>
Overview of DBMS, DBMS system vs file system, Data independence and abstraction level, Architecture of DBMS, Schemas, Instances and various DBMS models.</td></tr>
<tr><td><b>Unit Number: 2</b></td><td colspan="2">Title: Relational Query Languages</td><td><b>No. of hours: 10</b></td></tr>
<tr><td colspan="4">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

Mini project: Draw ER diagram to design a database to manage university course registration, including student records, courses, instructors, prerequisites, and enrolment.</td></tr>
<tr><td>Unit Number: 3</td><td colspan="2">Title: Database design and Transaction Processing</td><td>No. of hours: 8</td></tr>
</table>

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. Mini project: Design a database to manage a library's catalog, including books, authors, genres, and borrower information. Normalize the database to eliminate data duplication and maintain consistency.		
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 Mini project : Create Client_master with the following fields(ClientNO, Name, Address, City, State, bal_due) (a) Insert five records (b) Find the names of clients whose bal_due> 5000 . (c) Change the bal_due of ClientNO " C123" to Rs. 5100 (d) Change the name of Client_master to Client12. (e) Display the bal_due heading as "BALANCE"		
***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. 3. https://github.com/ossu/computer-science#databases.(OSSU computer science curriculum)		

Define Course Outcomes (CO)

COs	Statements
CO1	Analyze the key components and concepts of DBMS, including data independence, architecture, schemas and various DBMS models.

CO2	Apply data modeling techniques using ER model and understanding the concepts of keys
CO3	Evaluate the principles and techniques of relational modeling and the fundamental operations of relational algebra.
CO4	Design and implement effective database designs by analyzing functional dependencies and normalization.
CO5	Explain transaction processing, concurrency control and database recovery protocols in databases.

COs Mapping with Levels of Bloom's taxonomy

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

***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	2	3	2	1	2	2	1
CO2	2	2	2	2	2	1	1	2	2	1

CO3	2	2	2	2	2	1	1	2	2	1
CO4	2	2	3	2	2	1	1	2	2	1
CO5	2	2	2	2	3	1	1	2	2	1

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	Introduction to Database
Local	The syllabus on DBMS is relevant locally as it provides foundational knowledge and skills in managing databases, which are essential for businesses, organizations, and institutions at the local level.
Regional	The syllabus is also relevant regionally as the principles and concepts of DBMS are applicable and implemented in various industries and sectors within the region, contributing to the efficient management and utilization of data.
National	The syllabus holds national relevance as DBMS is a fundamental aspect of information management and technology infrastructure across the country. It helps in maintaining data integrity, supporting decision-making processes, and ensuring efficient data storage and retrieval.
Global	In today's interconnected world, the knowledge and skills related to DBMS have global significance. The concepts and principles covered in the syllabus align with international standards and practices in managing databases, making it relevant in a global context.

Employability	The syllabus on DBMS enhances employability as it equips students with the necessary knowledge and skills sought after by employers. Proficiency in DBMS is in demand across various industries, including IT, finance, healthcare, e-commerce, and more.
Entrepreneurship	Understanding DBMS is valuable for aspiring entrepreneurs. It enables them to design and implement effective database systems for their ventures, ensuring efficient data management and supporting business operations.
Skill Development	The syllabus contributes to skill development by providing theoretical knowledge and practical skills in DBMS. Students develop skills in data modeling, database design, SQL programming, and transaction management, which are transferable to various domains.
Professional Ethics	The syllabus indirectly addresses professional ethics by emphasizing data integrity, security, and privacy aspects of DBMS. Students learn about ethical considerations in handling sensitive data and ensuring responsible use of database systems.
Gender	
Human Values	The syllabus indirectly promotes human values by fostering responsible and ethical use of data. It encourages students to consider the impact of their actions on individuals, society, and the broader human community.
Environment & Sustainability	-
Unit II	**Data Modelling and Languages**
Local	The syllabus on Data Modeling and Relational Modeling is relevant locally as it provides foundational knowledge and skills in database design and management, which are essential for businesses, organizations, and institutions at the local level
Regional	The syllabus is also relevant regionally as database design and management principles are applicable and implemented in various industries and sectors within the region, contributing to efficient data organization and retrieval.
National	The syllabus holds national relevance as database design and management are fundamental aspects of

	information management and technology infrastructure across the country. It supports data integrity, effective decision-making, and efficient data storage and retrieval at a national level.
Global	In today's interconnected world, the knowledge and skills related to data modeling and relational modeling have global significance. The concepts and techniques covered in the syllabus align with international standards and practices in database design and management, making it relevant in a global context.
Employability	The syllabus on Data Modeling and Relational Modeling enhances employability as it equips students with the necessary knowledge and skills sought after by employers. Proficiency in data modeling, ER diagrams, relational algebra, and database design is in high demand across various industries globally.
Entrepreneurship	Understanding data modeling and relational modeling is valuable for aspiring entrepreneurs. It enables them to design and implement effective database systems for their ventures, ensuring efficient data management and supporting business operations.
Skill Development	The syllabus contributes to skill development by providing theoretical knowledge and practical skills in data modeling, ER diagrams, relational algebra, and database design. Students develop skills in conceptualizing data structures, mapping relationships, and performing database operations, which are transferable and valuable in various domains.
Professional Ethics	The syllabus indirectly addresses professional ethics by emphasizing data integrity, privacy, and responsible data management practices. Students learn about ethical considerations in designing databases and handling sensitive information.
Gender	
Human Values	The syllabus indirectly promotes human values by fostering responsible data management practices. It encourages students to consider the impact of their data modeling decisions on individuals, society, and ethical considerations.
Environment & Sustainability	

Unit III	**Database design and Transaction Processing**
Local	The syllabus on Database Design and Transaction Management is relevant locally as it provides foundational knowledge and skills in designing efficient and reliable databases, which are essential for businesses, organizations, and institutions at the local level.
Regional	The syllabus is also relevant regionally as database design and transaction management principles are applicable and implemented in various industries and sectors within the region, contributing to effective data management and transaction processing.
National	The syllabus holds national relevance as efficient database design and transaction management are crucial for information management and technology infrastructure across the country. It supports data integrity, data consistency, and reliable transaction processing at a national level.
Global	In today's interconnected world, the knowledge and skills related to database design and transaction management have global significance. The concepts and techniques covered in the syllabus align with international standards and practices, making it relevant in a global context.
Employability	The syllabus on Database Design and Transaction Management enhances employability as it equips students with the necessary knowledge and skills sought after by employers. Proficiency in functional dependencies, normalization, ACID properties, and transaction management is in high demand across various industries globally.
Entrepreneurship	Understanding database design and transaction management is valuable for aspiring entrepreneurs. It enables them to design and implement efficient and scalable database systems for their ventures, ensuring reliable data management and transaction processing.
Skill Development	The syllabus contributes to skill development by providing theoretical knowledge and practical skills in database design, functional dependencies, normalization, and transaction management. Students develop skills in identifying functional dependencies,

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

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

	principles when managing and utilizing databases for business purposes.

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: Introduction to Computer Network	**Course Code**:ENBC206	**L-T-P**	**Credits**
		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, Access networks, physical media, Forwarding, routing; packet switching; circuit switching; a network of network, packet delay and loss, end-end throughput.			
Unit Number: 2	**Title: Data Link Layer Design Issues**	**No. of hours: 12**	
Content Summary: Data Link Layer and Medium Access Sub Layer: Error Detection and Error Correction - Fundamentals, Block coding, Hamming Distance, CRC; Flow Control and Error control protocols - Stop and Wait, Go back – N ARQ, Selective Repeat ARQ, Sliding Window, Piggybacking, Random Access, Multiple access protocols -Pure ALOHA, Slotted ALOHA, CSMA/CD,CDMA/CA.			
Unit Number: 3	**Title: Introduction to Network Layer and Transport Services**	**No. of hours: 12**	

Content Summary: Network Layer: Switching, Logical addressing – IPV4, IPV6; Address mapping – ARP, RARP, BOOTP and DHCP–Delivery, Forwarding and Unicast Routing protocols. Transport Layer: Process to Process Communication, User Datagram Protocol (UDP), Transmission Control Protocol (TCP), SCTP Congestion Control; Quality of Service, QoS improving techniques: Leaky Bucket and Token Bucket algorithm.		
Unit Number: 4	**Title: Principles of Network Applications**	**No. of hours: 12**
Content Summary: Application Layer: Domain Name Space (DNS), DDNS, TELNET, EMAIL, File Transfer Protocol (FTP), WWW, HTTP, SNMP, Bluetooth, Firewalls, Basic concepts of Cryptography.		
***Self-Learning Components:** https://gaia.cs.umass.edu/kurose_ross/videos/1/ Cisco Networking Academy: network fundamentals, routing and switching, and network security. They provide free learning materials and hands-on practice: https://www.netacad.com/ Open-Source Networking Tools and Technologies • Open-source network monitoring tools (e.g., Nagios, Zabbix) • Open-source network management tools (e.g., OpenNMS) • Open-source network security tools (e.g., Snort, Suricata)		
Text Book: 1. Computer Networks (Fifth Edition) – Andrew S. Tanenbaum (Prentice Hall of India) 2. Data communication and Networking(Fourth Edition)- Behrouz A Forouzan(Tata Mcgraw Hill) **Reference Books:** 3. Computer Networking A Top-Down Approach(Fifth Edition)-James F. Kurose-Keith W. Ross (Pearson) 4. Computer Networks – Protocols, Standards and Interfaces (Second Edition) – UylessBlack(Prentice Hall of India Pvt. Ltd.)		

Define Course Outcomes (CO)

COs	Statements
CO1	Understand the fundamental concepts and principles of computer networks.
CO2	Demonstrate knowledge of network hardware and software components.
CO3	Develop skills in network administration and management.

CO4	Choose appropriate protocol for desired communication service.

COs Mapping with Levels of Bloom’s taxonomy

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

CO-PO Mapping

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

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

CO-PSO Mapping

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

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	

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: Fundamentals of Algorithm Design and 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: 8

Content Summary:

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

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

Content Summary:

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

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

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

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

Content Summary:
Computability of Algorithms, Computability classes – P, NP, NP complete and NP-hard. Cook's theorem, Standard NP-complete problems and Reduction techniques. String matching.

Self-Learning Components
Container loading problem, stable marriage problem, Coin Change problem

Reference Books

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

Define Course Outcomes (CO)

COs	Statements
CO1	Understand fundamental algorithmic concepts and how to analyze Complexities.
CO2	Analyze and evaluate algorithm performance.
CO3	Apply algorithmic problem-solving strategies.
CO4	Develop algorithm implementation skills.

COs Mapping with Levels of Bloom's taxonomy

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

CO1	C2	A3	P2
C02	C4	A4	P3
CO3	C3	A4	P4
CO4	C4	A5	P5

CO-PO Mapping

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

Justification for mapping must be relevant

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

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

National	-
Global	Addresses global understanding of the problems and how to find its solutions
Employability	After having knowledge about how to solve real world problems, new problems can be addressed to develop their algorithms.
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills to develop analytical skills
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Fundamental Algorithmic Strategies**
Local	-
Regional	-
National	-
Global	Employability: Proficiency in algorithm design techniques enhances employability opportunities globally.
Employability	Employability: Proficiency in algorithm design techniques enhances employability opportunities globally.
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills to develop analytical skills
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Graph and Tree Algorithms**
Local	-
Regional	-
National	-
Global	Addresses global understanding of the problems and how to find its solutions

Employability	After having knowledge about how to solve real world problems, new problems can be addressed to develop their algorithms.
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills to develop analytical skills
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Tractable and Intractable Problems**
Local	-
Regional	-
National	-
Global	Addresses global understanding of the problems and how to find its solutions
Employability	After having knowledge about how to solve real world problems, new problems can be addressed to develop their algorithms.
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills to develop analytical skills
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4, 8, 9
NEP 2020	Integration of Critical Thinking, Computational Thinking and Skill Development
POE/4th IR	Automation and Efficiency and Data Analysis.

Department:	**Department of Computer Applications**		
Course Name: Fundamentals of Algorithm Design and Analysis Lab	**Course Code**	**L-T-P**	**Credits**
	ENBC254	0-0-2	1
Type of Course:	Major		

Proposed Lab Experiments

Defined Course Outcomes

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

Ex. No	Experiment Title	Mapped CO/COs
1	Sort a given set of elements using the Quicksort method and determine the time required to sort the elements. Repeat the experiment for different values of n, the number of elements in the list to be sorted and plot a graph of the time taken versus n. The elements can be read from a file or can be generated using the random number generator	CO1
2	Design an algorithm to find the maximum and minimum elements in an unsorted array.	CO1
3	Implement Largest Common Subsequence.	CO1
4	Find Minimum Cost Spanning Tree of a given undirected graph using Kruskal's algorithm.	CO1
5	Find Minimum Cost Spanning Tree of a given undirected graph using Prim’s algorithm.	CO2
6	To Implement Optimal Binary Search Tree.	CO2
7	To Implement Strassen’s matrix multiplication Algorithm	CO2

8	Design an algorithm to find the maximum subarray sum in an array.	CO2
9	From a given vertex in a weighted connected graph, find shortest paths to other vertices using Dijkstra's algorithm.	CO2
10	Implement 0/1 Knapsack Problem using Dynamic algorithm concepts.	CO2
11	To implement Bellman Ford’s Algorithm.	CO2
12	To implement Depth First Search and Breadth First Search Algorithm.	CO2
13	To implement Naïve String-matching Algorithm.	CO3
14	Implement N Queen's problem using Back Tracking.	CO3
15	Design an algorithm to check if a given graph is acyclic (a DAG).	CO3
16	Obtain the Topological ordering of vertices in a given digraph.	CO3
17	Compute the transitive closure of a given directed graph using Warshall's algorithm	CO3
18	Design an algorithm to find the nth Fibonacci number using dynamic programming.	CO3
19	Design an algorithm to solve the 3-SAT problem using a backtracking approach.	CO4
20	Implement the brute-force algorithm to solve the Subset Sum Problem.	CO4
21	Design an algorithm to solve the Independent Set Problem using the branch and bound approach.	CO4
22	Design an algorithm to solve the Vertex Cover Problem using the 2-approximation algorithm.	CO4

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

Proposed Lab Experiments

Defined Course Outcomes

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

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:	CO3, CO4

	Find movies made after 1997 Find movies made by Hanson after 1997 Find all movies and their ratings Find all actors and directors Find Coen's movies with McDormand	
4	Database Schema for a customer-sale scenario Customer(Cust id : integer, cust_name: string) Item(item_id: integer, item_name: string, price: integer) Sale(bill_no: integer, bill_data: date, cust_id: integer, item_id: integer, qty_sold: integer) For the above schema, perform the following— Create the tables with the appropriate integrity constraints. Insert around 10 records in each of the tables. List all the bills for the current date with the customer names and item numbers. List the total Bill details with the quantity sold, price of the item and the final amount. List the details of the customer who have bought a product which has a price>200. Give a count of how many products have been bought by each customer Give a list of products bought by a customer having cust_id as 5. List the item details which are sold as of today. Create a view which lists out the bill_no, bill_date, cust_id, item_id, price, qty_sold, amount. Create a view which lists the daily sales date wise for the last one week	CO3, CO4
5	Database Schema for a Student Library scenario Student(Stud_no : integer, Stud_name: string) Membership(Mem_no: integer, Stud_no: integer) Book(book_no: integer, book_name:string, author: string) Iss_rec(iss_no:integer, iss_date: date, Mem_no: integer, book_no: integer) For the above schema, perform the following— Create the tables with the appropriate integrity constraints Insert around 10 records in each of the tables List all the student names with their membership numbers List all the issues for the current date with student and Book names List the details of students who borrowed book whose author is CJDATE Give a count of how many books have been bought by each student Give a list of books taken by student with stud_no as 5 List the book details which are issued as of today	CO3, CO4

	Create a view which lists out the iss_no, iss _date, stud_name, book name Create a view which lists the daily issues-date wise for the last one week	
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— Create the tables with the appropriate integrity constraints Insert around 10 records in each of the tables List all the customer names with their membership numbers List all the issues for the current date with the customer names and cassette names List the details of the customer who has borrowed the cassette whose title is " The Legend" Give a count of how many cassettes have been borrowed by each customer Give a list of book which has been taken by the student with mem_no as 5 List the cassettes issues for today Create a view which lists outs the iss_no, iss_date, cust_name, cass_name 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) Allotment(Stud_no: Integer, mach_no: integer, dayof week: string) For the above schema, perform the following— Create the tables with the appropriate integrity constraints Insert around 10 records in each of the tables List all the machine allotments with the student names, lab and machine numbers. List the total number of lab allotments day wise Give a count of how many machines have been allocated to the 'CSIT' class Give a machine allotment details of the stud_no 5 with his personal and class details Count for how many machines have been allocated in Lab_no 1 for the day of the week as "Monday"	CO3, CO4

<table>
<tr>
<td></td>
<td>How many students class wise have allocated machines in the labs
Create a view which lists out the stud_no, stud_name, mach_no, lab_no, dayofweek
Create a view which lists the machine allotment details for "Thursday".</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Consider the following table:
Table: CLASS
<table>
<tr><th>Id</th><th>Name</th></tr>
<tr><td>1</td><td>Bravo</td></tr>
<tr><td>2</td><td>Alex</td></tr>
<tr><td>4</td><td>Cheng</td></tr>
</table>
Give the output of the following SQL script:
> INSERT INTO class VALUES (5,'Rahul');
> COMMIT;
> UPDATE class SET name = 'Abhijeet' WHERE id= '5';
> SAVEPOINT A;
> INSERT INTO class VALUES (6, 'Chris');
> SAVEPOINT B;
> INSERT INTO class VALUES (7, 'Bravo');
> SAVEPOINT C
> SELECT * FROM class;
> ROLLBACK TO B;
> SELECT * FROM class;
> ROLLBACK TO A;</td>
<td>CO3, CO4</td>
</tr>
<tr>
<td>10</td>
<td>Consider the following two tables: SHOP and ACCESSORIES
Table: SHOP
<table>
<tr><th>ID</th><th>ShopName</th><th>Area</th></tr>
<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>
</table>
Table: ACCESSORIES
<table>
<tr><th>No</th><th>Name</th><th>Price</th><th>Id</th></tr>
<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>
</table></td>
<td>CO3, CO4</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>
<table>
<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>
</table>
Perform Cartesian product or Cross join of both tables.
To display the Name and Price of all the Accessories in ascending order of their price.
To display ID and ShopName of all shops located in Nehru Place.
To display minimum and maximum price of all accessories.
To display Name, Price of all accessories and their respective ShopName where they are available.
</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>In continuation with experiment no. 10, find the output of the following SQL queries based on above mentioned tables:
SELECT DISCTINCT NAME FROM ACCESSORIES WHERE PRICE >= 5000;
SELECT AREA, COUNT(*) FROM SHOP GROUP BY AREA;
SELECT COUNT(DISTINCT AREA) FROM SHOP;
SELECT NAME, PRICE*0.05 DISCOUNT FROM ACCESSORIES WHERE ID IN (‘S02’, ‘S03’);</td>
<td>CO3, CO4</td>
</tr>
<tr>
<td>12</td>
<td>Consider the following two tables: PRODUCT and CLIENT.
Table: Product
<table>
<tr><th>P_ID</th><th>ProdName</th><th>Manufacturer</th><th>Price</th><th>ExpiryDate</th></tr>
<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>
</table>
Note:
P_ID is the primary key.
Table: Client
<table>
<tr><th>C_ID</th><th>ClientName</th><th>City</th><th>P_ID</th></tr>
<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>
<tr><td>12</td><td>Live Life</td><td>Delhi</td><td>SH06</td></tr>
<tr><td>15</td><td>Pretty One</td><td>Delhi</td><td>FW05</td></tr>
<tr><td>16</td><td>Dreams</td><td>Bengaluru</td><td>TP01</td></tr>
<tr><td>14</td><td>Expressions</td><td>Delhi</td><td>NULL</td></tr>
</table>
Note:
C_ID is the primary key.
P_ID is the foreign key referencing P_ID of Client Table.

To display the ClientName and City of all Mumbai and Delhi based clients in Client table.</td>
<td>CO3, CO4</td>
</tr>
</table>

	Increase the price of all the products in Product Table by 10%. To display the ProdName, Manufacturer, ExpiryDate of all the products that expired on or before '2010-12-31'. To display C_ID, ClientName, City of all the clients including the ones that have not purchased a product and their corresponding ProdName sold. Display the distinct Manufacturer from Product table. Display the ClientName, C_ID who belong to a city starts with 'M'	
13	Consider the following schema for a Library Database: BOOK(Book_id, Title, Publisher_Name, Pub_Year) BOOK_AUTHORS(Book_id, Author_Name) PUBLISHER(Name, Address, Phone) BOOK_COPIES(Book_id, Programme_id, No-of_Copies) BOOK_LENDING(Book_id, Programme_id, Card_No, Date_Out, Due_Date) LIBRARY_PROGRAMME(Programme_id, Programme_Name, Address) Write SQL queries to 1. Retrieve details of all books in the library – id, title, name of publisher, authors, number of copies in each Programme, etc. 2. Get the particulars of borrowers who have borrowed more than 3 books, but from Jan 2017 to Jun 2017. 3. Delete a book in BOOK table. Update the contents of other tables to reflect this data manipulation operation. 4. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query. 5. Create a view of all books and its number of copies that are currently available in the Library	CO3, CO4
14	Consider the following schema for Order Database: SALESMAN(Salesman_id, Name, City, Commission) CUSTOMER(Customer_id, Cust_Name, City, Grade, Salesman_id) ORDERS(Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id) Write SQL queries to 1. Count the customers with grades above Bangalore's average. 2. Find the name and numbers of all salesman who had more than one customer. 3. List all the salesman and indicate those who have and do not have customers in their cities (Use UNION operation.) 4. Create a view that finds the salesman who has the customer with the highest order of a day. 5. Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.	CO3, CO4
15	Consider the schema for Movie Database: ACTOR(Act_id, Act_Name, Act_Gender) DIRECTOR(Dir_id, Dir_Name, Dir_Phone) MOVIES(Mov_id, Mov_Title, Mov_Year, Mov_Lang,	CO3, CO4

	Dir_id) MOVIE_CAST(Act_id, Mov_id, Role) RATING(Mov_id, Rev_Stars) Write SQL queries to 1. List the titles of all movies directed by 'Hitchcock'. 2. Find the movie names where one or more actors acted in two or more movies. 3. List all actors who acted in a movie before 2000 and in a movie after 2015 (use JOIN operation). 4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title. 5. Update rating of all movies directed by 'Steven Spielberg' to 5.	
16	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) Write SQL queries to 1. List all the student details studying in fourth semester 'C' section. 2. Compute the total number of male and female students in each semester and in each section. 3. Create a view of Test1 marks of student USN '1BI15CS101' in all Courses. 4. Calculate the FinalIA (average of best two test marks) and update the corresponding table for all students. 5. Categorize students based on the following criterion: If FinalIA = 17 to 20 then CAT = 'Outstanding' If FinalIA = 12 to 16 then CAT = 'Average' If FinalIA< 12 then CAT = 'Weak' Give these details only for 8th semester A, B, and C section students.	CO3, CO4
17	Consider the schema for Company Database: EMPLOYEE(SSN, Name, Address, Sex, Salary, SuperSSN, DNo) DEPARTMENT(DNo, DName, MgrSSN, MgrStartDate) DLOCATION(DNo,DLoc) PROJECT(PNo, PName, PLocation, DNo) WORKS_ON(SSN, PNo, Hours) Write SQL queries to 1. Make a list of all project numbers for projects that involve an employee whose last name is 'Scott', either as a worker or as a manager of the department that controls the project. 2. Show the resulting salaries if every employee working on the 'IoT' project is given a 10 percent raise. 3. Find the sum of the salaries of all employees of the 'Accounts' department, as well as the maximum salary, the minimum salary, and the average salary in this department.	CO3, CO4

	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.	

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

Proposed Lab Experiments

Defined Course Outcomes

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

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.	CO1
5	Enable inter-VLAN routing using a router or Layer 3 switch.	CO2
6	Test connectivity between PCs in different VLANs.	CO3
7	Set up a network with multiple routers.	CO4
8	Configure static routes on routers to enable communication between networks.	CO2
9	Verify routing tables and test connectivity between networks.	CO1
10	Set up a network with a private IP address space.	CO2
11	Configure NAT on a router to enable translation between private and public IP addresses.	CO3

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

Department:	**Department of Computer Applications**		
Course Name: **Life Skills for Professionals - II**	**Course Code** : AEC012	**L-T-P**	**Credits**
		3-0-0	3
Type of Course:	AEC		
Pre-requisite(s), if any:			

Brief Syllabus:
This course is a multifaceted initiative designed to enhance and optimize learner’s communication practices across various platforms. This program integrates a range of strategies, tools, and techniques to foster effective communication, facilitate collaboration, and promote a cohesive information flow within the learner’s area. This course is structured and comprehensive initiative designed to develop and improve individuals' aptitude across various cognitive and behavioral domains. This course incorporates a range of assessments, training modules, and activities to enhance critical thinking, problem-solving, decision-making, and other essential aptitudes required for personal and professional success.

UNIT WISE DETAILS

Unit Number: 1	**Title: Personality Improvement**	**No. of hours: 4**
Content Summary: Asking for and giving information, Offering and responding to offers, Requesting and responding to requests, Congratulating people on their success, Asking questions and responding politely, Apologizing and forgiving		
Unit Number: 2	**Title: Ratio & its application**	**No. of hours: 8**
Content Summary: Time & Work, Time & Distance, Train, Boat & Stream, Permutation & combination, Probability		
Unit Number: 3	**Title: Arithmetic**	**No. of hours: 8**
Content Summary: Inequalities, Log, progression, Mensuration, BODMAS		
Unit Number: 4	**Title: Presentation Skills**	**No. of hours: 8**

Content Summary: Presentation Skills, Telephone etiquettes, LinkedIn Profile and professional networking, Video resumes & Mock interview sessions.		
Unit Number: **5**	**Title:** **Leadership skills**	**No. of hours: 4**
Content Summary: Nurturing future leaders, Increasing productivity of the workforce, Imparting Self-leadership, Executive leadership		
***Self-Learning Components:** **Please Note:** **1) Students are supposed to learn the Interactive Learning Modules on the internet.** **2) Webinars and Podcasts/ Self-Assessment Tools/Case Studies and Projects** **3) At least 5-10 % syllabus will be asked in end term exams from self-learning components**		
Reference Books: Aggarwal, R. S. (2014). Quantitative aptitude (Revised edition). Gladwell, M. (2021). Talking to strangers. Scott, S. (2004). Fierce conversations.		

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** and apply the fundamental theories, models, and principles of communication.
CO2	**Apply** ability to communicate effectively through spoken and written forms. It includes developing skills in public speaking, interpersonal communication, professional writing, and persuasive communication.
CO3	**Evaluate** the development of teamwork and collaboration skills. It includes activities such as group projects, team-building exercises, and simulations that allow students to practice effective communication and collaboration within diverse teams
CO4	**Improve** their communication skills in different professional and personal contexts, such as interviews, networking events, customer interactions, and interpersonal relationships
CO5	**Analyze** ideas and information clearly and concisely through spoken language. They will develop the ability to articulate their thoughts, use appropriate vocabulary, and convey their message with clarity.

COs Mapping with Levels of Bloom’s taxonomy

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

***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

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

Relevance of the Syllabus to various indicators

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

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

Regional	-
National	Contributes to develop skill and improved productivity
Global	Aligns with global trends in understanding the deadlines.
Employability	Enhance the employability of individuals by developing essential skills and competencies sought by employers
Entrepreneurship	-
Skill Development	Strengthening critical thinking, problem-solving, memory, and other cognitive functions to improve overall mental agility and performance.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	Aligns with the concepts employability

MINOR PROJECT-I

Department:	**Department of Computer Applications**		
Course Name: Minor Project	**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 .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)

COURSE TEMPLATE

<table>
<tr><td><b>Department:</b></td><td colspan="4"><b>Department of Computer Applications</b></td></tr>
<tr><td colspan="2" rowspan="2"><b>Course Name:</b>
<b>Competitive Programming Lab</b></td><td><b>Course Code</b></td><td><b>L-T-P</b></td><td><b>Credits</b></td></tr>
<tr><td>SEC036</td><td>0-0-4</td><td>2</td></tr>
<tr><td><b>Type of Course:</b></td><td colspan="4">Skill Enhancement Course (SEC)</td></tr>
<tr><td colspan="5"><b>Pre-requisite(s), if any: None</b></td></tr>
<tr><td colspan="5"><b>Brief Syllabus:</b>
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</td></tr>
</table>

Table of Contents

S.N	Experiment Index	COs
1	Introduction to Competitive Coding • Overview of competitive coding and its importance in the field of computer science. • Understanding the significance of problem-solving skills and algorithmic thinking in competitive coding.	CO1
2	Data Structures and Algorithms • Review of fundamental data structures: arrays, linked lists, stacks, queues, trees, graphs, and hash tables. • Study of essential algorithms: searching, sorting, recursion, dynamic programming, greedy algorithms, and graph algorithms.	CO1
3	Time and Space Complexity Analysis • Understanding time and space complexity of algorithms. • Analysis of algorithm efficiency and choosing the most optimal solutions.	CO2
4	Problem Solving Techniques • Introduction to problem-solving techniques like brute force, divide and conquer, backtracking, and more. • Practice in applying different techniques to solve a variety of programming problems.	CO3
5	Advanced Data Structures • Study of advanced data structures: heaps, priority queues, segment trees, trie, and advanced graph structures. • Understanding the use of these data structures in solving complex programming problems.	CO4
6	Coding Paradigms • Introduction to different coding paradigms: procedural programming, object-oriented programming, and functional programming.	CO5

	• Understanding the benefits and drawbacks of each paradigm in competitive coding.	
7	Online Judges and Contest Platforms • Familiarization with popular online judge platforms like Codeforces, Topcoder, and LeetCode. • Practice solving problems from online contests and participating in coding competitions. List of suggested links to coding platforms ▪ Codeforces: https://codeforces.com/ ▪ Topcoder: https://www.topcoder.com/ ▪ AtCoder: https://atcoder.jp/ ▪ LeetCode: https://leetcode.com/ ▪ HackerRank: https://www.hackerrank.com/ ▪ CodeChef: https://www.codechef.com/ ▪ HackerEarth: https://www.hackerearth.com/ ▪ Project Euler: https://projecteuler.net/ ▪ UVa Online Judge: https://onlinejudge.org/ ▪ SPOJ (Sphere Online Judge): https://www.spoj.com/ ▪ Google Code Jam: https://codingcompetitions.withgoogle.com/codejam ▪ Kick Start by Google: https://codingcompetitions.withgoogle.com/kickstart ▪ ACM ICPC Live Archive: https://icpcarchive.ecs.baylor.edu/ ▪ A2 Online Judge: https://a2oj.com/ ▪ CodeSignal: https://codesignal.com/	CO5
8	Tips and Tricks for Competitive Coding • Learning effective coding techniques, shortcut methods, and best practices for competitive coding. • Developing strategies to optimize code, manage time, and improve problem-solving speed.	CO5
9	Mock Contests and Practice Sessions • Conducting mock contests and practice sessions to simulate real coding competitions. • Solving a wide range of problems to enhance coding skills and adaptability to different problem types.	CO5

10	**Self-Learning Component:** List of Suggested Competitive programming Courses: ▪ Competitive Programmer's Core Skills" by Coursera: This course covers fundamental algorithms and data structures used in competitive programming. Link: https://www.coursera.org/learn/competitive-programming-core-skills ▪ "Algorithms and Data Structures" by MIT OpenCourseWare: This course teaches essential algorithms and data structures for competitive programming. Link: https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-006-introduction-to-algorithms-fall-2011/ ▪ "Data Structures and Algorithms" by GeeksforGeeks: This course covers various data structures and algorithms commonly used in competitive programming. Link: https://practice.geeksforgeeks.org/courses/dsa-self-paced ▪ "Introduction to Competitive Programming" by NPTEL: This course introduces the basics of competitive programming and covers algorithms and problem-solving techniques. Link: https://onlinecourses.nptel.ac.in/noc21_cs07/ ▪ "Competitive Programming" by HackerRank: This course provides in-depth coverage of algorithms and data structures with hands-on coding exercises. Link: 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:	CO5

	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*	

Suggested Books

1. "Competitive Programming 3" by Steven Halim and Felix Halim: This book is a comprehensive guide to competitive programming, covering algorithms, data structures, problem-solving techniques, and contest strategies. It includes numerous examples, explanations, and practice problems. Book Link
2. "Algorithms" by Robert Sedgewick and Kevin Wayne: This book provides a thorough introduction to algorithms, including sorting, searching, graph algorithms, and dynamic programming. It includes detailed explanations, visualizations, and implementation examples. Book Link
3. "Introduction to Algorithms" by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein: Known as "CLRS," this book is a classic reference for algorithms. It covers a wide range of algorithms, data structures, and algorithm design techniques. Book Link
4. "Programming Challenges" by Steven S. Skiena and Miguel A. Revilla: This book presents a collection of programming problems from various competitions

and online judges. It provides problem-solving techniques, algorithmic approaches, and example solutions. Book Link

5. "The Art of Computer Programming" by Donald E. Knuth: This multi-volume series is considered a classic in computer science. It covers various algorithms, data structures, and mathematical techniques in great detail. Book Link
6. "Cracking the Coding Interview" by Gayle Laakmann McDowell: Although not specifically focused on competitive programming, this book is a popular resource for coding interview preparation. It covers essential data structures, algorithms, and problem-solving techniques. Book Link
7. "Programming Pearls" by Jon Bentley: This book presents a collection of programming challenges and discusses techniques for solving them efficiently. It emphasizes problem-solving skills and algorithmic thinking. Book Link

Web References

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

Course Outcomes

CO1	Proficiency in Algorithms and Data Structures: Demonstrate proficiency in implementing and analyzing various algorithms and data structures commonly used in competitive programming.
CO2	Efficient Problem Solving: Develop the ability to analyze problem statements, design efficient algorithms, and write optimized code to solve competitive programming problems within time and memory constraints.

CO3	Algorithmic Thinking: Cultivate algorithmic thinking and problem-solving skills by identifying patterns, applying appropriate algorithms, and selecting optimal data structures for a given problem.
CO4	Code Optimization and Complexity Analysis: Apply strategies to optimize code and improve time and space complexity of solutions, considering factors such as algorithm selection, data structure usage, and efficient coding techniques.
CO5	Competitive Programming Skills: Gain familiarity with different online competitive programming platforms, participate in coding competitions, and develop strong problem-solving and critical thinking skills in a competitive programming environment.

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

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: **Computer Organization & Architecture**	**Course Code** :ENBC301	**L-T-P**	**Credits**
		3-1-0	4
Type of Course:	Major (Core)		
Pre-requisite(s), if any: Concepts of Digital Electronics			

Brief Syllabus:

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

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

Text Book:

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

Reference Books:

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

Online References:

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

Course Outcomes (CO)

COs	Statements
CO1	Understand the basics of instructions sets and their impact on processor design
CO2	Demonstrate an understanding of the design of the functional units of a digital computer system
CO3	Evaluate cost performance and design trade-offs in designing and constructing a computer processor including memory.
CO4	Design a pipeline for consistent execution of instructions with minimum hazards
CO5	Manipulate representations of numbers stored in digital computers using I/O devices and store them into memory

COs Mapping with Levels of Bloom's taxonomy

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

CO-PO Mapping

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

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	**Introduction**
Local	Data representation and basic operations: Local, as it focuses on specific techniques and algorithms used within a computer system. Notion of IPC:

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

Employability	
Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Memory hierarchy, Storage and I/O**
Local	-
Regional	-
National	-
Global	Introduction to magnetic disks, notion of tracks, sectors, flash memory: Global, as they are fundamental concepts and technologies applicable to computer storage systems worldwide.
Employability	-
Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4, SDG 8, SDG 9
NEP 2020	Promoting universal access to education, holistic development, multidisciplinary approach, skill development, critical thinking, creativity, ICT integration, research and development, global competencies, and professional ethics.
POE/4th IR	Aligns with the concepts of parallel computing, advanced processors, and memory architectures.

Department:	**Department of Computer Applications**		
Course Name: **NATURAL LANGUAGE PROCESSING**	**Course Code**	**L-T-P**	**Credits**
	ENSP302	4-0-0	4
Type of Course:	Minor (Department Elective II)		

Pre-requisite(s), if any: Strong programming skills, particularly in Python.

Brief Syllabus:
The ultimate objective of NLP is to read, decipher, understand, and make sense of the human languages in a manner that is valuable.
It helps resolve ambiguity in language and adds useful numeric structure to the data for many downstream applications, such as speech recognition or text analytics.

UNIT WISE DETAILS

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

Content Summary: Natural Language Processing in real world, What is language, Approached to NLP,

Build NLP model: Eights Steps for building NLP Model, Web Scrapping

Unit Number: 2	**Title: Text Representation**	**No. of hours: 10**

Content Summary: Basic Vectorization, One-Hot Encoding, Bag of Words, Bag of N Grams, TF-IDF, Pre-trained Word Embedding, Custom Word Embeddings, Vector Representations via averaging, Doc2Vec Model, Visualizing Embeddings using TSNW and Tensorbaord

<table>
<tr><td colspan="3"><b>Text Classification:</b> Application of Text Classification, Steps for building text classification system, Text classification using Naïve Bayes Classifier, Logistic Regression, and Support Vector Machine, Neural embedding for Text Classification, text classification using deep learning, interpret text classification model</td></tr>
<tr><td><b>Unit Number: 3</b></td><td><b>Title: Information Extraction</b></td><td><b>No. of hours: 10</b></td></tr>
<tr><td colspan="3"><b>Content Summary:</b> Applications of Information Extraction, Processes for Information Extraction. Key phrase Extraction, Named Entity Recognition, Disambiguation and linking of named entity, Relationship extraction

<b>Chatbot:</b> Real life applications of chatbot, Chatbot Taxonomy, Dialog Systems, Process of building a dialog, Components of Dialog System, End to End Approach, Rasa NLU</td></tr>
<tr><td><b>Unit Number: 4</b></td><td><b>Title: NLP for social media</b></td><td><b>No. of hours: 10</b></td></tr>
<tr><td colspan="3"><b>Content Summary:</b> Application of NLP in social media, challenges with social media, Natural Language Processing for Social Data, Understanding Twitter Sentiments, Identifying memes and Fake News

<b>NLP for E-Commerce</b>: E-commerce catalog, Search in E-Commerce, How to build an e-commerce catalog, Review and Sentiment Analysis, Recommendations for E-Commerce</td></tr>
<tr><td colspan="3"><b>*SELF-LEARNING COMPONENTS:</b>

https://onlinecourses.nptel.ac.in/noc23_cs45/preview

<b>Please Note:</b>

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

2) At least 5-10 % syllabus will be asked in end term exams from self-learning components.</td></tr>
<tr><td colspan="3"><b>Reference Books:</b> Natural Language Processing with Python by Steven Bird, Ewan Klein and Edward Loper

Foundations of Statistical Natural Language Processing by Christopher Manning and Hinrich Schütze</td></tr>
</table>

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** the fundamentals of Natural Language Processing (NLP).
CO2	**Analyze** and represent text data using various techniques.
CO3	**Implement** text classification and information extraction techniques
CO4	**Apply** NLP techniques to analyze social media data
CO5	**Develop** practical solutions using NLP for real-world problems

COs Mapping with Levels of Bloom's taxonomy

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

CO4	C6	-	P3
CO5	C4	-	P3

CO-PO Mapping

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

Justification for mapping must be relevant

1=weakly mapped

2= moderately mapped

3=strongly mapped

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	**Introduction to NLP**
Local	Understanding NLP in real-world scenarios can address local language processing needs and applications.
Regional	NLP techniques can be applied regionally to various language-related tasks and projects.
National	NLP is significant for various national-level language processing initiatives and applications.
Global	NLP has widespread global applications, addressing language challenges across different cultures and regions.
Employability	Proficiency in NLP opens up employment opportunities in fields requiring language processing, AI, and data analysis skills.
Entrepreneurship	NLP knowledge can inspire entrepreneurial ventures, such as developing innovative NLP models or language-based startups.
Skill Development	Learning NLP enhances analytical and programming skills, vital for developing language-based AI applications.
Professional Ethics	Applying NLP ethically is essential to ensure responsible use of language processing technologies and avoid biases.
Gender	NLP applications should be unbiased and inclusive, considering the diverse linguistic needs of individuals of all genders.
Human Values	Implementing NLP with human values fosters language understanding and communication to improve human interactions.
Environment & Sustainability	NLP can be applied to develop sustainable language technologies and promote multilingual environmental awareness.
Unit II	**Text Representation**
Local	Text representation techniques can be tailored to local language data, improving the effectiveness of NLP applications in local contexts.

Regional	Text representation methods can be applied to regional language data for diverse NLP applications in specific geographic regions.
National	Text representation is crucial for handling various languages used nationally, enhancing the versatility of NLP systems.
Global	Text representation techniques enable effective NLP across different languages and cultures, making it globally applicable.
Employability	Proficiency in text representation methods enhances employability in NLP-related fields, data science, and AI industries.
Entrepreneurship	Knowledge of text representation can inspire the development of language processing tools and solutions catering to specific markets.
Skill Development	Learning text representation methods develops expertise in feature engineering and model training for NLP tasks.
Professional Ethics	Applying text representation ethically is essential to maintain accurate and unbiased language processing systems.
Gender	Gender-aware text representation avoids gender biases and ensures inclusive language understanding in NLP applications.
Human Values	Text representation with ethical considerations promotes human values and fosters responsible use of language technologies.
Environment & Sustainability	Text representation can be applied to promote environmental awareness and communication in diverse languages.
Unit III	**Information Extraction**
Local	Information extraction techniques can be applied to extract valuable insights from local language data for regional applications.
Regional	Regional information extraction applications can address specific language-related challenges in a particular geographic area.

National	Information extraction is valuable for various national-level language processing projects and knowledge discovery applications.
Global	Global information extraction aids in understanding multilingual data, supporting cross-cultural analysis and applications.
Employability	Proficiency in information extraction techniques enhances employability in NLP, data analysis, and information retrieval roles.
Entrepreneurship	Knowledge of information extraction can inspire the development of specialized language processing tools for diverse industries.
Skill Development	Learning information extraction methods develops expertise in pattern recognition and knowledge extraction from unstructured data.
Professional Ethics	Applying information extraction ethically ensures responsible use of data and protects individuals' privacy in language processing.
Gender	Gender-aware information extraction promotes inclusive language analysis and avoids reinforcing gender biases in the output.
Human Values	Information extraction with ethical considerations respects individual values and fosters meaningful knowledge discovery.
Environment & Sustainability	Information extraction can support environmental research and awareness by analyzing language data related to sustainability.
Unit IV	**NLP for Social Media**
Local	NLP for social media can address local language usage in social platforms, supporting community engagement and language understanding.

Regional	Regional NLP applications in social media can capture language nuances specific to various regions and enhance social analytics.
National	NLP for social media is significant for national-level sentiment analysis, understanding public opinions, and monitoring social trends.
Global	Global NLP for social media enables monitoring sentiments across cultures, tracking global trends, and addressing language-based challenges.
Employability	Proficiency in NLP for social media enhances employability in digital marketing, social media analytics, and sentiment analysis roles.
Entrepreneurship	Knowledge of NLP for social media can inspire the development of social media monitoring tools, fake news detection systems, and analytics.
Skill Development	Learning NLP for social media develops expertise in sentiment analysis, language processing for online content, and social data analytics.
Professional Ethics	Applying NLP in social media ethically ensures responsible use of social data, user privacy protection, and unbiased sentiment analysis.
Gender	Gender-aware NLP in social media avoids reinforcing gender stereotypes and respects the diverse language use of individuals of all genders.
Human Values	NLP for social media with ethical considerations promotes constructive online conversations, combats misinformation, and encourages empathy.
Environment & Sustainability	NLP for social media can be applied to analyze social discussions related to environmental issues, promoting awareness and action.
SDG	SDG 4
NEP 2020	-

POE/4th IR	-

Department:	**Department of Computer Applications**		
Course Name:	**Course Code**	**L-T-P**	**Credits**
NATURAL LANGUAGE PROCESSING LAB	**ENSP352**	0-0-2	1
Type of Course:	Minor (Department Elective II)		
Pre-requisite(s), if any: Strong programming skills, particularly in Python.			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	**Understand** the fundamental concepts and techniques of web scraping to extract data from websites efficiently and ethically.
CO 2	**Acquire** proficiency in using developer tools to inspect and analyze website elements, facilitating data extraction and understanding the underlying structure of web pages.
CO 3	**Implement** mechanisms to request permission for web scraping, ensuring compliance with legal and ethical guidelines related to data access and usage.
CO 4	**Develop** skills in inspecting specific HTML elements, such as the H1 element and table element, for targeted data extraction and analysis.
CO 5	**Demonstrate** proficiency in data preprocessing and cleaning techniques, including column list creation and cleaning, to prepare raw data for natural language processing tasks.

List of Experiments

Ex. No.	Experiment Title	Mapped COs
1	Write a program to scrap a website	CO1, CO3
2	Write a program to inspect a website using dev tools	CO2
3	Write a program to request permission to scrap a website	CO3
4	Write a program to inspect H1 element of a website	CO2, CO4
5	Write a program to inspect table element of a website	CO2, CO4
6	Write a program to create a column list	CO5
7	Write a program to clean a column list	CO5
8	Write a program for word tokenization	CO5
9	Write a program to implement RegEx for word tokenization	CO5
10	Write a program to implement stopwords	CO5
11	Write a program to implement LSTM	CO5

Department:	**Department of Computer Applications**		
Course Name: **Life Skills for Professionals -III**	**Course Code** : AEC013	**L-T-P**	**Credits**
		2-1-0	3
Type of Course:	AEC		
Pre-requisite(s), if any:			
Brief Syllabus: This Course designed to enhance the employability of individuals by developing essential skills and competencies sought by employers. This program equips participants with a wide range of skills necessary for success in the modern job market. To engage in interactive workshops, practical exercises, role-playing, and real-world simulations to reinforce their learning. The course is designed to be inclusive and caters to individuals from diverse backgrounds and career aspirations. The course is designed to enhance and develop various cognitive skills and mental abilities. This course focuses on strengthening critical thinking, problem-solving, memory, and other cognitive functions to improve overall mental agility and performance.			
UNIT WISE DETAILS			
Unit Number: 1	**Title:** **Data interpretation**	**No. of hours: 4**	
Content Summary: Table chart, Line graph, Bar graph, Pie chart			
Unit Number: 2	**Title: Logical Reasoning**	**No. of hours: 8**	
Content Summary: Coding & Decoding, Sitting arrangement, Calendar, Clock, Direction Sense, Blood relation, Syllogism.			
Unit Number: 3	**Title:** **Logical & Non-verbal reasoning**	**No. of hours: 8**	
Content Summary: Series, Puzzle Text, Statement & Arguments, Cube & Dice, Non-verbal Reasoning			
Unit Number: 4	**Title:** **Understanding Stress**	**No. of hours: 8**	

Content Summary:
Introduction to Stress (i) Introduction to stress: Meaning, Definition, Eustress, Distress, (ii) Types of stress: Acute stress, Episodic Acute stress and chronic stress, signs and Symptoms
Sources of stress (i) Psychological, Social, Environmental (ii) Academic, Family and Work stress
Impact of stress

Unit Number: 5	**Title:** **Employability skills**	**No. of hours: 4**

Content Summary: Identifying job openings, Enhancing interpersonal skills, including teamwork, Applying for a job, Preparing Cover letters, preparing a CV/Resume and Effective Profiling, Group Discussions, Preparing for and Facing a Job Interview, Mock Interview, Feed Back – Improvement

***Self-Learning Components:**

Please Note:
1) Students are supposed to learn the Interactive Learning Modules on the internet.
2) Webinars and Podcasts/ Self-Assessment Tools/Case Studies and Projects
3) At least 5-10 % syllabus will be asked in end term exams from self-learning components

Reference Books:
Aggarwal, R. S. (2014). Quantitative aptitude (Revised edition).
Gladwell, M. (2021). Talking to strangers.
Scott, S. (2004). Fierce conversations.

Define Course Outcomes (CO)

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

CO5	**Create** leadership skills and to motivate and inspire others, manage projects effectively, and demonstrate a proactive and responsible approach to their spoken language.

COs Mapping with Levels of Bloom's taxonomy

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

***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

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

Relevance of the Syllabus to various indicators

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

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

Human Values	-
Environment & Sustainability	-
Unit IV	Understanding Stress
Local	Attentively listen to others, understand their perspectives, and respond appropriately
Regional	-
National	Contributes to develop skill and improved productivity
Global	Aligns with global trends in encouraged to establish professional connections and learn effective techniques for engaging in informational interviews or networking events
Employability	Enhance the employability of individuals by developing essential skills and competencies sought by employers
Entrepreneurship	-
Skill Development	Strengthening critical thinking, problem-solving, memory, and other cognitive functions to improve overall mental agility and performance.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit V	Employability skills
Local	Attentively listen to others, understand their perspectives, and respond appropriately with timelines
Regional	-
National	Contributes to develop skill and improved productivity
Global	Aligns with global trends in understanding importance of networking during the job search process
Employability	Enhance the employability of individuals by developing essential skills and competencies sought by employers
Entrepreneurship	-
Skill Development	Strengthening critical thinking, problem-solving, memory, and other cognitive functions to improve overall mental agility and performance.
Professional Ethics	-

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

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: Computational Services In The Cloud	**Course Code** :ENSP401	**L-T-P**	**Credits**
		4-0-0	4
Type of Course:	Departmental Elective I -Minor		
Pre-requisite(s), if any:			

Brief Syllabus:

This course covers evolutionary computing paradigms from multi-processor systems to Cloud, Edge, and Fog Computing. The course design follows cloud deployment models, service models, virtualization techniques and cloud architectural solutions. It also elaborates cloud compliances and security at fine-grained level by following a shared responsibility model. It reveals a design pattern to the students, enabling them to think through the process of designing and implementing cloud infrastructure and optimal IT solutions

Unit Number: 1	**Title: : Introduction to Cloud Computing**	**No. of hours: 11**

Content Summary:

Cloud Computing, Adoption of cloud-based IT resources, Service Models: Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS), Software-as-a-Service(SaaS), Deployment models: Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud, Cloud Computing Characteristics, Challenges of cloud computing,Virtualization concept, Types of virtualizations, Demo of virtualization, Virtualization Merits, Role of virtualization in cloud computing, Virtualization Demerits, VMPlacement, VM Migration, VM Migration Demo, VM clustering, Design Issues in VM Clustering, Need of Dockers and Containers, Docker Eco-System, Hypervisor vs Docker.

Unit Number: 2	**Title: Microservices**	**No. of hours: 10**

Content Summary:

Microservices, Service-Oriented Architecture, REST API, IP Addressing, Subnetting, Supernetting, Designing of Virtual Private Cloud, Demo of VPC, VPC Peering,VPC Case Study, Cloud Storage, Serverless Computing, Cloud API Gateway, Cloud Databases, Resource Provisioning, Time shared and space shared, Efficient VMConsolidation on cloud server, Task/DAG Scheduling Algorithms, Min-Min, Max-Min, MET, B-level Demo, T-level Demo, Task-VM Mapping, Auto Scaling, Load Balancing.

Unit Number: 3	**Title: Case Study**	**No. of hours: 08**
Content Summary: Case Study: Cloud Market analysis, Security and Compliances, Shared security model in IAAS/PAAS/SAAS, Shared technology issues, Data loss or leakage, Accountor service hijacking, Implementation of cloud security, Security Groups, Network Access Control Lists, Cloud databases, Parallel Query Execution with NoSQLDatabase, Big Data, Handling Big Data on Cloud Platform, Map- Reduce framework for large clusters using Hadoop, Design of data applications based on Map Reducein Apache Hadoop.		
Unit Number: 4	**Title: : Comparative study/analysis of public clouds**	**No. of hours: 09**
Content Summary: Comparative study/analysis of public clouds, Edge Computing, Fog Computing, Data Offloading, Cloud-Based DevOps Tools, Task Partitioning, Data Partitioning, Data Synchronization, Distributed File System, Data center, Ongoing Research Topics.		
Self-Learning Component: - The students are expected to choose a topic in discussion with the industry expert and implement the concepts of cloud computing. The student should present the progress at the end of the semester.		
Reference Books: 1. Lizhe Wang, Rajiv Ranjan, Jinjun Chen and Boualem Benatallah, Cloud Computing (1 ed.), CRC Press, 2017. ISBN 978-1351833097. 2. Judith S. Hurwitz and Daniel Kirsch, Cloud Computing For Dummies (2 ed.), Hoboken: John Wiley & Sons, 2020. ISBN 978-1119546658. 3. Prerna Sharma, Moolchand Sharma and Mohamed Elhoseny, Applications of Cloud Computing (1 ed.), CRC Press, 2020. ISBN 9780367904128.		

Define Course Outcomes (CO)

COs	**Statements**
CO1	Explain the core concepts of the cloud computing paradigm: how and why this paradigm shift came about, the characteristics, advantages and challenges brought about by the various models and services in cloud computing.
CO2	Apply the fundamental concepts in datacenters to understand the tradeoffs in power, efficiency and cost.
CO3	Identify resource management fundamentals, i.e. resource abstraction, sharing and sandboxing and outline their role in managing infrastructure in cloud computing.
CO4	Analyze various cloud programming models and apply them to solve problems on the cloud.

COs Mapping with Levels of Bloom’s taxonomy

CO	Cognitive levels© 1. Knowledge 2. Understand 3. Apply 4. Analyze 5. Evaluate 6. Create	Affective levels(A) 1. Receiving 2. Responding 3. Valuing 4. Organizing 5. Characterizing	Psychomotor levels(P) 1. Imitation 2. Manipulation 3. Precision 4. Articulation 5. Improving
CO1	C3	A3	P4
CO2	C3	A4	P2
CO3	C2	A2	P1
CO4	C4	A5	P5

CO-PO Mapping

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

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	**Introduction to Design thinking**
Local	-
Regional	-
National	-
Global	It promotes international collaboration, data sharing, and connectivity, facilitating global trade, research, and innovation.
Employability	Proficiency in cloud computing technologies, service models, and deployment models enhances employability and career prospects.
Entrepreneurship	-
Skill Development	This includes skills in virtualization, cloud service models, deployment models, and containerization.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Case studies in Design thinking**
Local	-
Regional	-
National	-
Global	In the global context, microservices, service-oriented architecture, and cloud computing are essential technologies driving digital innovation and transformation. Global organizations heavily rely on these technologies to deliver scalable and resilient applications and services to a global user base.
Employability	Acquiring skills in microservices, service-oriented architecture, and cloud computing enhances employability prospects in the IT industry. Proficiency in these technologies is in high demand as organizations increasingly adopt cloud-based architectures and microservice-oriented approaches
Entrepreneurship	-

Skill Development	The course on microservices, service-oriented architecture, and cloud computing promotes skill development in areas such as cloud infrastructure design, API development, virtualization, storage management, and resource provisioning
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Design Frameworks**
Local	-
Regional	-
National	-
Global	It helps address global challenges such as data privacy, security breaches, and international data transfer regulations.
Employability	Knowledge of cloud market analysis, security, and compliance enhances employability in various roles, including cloud architects, cloud security specialists, and cloud consultants.
Entrepreneurship	-
Skill Development	Studying cloud market analysis, security, and compliance develops critical skills such as risk assessment, security implementation, and compliance management.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Innovation & Creativity**
Local	-
Regional	-
National	-
Global	Cloud computing is a global phenomenon, and a course on comparative study/analysis of public clouds and related topics provides students with a global perspective on cloud technologies.
Employability	Proficiency in cloud computing technologies and understanding the different deployment models, such as public clouds and edge computing, enhances employability prospects in the IT industry.
Entrepreneurship	-
Skill Development	A course on comparative study/analysis of public clouds and related topics enhances various skills such as research, critical analysis, problem-solving, and technical proficiency.
Professional Ethics	-

Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4, 9
NEP 2020	Integration of Emerging Technologies and Skill Development
POE/4th IR	The course on cloud computing aligns with the Fourth Industrial Revolution (IR 4.0) in the following ways: Advanced Infrastructure, Data Management and Analytics and Digital Transformation.

Department:	**Department of Computer Applications**		
Course Name: **Computational Services in the Cloud Lab**	**Course Code** :ENSP451	**L-T-P**	**Credits**
		0-0-2	1
Type of Course:	Departmental Elective I -Minor		
Pre-requisite(s), if any: Concepts of Digital Electronics			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Apply the concept cloud computing to solve practical use cases.
CO 2	Analyzing different services in cloud computing
CO 3	Evaluate different available services provided by cloud vendors
CO 4	Design Cloud based application

List of Programs

Ex No	Experiment Title	Mapped CO/COs
1	Install Virtualbox / VMware Workstation with different flavours of linux or windows OS	CO1
2	Install a C compiler in the virtual machine created using virtual box and execute Simple Programs	CO1
3	Install Google App Engine. Create hello world app and other simple web applications using python/java.	CO1
4	Use GAE launcher to launch the web applications.	CO1

5	Simulate a cloud scenario using CloudSim and run a scheduling algorithm that is not present in CloudSim.	CO2
6	Find a procedure to transfer the files from one virtual machine to another virtual machine	CO2
7	Find a procedure to launch virtual machine using trystack (Online Openstack Demo Version)	CO2
8	Install Hadoop single node cluster and run simple applications like wordcount.	CO1
9	Deploy a simple web application using a Platform as a Service (PaaS) offering like AWS Elastic Beanstalk, Azure App Service, or Google App Engine	CO4
10	Create and manage databases using services like AWS RDS, Azure SQL Database, or Google Cloud SQL.	CO3
11	Containerize an application using Docker and create a container registry on a cloud platform.	CO3
12	Deploy and manage containers using container orchestration tools like AWS ECS, Azure Kubernetes Service (AKS), or Google Kubernetes Engine (GKE)	CO4
13	Develop and deploy a serverless function using AWS Lambda, Azure Functions, or Google Cloud Functions	CO4
14	Configure event triggers, access permissions, and monitoring for serverless functions	CO2
15	Utilize cloud-based data analytics tools like AWS Athena, Azure Data Lake Analytics, or Google BigQuery to query and analyze large datasets	CO2
16	Build and train machine learning models using cloud-based services like AWS SageMaker, Azure Machine Learning, or Google Cloud AutoML	CO2
17	Use infrastructure provisioning tools such as AWS CloudFormation, Azure Resource Manager, or Google Cloud Deployment Manager to define and deploy infrastructure components	CO1
18	Set up cloud monitoring and logging services like AWS CloudWatch, Azure Monitor, or Google Cloud Monitoring to track the performance and health of cloud resources	CO3
19	Implement access control policies and roles using AWS IAM, Azure Active Directory, or Google Cloud Identity and Access Management (IAM)	CO4
20	Enable encryption for data at rest and in transit using cloud security services	CO3

21	Explore cost optimization techniques like auto-scaling, spot instances, or reserved instances to optimize cloud resource usage and reduce costs	CO2

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: Microsoft Azure Cloud Fundamentals	**Course Code**	**L-T-P**	**Credits**
	ENSP403	4-0-1	4
Type of Course:	Departmental Elective I -Minor		
Pre-requisite(s), if any:			

Brief Syllabus:

The Microsoft Azure Cloud Fundamentals course introduces the concepts of cloud computing and the Azure platform. It covers Azure services such as Virtual Machines, Storage, Networking, Identity, App Services, and Databases. The course focuses on security, monitoring, and management in Azure. Real-world case studies and hands-on labs enable practical application. The subject equips students to design trustworthy intrusion detection systems and enhances security in IoT networks using Azure.

UNIT WISE DETAILS

Unit Number: 1	**Title: Introduction to Cloud Computing**	**No. of hours: 12**

Content Summary: Introduction to Cloud Computing and its Characteristics, Benefit, and Challenges of cloud computing. Cloud service models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS). Cloud Deployment Models, for example, Public, private, hybrid, and community clouds. shared responsibility model. Identify appropriate use cases for each cloud model. Consumption-based model. Comparison of cloud pricing models.

Unit Number: 2	**Title: Introduction to Microsoft Azure**	**No. of hours: 10**

Content Summary: Microsoft Azure cloud platform and its significance in the industry, Azure regions, and availability zones for understanding global data center distribution. Introduction of various Azure services and solutions available for different scenarios.

Unit Number: 3	**Title: Azure Virtual Machines (VMs) and Storage**	**No. of hours: 10**

Content summary: Create and manage virtual machines using Azure. Different VM sizes and types based on performance requirements. VM scaling and load balancing for optimizing application performance. Azure storage services: Blob Storage, Table Storage, File Storage, and Disk Storage.

Unit Number: 4	**Title: Azure Networking, Identity and Access Management**	**No. of hours: 12**

Content Summary: creation and configuration of virtual networks and subnets in Azure. Azure Load Balancer for distributing incoming network traffic, VPN Azure Active Directory (Azure AD) for managing identities and authentication. Gateway for secure communication between on-premises networks, and Azure. Azure Active Directory (Azure AD) for managing identities and authentication. Database Services, Azure Storage Account for data storage and retrieval.

Reference Books:

1. **Microsoft Azure Essentials Fundamentals of Azure by Michael S. Collier and Robin E. Shahan**
2. **Mastering Microsoft Azure Infrastructure Services by John Savill**
3. **Azure for Architects by Ritesh Modi**

Text Books

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

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** Microsoft Azure concepts
CO2	**Express** proficiency in the handling of Azure services
CO3	**Determine** methods to create and manipulate virtual machines
CO4	**Identify** commonly used models to implement cloud network
CO5	**Articulate** Azure database services

COs Mapping with Levels of Bloom's taxonomy

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

CO1	C2	–	P1
C02	C3	–	P2
CO3	C3	–	P4
CO4	C1	–	-
CO5	C1	–	P4

CO-PO Mapping

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	**Introduction to Cloud Computing**
Local	
Regional	
National	
Global	Develops skills in using cloud-based services
Employability	Develops skills in using cloud-based services
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills regarding Cloud concepts as well as in cloud computing, network management, and Azure services
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	**Introduction to Microsoft Azure**
Local	Understanding of Microsoft Azure services
Regional	-
National	-
Global	Aligns with global trends of Cloud
Employability	Develops cloud-based Knowledge for cloud architects
Entrepreneurship	-
Skill Development	Develops conceptual knowledge of Cloud computing
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Azure Virtual Machines (VMs) and Storage**
Local	
Regional	-
National	-
Global	Designing Virtual machines in Azure and storage management
Employability	Develops skills regarding the cloud security and cloud architecture
Entrepreneurship	-

Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Azure Networking, Identity, and Access Management**
Local	
Regional	-
National	-
Global	Access management of data
Employability	Develops skills in cloud security and network management
Entrepreneurship	-
Skill Development	Develops knowledge and skills in data storage and access management
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	Aligns with the concepts of technology and coding

Department:	**Department of Computer Applications**		
Course Name: **Microsoft Azure Cloud Fundamental Lab**	**Course Code** :ENSP453	**L-T-P**	**Credits**
		0-0-2	1
Type of Course:	Departmental Elective I -Minor		
Pre-requisite(s), if any: Concepts of Digital Electronics			

Proposed Lab Experiments

Defined Course Outcomes

COs	
CO 1	Acquire a practical understanding of cloud computing through hands-on lab experiments.
CO 2	Develop proficiency in using global data centers.
CO 3	Demonstrate the ability to create virtual machines and handle data storage
CO 4	Analyze and maintain the security and access management system.

Ex. No	Experiment Title	Mapped CO/COs
1	Familiarize students with the lab environment, software, and tools.	CO1
2	Creating and Managing Virtual Machines with Virtual Box	CO1
3	Create and configure virtual machines running different operating systems (e.g., Windows and Linux).	CO2
4	Install and configure Next Cloud on a local server or cloud-based virtual machine.	CO2
5	Set up your (user) accounts and storage quotas.	CO2
6	Try to implement -Test file uploads, sharing, and collaboration features	CO1
7	Install pfSense as a virtual router/firewall.	CO2

8	Configure virtual networks, subnets, and VLANs	CO2
9	Network Setup with pf Sense: for Test routing, port forwarding, and firewall rules.	CO1
10	Identity and Access Management with Key Cloak: Explore user authentication methods, roles, and permissions.	CO3
11	Identity and Access Management with Key Cloak Configure single sign-on (SSO) for different applications.	CO3
12	Install Key Cloak as an identity provider on a virtual machine.	CO3
13	Install Azure CLI and PowerShell on your machines.	CO3
14	Configure single sign-on (SSO) for different applications.	CO3
15	Use Azure CLI and PowerShell to create and manage Azure resources (e.g., VMs, storage accounts).	CO3
16	Create a simple web app (e.g., using HTML/CSS/JS or a web framework).	CO3
17	Deploy the web app to Azure App Service using Azure portal or Azure CLI.	CO3
18	Test the app's accessibility and scalability	CO4
19	Create an Azure SQL Database instance.	CO4
20	Monitor security alerts and take remedial actions.	CO4

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: Storages and Databases on Cloud	**Course Code** :ENSP405	**L-T-P**	**Credits**
		4-0-0	4
Type of Course:	Departmental Elective I -Minor		
Pre-requisite(s), if any:			

Brief Syllabus:
The course on cloud databases and storage provides a comprehensive understanding of the principles, technologies, and best practices associated with storing and managing data in the cloud. The syllabus covers various topics, starting with an introduction to cloud computing and an exploration of different types of cloud storage and databases, including object storage, block storage, file storage, relational databases, NoSQL databases, and more. Students delve into popular cloud storage and database services, such as Amazon S3, Google Cloud Storage, and Azure Blob Storage gaining practical knowledge of their features, deployment options, scalability, and high availability. The curriculum also includes essential aspects like database design, data migration, security measures, backup and recovery strategies, performance optimization, and monitoring techniques. Real-world case studies provide insights into organizations utilizing cloud storage and databases effectively.

UNIT WISE DETAILS

Unit Number: 1	**Title: Introduction to Storage on cloud**	**No. of hours: 4**

Content Summary:
Introduction to Cloud Computing, Overview of cloud databases and cloud storages, types of cloud storages(Object, block and file), different types of cloud database management systems, Gartner Magic Quadrant for Cloud Database Management Systems, Advantages of Working with Cloud Databases, Considerations for Cloud Databases, Top Cloud Database, Factors that help in choosing the right cloud database, Challenges involved in using cloud storages and databases.

Unit Number: 2	**Title: Data Integration, Migration, Security and performance on cloud**	**No. of hours: 8**

Content Summary:
Techniques, tool, methods and considerations for migrating from premise database to cloud databases
Backup, Recovery, and Disaster Planning including automated backups, point-in-time recovery and replication

Performance Optimization and Monitoring including query optimization, indexing, caching, and monitoring tools
Scalability and High Availability: load balancing, replication, sharding, and auto-scaling, Cloud Data Warehousing

Unit Number: 3	**Title: AWS Cloud Storage**	**No. of hours: 8**

Content Summary:
Introduction to AWS cloud storage, AWS management console, AWS Storage Services, Uploading files and images , Creating a web server, Overview of Amazon S3, Storage Classes, EC2 Instance Storage, network file system Amazon Elastic Block Store, Amazon Elastic file system, Amazon Cloud Front.
Brief introduction to Google Cloud Storage, and Azure Blob Storage.

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

Content Summary: Case Studies and Real-world Examples of Netflix , Airbnb, Pinterest, spotify, coca-cola etc. Analyzing real-world use cases of organizations using cloud storage and databases, discussing architecture decisions, challenges, and lessons learned.

***Self-Learning Components**:

1) Explore open-source projects related to cloud databases and storage. discover open-source projects from GitHub, learn from the codebase, and contribute to the development of cloud-based storage and database solutions. (https://github.com/topics/cloud-database)

2) Join the course : https://www.coursera.org/specializations/cloud-computing

2) Join Online Courses like

- https://www.udemy.com/topic/cloud-computing/
- https://www.coursera.org/courses?query=data%20storage
- https://www.mygreatlearning.com/academy/learn-for-free/courses/databases-and-files-systems-in-aws
- https://www.youtube.com/watch?v=EN4fEbcFZ_E
- https://www.codecademy.com/catalog/subject/cloud-computing
- https://www.simplilearn.com/tutorials/cloud-computing-tutorial

2) Read case studies of organizations that have successfully implemented cloud storage and database solutions.

- **Case Study of Delloite:** https://www2.deloitte.com/us/en/pages/consulting/articles/cloud-computing-case-studies.html
- **Case Study of Amazon:** https://aws.amazon.com/solutions/case-studies/amazon/

Learn about their challenges, architectural decisions, and best practices they adopted.

Reference Books:

1. "Database Cloud Storage: The Essential Guide to Oracle Automatic Storage Management" by Nitin Vengurlekar, 2013

2. "Cloud Database Development and Management" by Lee chao, 2013

3. "Advancing Cloud Database Systems and Capacity Planning with Dynamic Applications" by Narendra Kumar Kamila, 2017

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** & Recall the fundamental concepts and principles of cloud storage and databases. Interpret the security considerations and best practices for cloud storage and databases.
CO2	**Express** ideas and solutions for optimizing performance and improving efficiency in cloud storage and databases through indexing, caching, and query optimization techniques.
CO3	**Determine** the requirements of an application or system and determine the most suitable cloud storage and database solutions to meet those requirements.
CO4	**Identify** different types of cloud storage and database services available.
CO5	**Articulate** the best practices and considerations for designing scalable, reliable, and secure cloud storage and database architectures.
CO6	**Design** data storage architecture for cloud-based applications.

COs Mapping with Levels of Bloom's taxonomy

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

***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	2	2
CO2	2	3	1	1	1	1	1	1	2	2
CO3	2	2	3	2	2	3	2	2	3	3
CO4	1	2	-	3	1	1	-	-	2	2
CO5	-	2	1	1	3	2	-	-	2	3
CO6	1	1	2	-	2	3	1	1	3	3

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	Introduction to cloud databases and storages
Local	local considerations may include factors like local network connectivity, data centre locations, and latency between local systems and cloud databases.
Regional	regional considerations can include regulatory compliance specific to a particular region, data sovereignty issues, and availability zones
National	national considerations such as national data protection laws and regulations, government policies related to cloud computing, and national cybersecurity guidelines that may impact the usage and storage of data in cloud databases.
Global	Aligns with global trends in internet technologies and network protocols accessibility of cloud databases and storage options across multiple regions, data replication and backup strategies for disaster recovery on a global scale
Employability	The unit syllabus may indirectly contribute to employability by introducing students to cloud computing concepts, cloud databases, and storage technologies
Entrepreneurship	-
Skill Development	The unit syllabus primarily focuses on developing skills related to cloud computing, cloud databases, and storage
Professional Ethics	-
Gender	-

Human Values	-
Environment & Sustainability	-
Unit II	Data Integration, Migration, Security and performance on cloud
Local	local considerations may include factors like local network connectivity, data centre locations, and latency between local systems and cloud databases.
Regional	regional considerations can include regulatory compliance specific to a particular region, data sovereignty issues, and availability zones
National	national considerations such as national data protection laws and regulations, government policies related to cloud computing, and national cybersecurity guidelines that may impact the usage and storage of data in cloud databases.
Global	Aligns with global trends in internet technologies and network protocols accessibility of cloud databases and storage options across multiple regions, data replication and backup strategies for disaster recovery on a global scale
Employability	local considerations may include factors like local network connectivity, data centre locations, and latency between local systems and cloud databases.
Entrepreneurship	regional considerations can include regulatory compliance specific to a particular region, data sovereignty issues, and availability zones
Skill Development	national considerations such as national data protection laws and regulations, government policies related to cloud computing, and national cybersecurity guidelines that may impact the usage and storage of data in cloud databases.
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	AWS Cloud Storage
Local	Primarily focuses on cloud storage technologies and their local implementation within the AWS ecosystem.

Regional	-
National	Contributes to national network security strategies and protocols
Global	AWS, Google Cloud, and Azure are global cloud service providers, enabling businesses worldwide to access their storage services and infrastructure.
Employability	Acquiring knowledge and skills in these cloud storage platforms enhances employability in the field of cloud computing and storage management.
Entrepreneurship	Knowledge of AWS, Google Cloud, and Azure storage services can empower entrepreneurs to build and scale their businesses by utilizing scalable and reliable cloud storage solutions for their applications and data.
Skill Development	The unit syllabus focuses on developing skills related to AWS cloud storage, including uploading files and images, creating a web server, utilizing different storage services
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	Case Studies
Local	Addresses local understanding and implementation of internet-based services
Regional	-
National	Contributes to national digital communication strategies and multimedia applications
Global	Aligns with global trends in internet telephony, multimedia applications, and SEO
Employability	Develops skills in internet telephony, multimedia applications, and SEO
Entrepreneurship	-
Skill Development	Develops knowledge and skills in internet telephony, multimedia applications, and SEO
Professional Ethics	-
Gender	-
Human Values	-

Environment & Sustainability	-
SDG	SDG 4
NEP 2020	It supports NEP's objective of enhancing the use of technology in education and research by providing students with knowledge and skills in cloud databases and storage, which are essential components of modern digital infrastructure
POE/4th IR	The course can incorporate discussions on ethical considerations in cloud databases and storage, addressing issues such as data privacy, security, and responsible data management.

Department:	**Department of Computer Applications**			
Course Name: **Storage and Databases on Cloud Lab**	**Course Code** :ENSP455	**L-T-P**	**Credits**	
		0-0-2	1	
Type of Course:	Departmental Elective I -Minor			
Pre-requisite(s), if any: Concepts of Digital Electronics				

Proposed Lab Experiments

Defined Course Outcomes

COs	Understand the principles and concepts of cloud storage and databases.
CO 1	Gain hands-on experience in using cloud storage services for data storage and retrieval.
CO 2	Implement backup and recovery strategies for cloud-based storage and databases.
CO 3	Optimize database performance in the cloud through indexing and query optimization techniques.
CO 4	Analyse real-world case studies to understand the challenges and solutions related to cloud storage and databases.

Ex. No	**Experiment Title**	**Mapped CO/COs**
1	**Create accounts and configure cloud storage services such as Amazon S3, Google Cloud Storage, or Azure Blob Storage:** Practice creating buckets/containers, uploading files, setting access permissions, and managing storage resources.	CO1, CO4, CO5
2	**Perform operations on object storage, including uploading, downloading, and deleting files:** Explore advanced features like versioning, lifecycle policies, and metadata management.	CO1, CO2, CO5
3	**Perform Block Storage Configuration:** Create and attach storage volumes to virtual machines, perform formatting and mounting, and understand snapshotting and resizing operations.	CO1, CO3, CO5
4	**File Storage Implementation:** Work with shared file systems such as Amazon EFS, Google Cloud Filestore, or Azure Files. They configure file shares, mount them on virtual machines, and explore features like file locking and access control.	CO1, CO4, CO5

5	**Database Provisioning and Management:** • learn to set up and configure cloud databases like Amazon RDS, Google Cloud SQL, or Azure Database Services • create database instances, manage security settings, and perform basic administration tasks such as backups and restores.	CO1, CO3, CO4, CO5
6	**Data Migration to the Cloud:** • Practice migrating databases from on-premises or other cloud providers to the selected cloud database service. • Explore different migration methods, tools, and validate the successful transfer of data	CO1, CO3, CO4, CO5
7	**Database Design and Schema Management:** • Work on designing and creating database schemas for various scenarios. • Implement tables, relationships, indexes, and constraints to support specific use cases and understand database modelling concepts.	CO1, CO3, CO5, CO6
8	**Data Security and Access Control:** Configure security measures for cloud databases, including setting up user accounts, managing roles and permissions, and implementing encryption techniques to protect data at rest and in transit	CO1, CO5
9	**Performance Tuning and Optimization:** Learn performance tuning techniques for cloud databases, including query optimization, index creation, caching strategies, and monitoring tools to identify and resolve performance bottlenecks	CO1, CO2, CO5
10	**High Availability and Scalability:** • Explore features like replication, automatic scaling, and load balancing to ensure high availability and scalability of databases in the cloud • Configure and test failover scenarios and evaluate the performance impact of scaling operations.	CO1, CO3, CO5
11	**Project** **Backup and Recovery Strategies:** • Practice implementing backup and recovery mechanisms for cloud databases • perform regular backups, schedule automated backups, and simulate recovery scenarios to restore databases to a consistent state.	CO1, CO5
12	**Project** **Cloud Data Warehousing:** • Gain hands-on experience with cloud data warehousing platforms like Amazon Redshift • load data, run complex queries, and analyze performance optimization techniques for data warehousing.	CO1, CO2, CO5, CO6
13	**Project** Design and implement an inventory management system that utilizes cloud storage for storing inventory data. Include features such as real-time inventory tracking, order management, and reporting capabilities.	CO1, CO2, CO6

14	**Project** Build a document management system that leverages cloud storage for storing and organizing documents. Include features like full-text search, document tagging, and access control for secure document sharing.	CO1, CO2, CO5s

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: **APPLICATION DEVELOPMENT AND DEVOPS ON CLOUD**	**Course Code**	**L-T-P**	**Credits**
	ENSP407	4-0-0	4
Type of Course:	Departmental Elective I (Minor)		

Pre-requisite(s), if any: Nil

Brief Syllabus:

The syllabus aims to equip students with practical skills and theoretical knowledge to design, develop, and deploy applications in cloud environments while implementing DevOps practices to enhance software development, delivery, and operations on the cloud. It prepares them for a career in the dynamic and rapidly growing field of cloud computing and DevOps, where demand for skilled professionals is high due to the increasing adoption of cloud technologies in various industries.

UNIT WISE DETAILS

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

Content Summary:
Overview of Cloud Computing: Definition, Characteristics, Advantages, and Disadvantages.
Cloud Service Models: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS).
Cloud Deployment Models: Public Cloud, Private Cloud, Hybrid Cloud, Community Cloud.
Cloud Providers: Introduction to popular cloud service providers (e.g., AWS, Azure, Google Cloud).

Cloud Architecture: Understanding cloud components, virtualization, and multi-tenancy.

Introduction to DevOps: Principles, Practices, and Benefits.

<table>
<tr><td><b>Unit Number: 2</b></td><td><b>Title: Cloud-Based Application Development</b></td><td><b>No. of hours: 10</b></td></tr>
<tr><td colspan="3"><b>Content Summary:</b>
Cloud Native Applications: Principles and architecture for building applications optimized for cloud environments.
Containerization: Docker and Kubernetes for application deployment, scaling, and management.
Microservices Architecture: Designing, developing, and deploying microservices-based applications.
Serverless Computing: Introduction to serverless architecture and Function as a Service (FaaS).
Application Security in the Cloud: Best practices for securing cloud-based applications.
Cloud-Based Databases: Database options and considerations in cloud environments.</td></tr>
<tr><td><b>Unit Number: 3</b></td><td><b>Title: DevOps Practices in Cloud</b></td><td><b>No. of hours: 10</b></td></tr>
<tr><td colspan="3"><b>Content Summary:</b>
Continuous Integration (CI) and Continuous Deployment (CD) in Cloud: Setting up CI/CD pipelines.
Infrastructure as Code (IaC): Automating infrastructure provisioning using tools like Terraform and CloudFormation.
Configuration Management: Managing application configurations in a cloud-based environment.
Monitoring and Logging in Cloud: Tools and techniques for monitoring application performance and gathering logs.
Auto-scaling and Load Balancing: Scaling applications dynamically based on demand.
High Availability and Disaster Recovery: Designing and implementing resilient applications in the cloud.</td></tr>
</table>

Unit Number: 4	Title: Cloud-Based DevOps Tools and Best Practices	No. of hours: 10

Content Summary:

Cloud-Based Version Control: Using Git and other version control tools in cloud-based development.

Collaboration and Communication Tools: Utilizing cloud-based collaboration tools for distributed teams.

Cloud-Based Testing and Quality Assurance: Strategies for testing applications in cloud environments.

Cost Management in Cloud: Optimizing cloud resource usage and cost control.

Performance Optimization in Cloud: Techniques for improving application performance in cloud environments.

Case Studies and Real-World Projects: Analyzing successful cloud-based application development and DevOps projects.

***SELF-LEARNING COMPONENTS**:

- https://elearn.nptel.ac.in/shop/iit-workshops/completed/azure-devops-and-micro-services-azure-kubernetes-deployment-models/
- https://nptel.ac.in/courses/106105167

Please Note:

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

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

Reference Books:

1. Jez Humble and David Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation, Pearson Education, Inc., 2011.
2. Thomas Erl, Ricardo Puttini, and Zaigham Mahmood, Cloud Computing: Concepts, Technology & Architecture, Prentice Hall, 2013.
3. Arun Eapen, Docker on Amazon Web Services: Build, deploy, and manage your container applications at scale on AWS, Packt Publishing, 2017.
4. Sam Newman, Building Microservices: Designing Fine-Grained Systems, O'Reilly Media, Inc., 2015.

5. Mark Richards and Neal Ford, Fundamentals of Software Architecture: An Engineering Approach, O'Reilly Media, Inc., 2020.

Define Course Outcomes (CO)

Cos	Statements
CO1	**Understand** the fundamental concepts and principles of cloud computing, including different cloud service models and deployment models. Gain insights into various cloud providers and their offerings to make informed decisions for application development and deployment.
CO2	**Acquire** practical knowledge and hands-on experience in developing cloud-native applications, utilizing containerization with Docker and orchestration using Kubernetes. Demonstrate proficiency in designing and implementing microservices-based architectures.
CO3	**Master** the principles and practices of DevOps in a cloud environment. Learn how to set up Continuous Integration (CI) and Continuous Deployment (CD) pipelines, automate infrastructure provisioning with Infrastructure as Code (IaC), and manage application configurations effectively.
CO4	**Develop** the skills to monitor, log, and optimize the performance of cloud-based applications. Explore auto-scaling and load balancing techniques to ensure high availability and disaster recovery strategies for resilient applications.
CO5	**Apply** cloud-based DevOps tools and best practices to collaborate effectively in distributed teams. Gain knowledge of cloud-based version control, testing, cost management, and performance optimization techniques to deliver efficient and cost-effective cloud solutions.

COs Mapping with Levels of Bloom's taxonomy

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

CO-PO Mapping

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

Justification for mapping must be relevant.

1=weakly mapped

2= moderately mapped

3=strongly mapped

CO-PSO Mapping

PO	PSO1	PSO2	PSO3	PSO4
CO1	3	1	2	-
CO2	2	1	-	-
CO3	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/4th IR	Aligns with the concepts of internet telephony, multimedia applications, and SEO

Department:	**Department of Computer Applications**		
Course Name: **APPLICATION DEVELOPMENT AND DEVOPS ON CLOUD LAB**	**Course Code**	**L-T-P**	**Credits**
	ENSP457	0-0-2	1
Type of Course:	Department Elective I- Minor		
Pre-requisite(s), if any: Nil			

Proposed Lab Experiments

Defined Course Outcomes

COs	**Course Outcomes (COs)**
CO 1	**Understand** the fundamental concepts of cloud computing, cloud service models, and cloud-based application development
CO 2	**Acquire** hands-on experience in deploying and managing applications on cloud platforms using various cloud services and tools
CO 3	**Implement** DevOps practices in cloud-based application development, including continuous integration, continuous deployment, and version control
CO 4	**Demonstrate** proficiency in using containerization (Docker and Kubernetes), Infrastructure as Code (IaC), and serverless computing for cloud-based application deployment and scaling

CO 5	**Develop** skills in application monitoring, logging, security, and performance optimization for cloud-based applications, and apply cost optimization strategies for cloud resources

Experiment No.	**Experiment Title**	**Mapped COs**
1	Setting up a cloud-based development environment	CO1
2	Deploying a web application on a cloud platform	CO1, CO2
3	Containerizing and deploying with Docker and Kubernetes	CO2, CO4
4	Implementing Continuous Integration (CI)	CO2, CO3
5	Implementing Infrastructure as Code (IaC)	CO2, CO4
6	Configuring auto-scaling for an application	CO2
7	Implementing high availability and disaster recovery	CO2
8	Building a serverless application	CO2, CO4
9	Implementing application monitoring and logging	CO2
10	Creating and managing a cloud-based database	CO2
11	Integrating third-party APIs with an application	CO2
12	Implementing security best practices	CO2

13	Managing application configurations	CO2
14	Performing load testing on an application	CO2
15	Migrating an application to the cloud	CO2
16	Implementing DevOps practices	CO3
17	Deploying a multi-tier application architecture	CO1, CO2
18	Integrating CI/CD pipeline with version control	CO2, CO3
19	Implementing blue-green deployment	CO3
20	Implementing Canary deployment	CO3
21	Performance optimization in cloud	CO2, CO4
22	Implementing feature flags	CO2, CO3
23	Implementing cost optimization strategies	CO2, CO5
24	Deploying a microservices-based application	CO2, CO4
25	Real-world project showcasing cloud-based development	CO1, CO2, CO3, CO4, CO5

1. **Exercise for Setting up a cloud-based development environment**: Exercise: Students set up an account with a cloud service provider (e.g., AWS, Azure) and create a virtual machine instance to host a basic web application.

2. **Project for Deploying a web application on a cloud platform**: Project: Students develop a simple web application and deploy it on a cloud platform using PaaS. They configure auto-scaling based on application demand and perform load testing to observe scaling behavior.

3. **Exercise for Containerizing and deploying with Docker and Kubernetes**: Exercise: Students containerize an existing application using Docker and deploy it on a Kubernetes cluster. They configure Kubernetes services, pods, and replicas for the application.

4. **Exercise for Implementing Continuous Integration (CI)**: Exercise: Students set up a CI/CD pipeline using a cloud-based CI/CD tool (e.g., Jenkins) to automatically build and test their application whenever code changes are pushed to the repository.

5. **Exercise for Implementing Infrastructure as Code (IaC)**: Exercise: Students use Terraform or CloudFormation to define and provision cloud resources for their application, such as virtual machines, storage, and networking.

6. **Project for Configuring auto-scaling for an application**: Project: Students design a scalable architecture for a multi-tier application and implement auto-scaling based on CPU utilization. They monitor application performance during auto-scaling events.

7. **Exercise for Implementing high availability and disaster recovery**: Exercise: Students configure a highly available architecture using load balancers and multiple availability zones to ensure application resilience. They test disaster recovery scenarios.

8. **Project for Building a serverless application**: Project: Students develop a serverless application using FaaS. They implement serverless functions to handle specific application features and integrate them with event triggers.

9. **Exercise for Implementing application monitoring and logging**: Exercise: Students set up monitoring and logging solutions (e.g., CloudWatch, Stackdriver) for their applications to collect metrics, logs, and perform analysis.

10. **Project for Creating and managing a cloud-based database**: Project: Students design a database schema and implement a cloud-based database (e.g., Amazon RDS, Azure SQL Database). They perform data migration and backup strategies.

11. **Project for Integrating third-party APIs with an application**: Project: Students integrate a popular third-party API (e.g., Google Maps API, Twitter API) into their application to enhance its functionality and showcase real-time data retrieval.

12. **Exercise for Implementing security best practices**: Exercise: Students configure Identity and Access Management (IAM) roles and policies to enforce security for their cloud-based applications. They implement Network Security Groups and security groups to control traffic flow.

13. **Exercise for Managing application configurations**: Exercise: Students use cloud-native configuration management tools (e.g., AWS Systems Manager, Azure Configuration Management) to manage application configurations across multiple instances.

14. **Project for Performing load testing on an application**: Project: Students use load testing tools (e.g., JMeter, Gatling) to simulate high user loads on their cloud-based applications and analyze performance metrics under heavy traffic.

15. **Project for Migrating an application to the cloud**: Project: Students migrate an existing on-premises application to a cloud platform, considering factors like data transfer, application dependencies, and post-migration testing.

16. **Project for Implementing DevOps practices**: Project: Students collaborate in teams to develop a complete cloud-based application and implement DevOps practices, including version control, CI/CD, and automated testing.

17. **Exercise for Deploying a multi-tier application architecture**: Exercise: Students design and deploy a multi-tier application on the cloud, comprising frontend web servers, application servers, and database servers.

18. **Project for Integrating CI/CD pipeline with version control and issue tracking tools**: Project: Students set up integration between a CI/CD pipeline and version control (e.g., Git) and issue tracking tools (e.g., Jira) for automated build and issue tracking.

19. **Exercise for Implementing blue-green deployment**: Exercise: Students implement blue-green deployment strategies to minimize downtime during application updates, allowing seamless rollback if issues arise.

20. **Exercise for Implementing Canary deployment**: Exercise: Students deploy a new feature of their application to a small subset of users using Canary deployment to gather feedback and monitor performance.

21. **Project for Performance optimization in the cloud**: Project: Students optimize the performance of their cloud-based application by tuning parameters, caching, and optimizing database queries for better response times.

22. **Project for Implementing feature flags**: Project: Students implement feature flags (feature toggles) in their application to enable/disable features dynamically and gradually release new functionalities.

23. **Project for Implementing cost optimization strategies**: Project: Students analyze the cost of their cloud resources and implement cost optimization techniques, such as using reserved instances and rightsizing resources.

24. **Project for Deploying a microservices-based application**: Project: Students design and deploy a microservices-based application architecture on the cloud, using containers and orchestrators like Kubernetes.

25. **Project for Real-world project showcasing cloud-based development**: Project: Students work on a real-world cloud-based development project, applying the concepts learned throughout the course to design, develop, deploy, and manage an application in the cloud.

COURSE TEMPLATE

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

Brief Syllabus: The objective of the course is to provide skills to develop applications for OS X and iOS. It includes introduction to development framework Xcode. Objective-C is used as programming language to develop the applications. Objective-C is the superset of the C programming language and provides object-oriented capabilities and a dynamic runtime. Objective-C inherits the syntax, primitive types, and flow control statements of C and adds syntax for defining classes and methods.		
UNIT WISE DETAILS		
Unit Number: 1	**Title: Introduction to IDE and SDK of iOS App Development**	**No. of hours: 10**
Content Summary: Xcode-The SDK environment, Supporting tools, Advance settings. Development Technique, Fundamental of Object-Oriented Programming, The MVC architecture.		
Unit Number: 2	**Title: Objective-C**	**No. of hours: 10**
Content Summary:		

<table>
<tr><td colspan="3">Introduction to Objective C, Primitive Data Types, Conditions, Loops, Functions, Arrays, Pointers, Structures, Classes, Objects, Foundation, Memory Management, Inheritance, Categories, Protocols, Predicates, Blocks, Multi-Threading.
Objects Send and Receive Messages concept, Use of Pointers to Keep Track of Objects, Methods - Return Values.</td></tr>
<tr><td><b>Unit Number: 3</b></td><td><b>Title: Encapsulating Data</b></td><td><b>No. of hours: 10</b></td></tr>
<tr><td colspan="3"><b>Content Summary:</b>
Properties of Encapsulation of an Object’s Values, Declare Public Properties for Exposed Data, Use Accessor Methods to Get or Set Property Values, Concept of Dot Syntax, Properties Are Backed by Instance Variables.
Dealing with Errors: Use NSError for Most Errors, Some Delegate Methods Alert You to Errors, Some Methods Pass Errors by Reference</td></tr>
<tr><td><b>Unit Number: 4</b></td><td><b>Title: Developing iOS Applications</b></td><td><b>No. of hours: 10</b></td></tr>
<tr><td colspan="3"><b>Content Summary:</b>
iOS App Anatomy, Design Principles, Creating a Basic Hello World App with interface elements, UI View & Controller, UI Elements, Trigger Actions, Storyboard, Device Orientations, Using Gestures, Popovers and Modal Dialogs, Creating Universal Apps, Status Bar, Navigation Bar, Tab Bar, Content Views (e.g. Image view, Map View etc.), UI Table View and Table View Controller, Core Data, Test your App, Publishing your App.</td></tr>
<tr><td colspan="3"><b>*Self-Learning Components</b>:
1. XCode Documentation
<b>References:</b>
1. https://www.tutorialspoint.com/objective_c/objective_c_quick_guide.htm
2. https://www.coursera.org/learn/introduction-to-ios-mobile-application-development
3. https://www.geeksforgeeks.org/classes-objects-in-objective-c/
<b>Please Note:</b>
<b>At least 5-10 % syllabus will be asked in end term exams from self-learning components</b></td></tr>
<tr><td colspan="3">Text Book:
1. Effective objective C 2.0, Matt Galloway, Effective software development series, Scott Meyers.
Reference Books:
1. Programming in Objective-C (5th Edition) (Developer's Library) by Stephen G. Kochan.
2. iOS 6 Development Unleashed: Developing Mobile Applications for Apple iPhone, iPad, and iPod Touch by Robert McGovern</td></tr>
</table>

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

Course Outcomes (CO)

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

COs Mapping with Levels of Bloom’s taxonomy

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

CO-PO Mapping

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

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

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

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

Employability	-
Entrepreneurship	-
Skill Development	-
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4, SDG 8, SDG 9
NEP 2020	Promoting universal access to education, holistic development, multidisciplinary approach, skill development, critical thinking, creativity, ICT integration, research and development, global competencies, and professional ethics.

Department:	**Department of Computer Applications**		
Course Name: **Mobile Application Development using iOS Lab**	**Course Code** :ENSP459	**L-T-P**	**Credits**
		0-0-2	1
Type of Course:	Departmental Elective II(Minor)		
Pre-requisite(s), if any: Concepts of Digital Electronics			

Defined Course Outcomes

COs	
CO 1	Create iPhone apps using Objective-C and Apple's new programming language, use industry tools and frameworks such as Cocoa, Xcode, UIKit, Git.
CO 2	Understand and know how to use properly UIKit, asynchronous code, Core Image, NSURL Session and JSON Map Kit and Core Location, Auto Layout, Source Control, Core Data, Animation, and the app submission process.
CO 3	Read and write programs based on Objective-C, also have a strong grasp of Objective-C objects
CO 4	Organize their code professionally using objects and blocks, prototype several entry- level apps and try to publish on App store.

Ex No	**Experiment Title**	**Mapped CO/COs**
1	Case Study of Objective-C language.	CO2
2	Case study of Windows and MAC systems	CO2
3	Case Study of XCode based on MAC Systems	CO2
4	Design an App for UISwitch based on Objective-C language	CO1
5	Design an App for UISlider based on Objective-C language	CO1
6	Design an App for UIStepper based on Objective-C language	CO1
7	Write a program for creating Story Boards	CO1
8	Design an App for UIAnimation based on Objective-C language	CO1

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

COURSE TEMPLATE

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

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

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

Content Summary:

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

***Self-Learning Components:**

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

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

Reference/Text Books:

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
C02	C3	A2	P2
CO3	C2	A3	P3
CO4	-	-	-
CO5	C5	-	P5

CO-PO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	**Introduction to .NET Framework**
Local	
Regional	
National	Understanding the evolution and history of the .NET Framework provides context specific to the national software development landscape.
Global	
Employability	Understanding the .NET Framework provides valuable skills and knowledge sought by employers in the software development industry.
Entrepreneurship	Understanding the .NET platform can support the development of innovative software products and services.
Skill Development	Studying the .NET Framework helps develop technical skills in application development and programming.
Professional Ethics	Familiarity with the .NET Framework enables professionals to adhere to ethical practices while developing software applications.
Gender	
Human Values	-
Environment & Sustainability	-
Unit II	**NET Framework Fundamentals**
Local	Understanding OOP in .NET can have local relevance in terms of specific programming practices and patterns adopted within the local software development community.
Regional	-

National	Understanding classes, objects, and inheritance in the .NET Framework is relevant at all levels of software development.
Global	Aligns with global trends in internet technologies and network protocols
Employability	Knowledge of classes, objects, and inheritance enhances employability in the software development field.
Entrepreneurship	Knowledge of OOP in .NET supports the development of innovative software products and services by entrepreneurs.
Skill Development	Learning about classes, objects, and inheritance improves technical skills required
Professional Ethics	
Gender	-
Human Values	-
Environment & Sustainability	-
Unit III	**Building Applications with .NET Framework**
Local	Understanding Windows Forms and WPF applications is relevant at a local level as it involves designing user interfaces and developing desktop applications specific to the local context.
Regional	Understanding Windows Forms and WPF applications is relevant at a regional level as these frameworks are commonly used in software development within a specific region.
National	Introduction to Windows Forms and WPF is important at a national level as these frameworks serve as the foundation for developing various types of applications used nationwide.
Global	Introduction to Windows Forms and WPF is important globally as these frameworks are fundamental to developing user interfaces and applications used on a global scale.
Employability	Understanding the basics of Windows Forms and WPF is crucial for employability in software development roles. Proficiency in these frameworks demonstrates competence and versatility, making individuals more desirable to potential employers.

Entrepreneurship	Knowledge of Windows Forms and WPF applications enables entrepreneurs to create innovative software products and services, driving business growth and success.
Skill Development	Understanding Windows Forms and WPF applications enhances technical skills in software development, enabling individuals to design and develop user-friendly and visually appealing applications.
Professional Ethics	Familiarity with Windows Forms and WPF applications ensures adherence to ethical standards in software development, including data privacy, accessibility, and industry best practices for usability and security.
Gender	-
Human Values	-
Environment & Sustainability	-
Unit IV	**Advanced Topics in .NET Framework**
Local	the content on .NET Core, Entity Framework, and Windows Communication Foundation (WCF) has local relevance as it addresses the specific development and database connectivity needs within the local context.
Regional	-
National	The content on .NET Core, Entity Framework, and Windows Communication Foundation (WCF) is nationally relevant as it addresses specific development and database connectivity needs within the country, considering national requirements and technologies.
Global	addresses development and database connectivity needs on a global scale, considering international requirements and technologies.
Employability	Content equips individuals with the skills and knowledge necessary for software development roles. Proficiency in these technologies enhances employability prospects and opens up opportunities in the job market.
Entrepreneurship	Understanding these technologies enables entrepreneurs to create and scale their own software ventures.
Skill Development	Learning and applying these technologies contribute to the development of practical skills that are in demand in the industry, improving professional capabilities and career prospects

Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 9, SDG 4, and SDG 8,
NEP 2020	-
POE/4th IR	the content on .NET Core, Entity Framework, and Windows Communication Foundation (WCF) addresses professional ethics and aligns with the demands and innovations of the Fourth Industrial Revolution (4IR).

Department:	**Department of Computer Applications**		
Course Name: **.Net Framework Lab**	**Course Code** :ENSP463	**L-T-P**	**Credits**
		0-0-2	1
Type of Course:	Departmental Elective II (Minor)		
Pre-requisite(s), if any: NA			

Proposed Lab Experiments

Defined Course Outcomes

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

Ex. No	Experiment Title	Mapped CO/COs
1	Installing and setting up the .NET Framework, Visual Studio IDE, and NuGet package manager	CO1
2	Creating a basic console application in C# or Visual Basic.NET and running it in Visual Studio.	CO1
3	Write a program to display "Hello World" using C#.	CO2
4	Create a Windows Forms application to design a simple calculator.	CO2
5	Develop a console application to perform basic arithmetic operations	CO2

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

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: New-Age programming languages (GO, F#, Clojure, Kotlin)	**Course Code** : ENSP415	**L-T-P**	**Credits**
		4-0-0	4
Type of Course:	Departmental Elective II (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

<table>
<tr><td><b>Unit Number: 3</b></td><td><b>Title: Introduction to Clojure Programming</b></td><td><b>No. of hours: 10</b></td></tr>
<tr><td colspan="3"><b>Content Summary:</b>
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</td></tr>
<tr><td><b>Unit Number: 4</b></td><td><b>Title: Introduction to Kotlin Programming</b></td><td><b>No. of hours: 10</b></td></tr>
<tr><td colspan="3"><b>Content Summary:</b>
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.</td></tr>
<tr><td colspan="3"><b>*Self-Learning Components:</b>
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
<b>Please Note:</b>
<b>At least 5-10 % syllabus will be asked in end term exams from self-learning components</b></td></tr>
</table>

Reference Books:

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

Online References:

1. https://gobyexample.com/ [
2. https://golang.org/doc/
3. https://www.youtube.com/playlist?list=PLIxmoA0rQ-LwgK1JsnMsakYNACYGa1cjR
4. https://kotlinlang.org/docs/home.html
5. https://docs.microsoft.com/en-us/dotnet/fsharp/
6. https://www.udemy.com/course/learning-functional-programming-with-f/
7. https://clojure.org/guides/getting_started

Course Outcomes (CO)

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

CO5	**Design and implement** projects that integrate multiple programming languages, using appropriate inter-language communication mechanisms and libraries (e.g., Go and Kotlin interacting via REST APIs, F# and Clojure communicating via message queue

COs Mapping with Levels of Bloom's taxonomy

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

CO-PO Mapping

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

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	Introduction to New-Age Programming Languages and GO programming Language
Local	-
Regional	-
National	Provides essential knowledge and skills related to modern programming languages like GO, F#, Clojure, and Kotlin, which are widely used in national software development projects and initiatives.
Global	Relevant in the global software development community, as these languages and concepts have international adoption and usage.
Employability	Highly valued in the job market, as these languages are used in various industries and offer opportunities for software development roles.
Entrepreneurship	Equips students with knowledge of modern programming languages and advanced concepts, enabling them to explore entrepreneurial opportunities in software development and innovation using these languages.
Skill Development	Enhances students' technical skills in software development, making them more competent in the field of computer science.
Professional Ethics	While not directly related to professional ethics, the syllabus indirectly promotes ethical practices by emphasizing the importance of error handling, writing efficient and concurrent programs, and following best practices in software development.
Gender	-
Human Values	Indirectly supports human values by fostering the development of software solutions that are efficient, maintainable, and user-friendly, aligning with values such as accessibility, usability, and user-centric design.

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

Employability	Language is highly valued in the job market. The skills are sought after by companies and organizations involved in software development, web application development, and data analysis.
Entrepreneurship	Explores entrepreneurial opportunities in software development, web application startups, and data-centric solutions
Skill Development	Enhances technical skills, making more competent in the field of computer science
Professional Ethics	Indirectly promotes ethical practices by emphasizing good programming practices, data integrity, and security considerations.
Gender	-
Human Values	Indirectly supports human values by fostering the development of software solutions that are efficient, maintainable, and user-friendly, aligning with values such as accessibility, usability, and user-centric design. .
Environment & Sustainability	-
Unit IV	Introduction to Kotlin Programming
Local	-
Regional	-
National	Provides essential knowledge relevant at the national level for software development, mobile app development, and general-purpose programming.
Global	Language has global applicability.
Employability	Highly valued in the job market. These skills are sought after by companies and organizations involved in software development, mobile app development, and DSL-based solutions.
Entrepreneurship	Explore entrepreneurial opportunities in software development, mobile app startups, and domain-specific language development
Skill Development	Make more competent in the field.
Professional Ethics	Indirectly promotes ethical practices by emphasizing good programming practices, code readability, and modularity.
Gender	-
Human Values	Indirectly by fostering the development of software solutions efficient, maintainable, and user-friendly.

Environment & Sustainability	-
SDG	SDG 4, SDG 8, SDG 9
NEP 2020	Quality education, equity, critical thinking, digital literacy, skill development.
POE/4th IR	Technological advancements, digital transformation, and future-ready skills.

Department:	**Department of Computer Applications**		
New Age Programming languages Lab	**Course Code**	**L-T-P**	**Credits**
	ENSP415	0-0-2	1
Type of Course:	Departmental Elective-II (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.

Department:	**Department of Computer Applications**		
Course Name: New-Age programming languages (GO, F#, Clojure, Kotlin) **Lab**	**Course Code** :ENSP465	**L-T-P**	**Credits**
		0-0-2	1
Type of Course:	Departmental Elective II(Minor)		
Pre-requisite(s), if any:			

Proposed Lab Experiments

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

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

<table>
<tr><td>15</td><td colspan="2">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.</td><td>CO4</td></tr>
<tr><td>16</td><td colspan="2"><b>Mini Project</b>: 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.</td><td>CO5</td></tr>
<tr><td colspan="4">Practicals on Clojure Programming Language</td></tr>
<tr><td>17</td><td colspan="2">Write a program that demonstrates the basic syntax and data structures in Clojure, such as lists, vectors, maps, and sets.</td><td>CO1</td></tr>
<tr><td>18</td><td colspan="2">Write functions that manipulate and transform sequences using operations such as map, filter, reduce, and take.</td><td>CO2</td></tr>
<tr><td>19</td><td colspan="2">Implement a program that showcases asynchronous programming using the core.async library.</td><td>CO3</td></tr>
<tr><td>20</td><td colspan="2">Write code that calls Java methods, creates Java objects, and works with Java collections and objects from Clojure.</td><td>CO4</td></tr>
<tr><td>21</td><td colspan="2">Develop a web application using Clojure and the Ring library. Set up routes, handle HTTP requests and responses, and render dynamic content.</td><td>CO5</td></tr>
<tr><td>22</td><td colspan="2">Write functions that interact with the database, perform CRUD operations, and handle transactions.</td><td>CO5</td></tr>
<tr><td>23</td><td colspan="2">Implement error handling mechanisms, such as exception handling and error management, in Clojure.</td><td>CO4</td></tr>
<tr><td>24</td><td colspan="2"><b>Mini Project</b>: 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.</td><td>CO5</td></tr>
<tr><td colspan="4">Practicals on Kotlin Programming Language</td></tr>
<tr><td rowspan="2">25</td><td>19</td><td>WAP for print following o/p
Hello Kotlin!!!</td><td rowspan="2">CO2</td></tr>
<tr><td>20</td><td>WAP to take employee’s basic salary, dept_code and experience. Calculate bonus according to following criteria</td></tr>
</table>

	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	CO3

	extension function that counts the number of vowels in a string or reverses the string.	
32	**Mini Project**: Quiz App Build a quiz application that presents users with multiple-choice questions on various topics. Users can select their answers, and the app provides instant feedback on correctness. Keep track of the user's score and display the result at the end of the quiz. Include features like a timer, score calculation, and a database of questions.	CO5

COURSE TEMPLATE

Department:	**Department of Computer Applications**		
Course Name: **DevOps & Automation**	**Course Code**	**L-T-P**	**Credits**
	ENSP411	4-0-0	4
Type of Course:	Department Elective II (Minor)		

Pre-requisite(s), if any: Nil

Brief Syllabus:

Throughout the subject, students will engage in hands-on exercises and projects to gain practical experience with various DevOps tools and practices. By the end of the course, students will be well-equipped to embrace the DevOps culture and apply automation techniques to enhance software development, delivery, and operations processes.

UNIT WISE DETAILS

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

Content Summary:

Overview of DevOps: Definition, objectives, and benefits.

DevOps Principles: Collaboration, automation, continuous integration, continuous delivery, and continuous deployment.

DevOps Tools: Introduction to popular DevOps tools like Git, Jenkins, Docker, Kubernetes, and Ansible.

Version Control with Git: Branching, merging, and collaborative development using Git.

Continuous Integration (CI): Setting up CI pipelines with Jenkins for automated building and testing.

Continuous Delivery and Deployment: Implementing CD pipelines for deploying applications to various environments.

Unit Number: 2	**Title: Containerization and Orchestration**	**No. of hours: 10**
Content Summary: Introduction to Containers: Docker and containerization concepts. Container Management: Working with Docker containers, images, and registries. Docker Compose: Managing multi-container applications. Introduction to Kubernetes: Container orchestration and Kubernetes architecture. Deploying Applications with Kubernetes: Deploying, scaling, and managing applications on Kubernetes.		
Unit Number: 3	**Title: Configuration Management and Infrastructure as Code (IaC)**	**No. of hours: 10**
Content Summary: Introduction to Configuration Management: Need for configuration management tools. Managing Infrastructure with Ansible: Ansible architecture and playbooks for automated configuration management. Infrastructure as Code (IaC) Concepts: Managing infrastructure using code, benefits of IaC. IaC with Terraform: Infrastructure provisioning using Terraform and cloud service providers (e.g., AWS, Azure).		
Unit Number: 4	**Title: Monitoring, Logging, and Security in DevOps**	**No. of hours: 10**
Content Summary: Application Monitoring: Monitoring tools and techniques for tracking application performance and health. Log Management: Centralized log collection, analysis, and visualization. Security in DevOps: Implementing security best practices in CI/CD pipelines and containerized environments.		

DevOps Culture and Collaboration: Encouraging collaboration between development and operations teams.

***SELF-LEARNING COMPONENTS:**

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

Please Note:

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

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

Reference Books:

1. Jez Humble and David Farley, "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation," Pearson Education, Inc., 2011.
2. Nigel Poulton, "The Kubernetes Book," Independently published, 2018.
3. Sam Newman, "Building Microservices: Designing Fine-Grained Systems," O'Reilly Media, Inc., 2015.
4. Eberhard Wolff, "Microservices Patterns: With examples in Java," Manning Publications, 2018.
5. Yevgeniy Brikman, "Terraform: Up & Running: Writing Infrastructure as Code," O'Reilly Media, Inc., 2017.

Define Course Outcomes (CO)

COs	Statements
CO1	**Understand** the principles and benefits of DevOps, and its role in enhancing collaboration and efficiency between development and operations teams.
CO2	**Acquire** hands-on experience with popular DevOps tools such as Git, Jenkins, Docker, Kubernetes, and Ansible for implementing continuous integration, continuous delivery, and automated deployment processes.

CO3	**Demonstrate** proficiency in containerization and orchestration techniques using Docker and Kubernetes for efficient and scalable application deployment and management.
CO4	**Implement** configuration management and Infrastructure as Code (IaC) using Ansible and Terraform to automate the provisioning and management of infrastructure resources.
CO5	**Develop** skills in monitoring, logging, and security practices in the context of DevOps, ensuring application performance, resilience, and adherence to security best practices.

COs Mapping with Levels of Bloom's taxonomy

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

CO-PO Mapping

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

Justification for mapping must be relevant.

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

CO-PSO Mapping

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

Relevance of the Syllabus to various indicators

Unit I	Introduction
Local	Addresses local understanding of the Internet and its impact on society
Regional	Addresses regional internet connectivity and network infrastructure requirements
National	Contributes to national digital literacy and internet connectivity strategies
Global	Aligns with global trends in internet technologies and network protocols

Employability	Develops skills in using internet-based services and understanding network protocols
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in internet technologies and network protocols
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
Unit II	
Local	Addresses local understanding of the Internet and its impact on society
Regional	-
National	Contributes to national digital literacy and internet connectivity strategies
Global	Aligns with global trends in internet technologies and network protocols
Employability	Develops skills in using internet-based services and understanding network protocols
Entrepreneurship	-
Skill Development	Develops basic knowledge and skills in internet technologies and network protocols
Professional Ethics	-
Gender	-
Human Values	-

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

Employability	Develops skills in internet telephony, multimedia applications, and SEO
Entrepreneurship	-
Skill Development	Develops knowledge and skills in internet telephony, multimedia applications, and SEO
Professional Ethics	-
Gender	-
Human Values	-
Environment & Sustainability	-
SDG	SDG 4
NEP 2020	-
POE/4th IR	Aligns with the concepts of internet telephony, multimedia applications, and SEO

DEVOPS & AUTOMATION LAB

Department:	**Department of Computer Applications**		
Course Name: **DevOps & Automation Lab**	**Course Code**	**L-T-P**	**Credits**
	ENSP461	0-0-2	1
Type of Course:	Minor (Department Elective IV)		
Pre-requisite(s), if any:			

Proposed Lab Experiments

Defined Course Outcomes

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

CO 5	**Understand** and **apply** monitoring, logging, and security practices in DevOps, ensuring application performance, resilience, and adherence to security best practices throughout the software development lifecycle.

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

16	Implementing feature flags for controlled feature rollout	CO5
17	Load testing and performance optimization	CO5
18	Automating application tests with Selenium	CO2, CO5
19	Integrating automated testing in CI/CD pipelines	CO2, CO5
20	Blue-green deployment for zero-downtime updates	CO3, CO5
21	Canary deployment for testing new features	CO3, CO5
22	Implementing GitOps for application deployments	CO3, CO5
23	Managing secrets and sensitive data securely	CO5
24	Disaster recovery planning and testing	CO5
25	Creating a DevOps project integrating multiple tools	CO1, CO2, CO3, CO4, CO5

1. **Setting up version control with Git**: Exercise: Initialize a Git repository, create branches, perform commits, and push changes to a remote repository. Project: Collaboratively work on a project using branching and merging techniques in Git.

2. **Implementing a basic Jenkins CI/CD pipeline**: Exercise: Set up a simple Jenkins pipeline to build and test a sample application from version control. Project: Develop a complete CI/CD pipeline that includes code building, automated testing, and deployment to a staging environment.

3. **Automating application deployment with Jenkins**: Exercise: Configure Jenkins to automatically deploy the application to a test server upon successful build. Project: Implement a full-fledged CD pipeline with Jenkins, including deployment to production after successful testing.

4. **Containerizing an application using Docker**: Exercise: Dockerize a basic application and run it in a container. Project: Containerize a multi-service application with Docker Compose for easier deployment.
5. **Managing Docker containers and images**: Exercise: Explore Docker commands to manage containers and images, such as starting, stopping, and cleaning up. Project: Implement a container registry and manage images for different application versions.
6. **Deploying applications with Kubernetes**: Exercise: Set up a Kubernetes cluster and deploy a basic application using YAML manifests. Project: Deploy a microservices-based application with Kubernetes, configuring services and network policies.
7. **Implementing Kubernetes deployment strategies**: Exercise: Implement rolling updates and rollbacks in Kubernetes. Project: Use Kubernetes deployment strategies like blue-green and canary deployments for a real-world application.
8. **Continuous deployment with Kubernetes**: Exercise: Set up a Jenkins pipeline for continuous deployment to Kubernetes. Project: Create an end-to-end automated CD pipeline with Jenkins and Kubernetes.
9. **Configuring infrastructure with Ansible**: Exercise: Use Ansible to provision and configure virtual machines. Project: Create a playbook to configure a complete development environment for an application.
10. **Automating application configuration with Ansible**: Exercise: Create Ansible playbooks to automate application-specific configurations. Project: Implement dynamic inventory and use Ansible roles for better code organization.
11. **Implementing Infrastructure as Code (IaC) with Terraform**: Exercise: Set up a basic Terraform configuration to create cloud resources. Project: Use Terraform to define infrastructure for a scalable and fault-tolerant application.
12. **Creating scalable and resilient infrastructure with Terraform**: Exercise: Implement auto-scaling and load balancing in Terraform. Project: Design a Terraform template for a highly available architecture using multiple availability zones.
13. **Monitoring application performance with Prometheus**: Exercise: Set up Prometheus for monitoring application metrics. Project: Create custom Prometheus metrics and use Grafana for visualization and alerting.

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

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

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

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

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

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

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

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

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

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

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

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