SYLLABUS

(With effect from 2025 -26)

ಪಠ್ಯಕ್ರಮ

(ಶೈಕ್ಷಣಿಕ ವರ್ಷ 2025-26)

Bachelor Degree

In

Information Science & Engineering

III & IV Semester

Out Come Based Education

With

Choice Based Credit System

P.E.S. College of Engineering, Mandya - 571 401, Karnataka

[An Autonomous Institution affiliated to VTU, Belagavi,
Grant – in – Aid Institution (Government of Karnataka),
Accredited by NBA (All UG Programs), NAAC and Approved by AICTE, New Delhi]

ಪಿ.ಇ.ಎಸ್. ತಾಂತ್ರಿಕ ಮಹಾವಿದ್ಯಾಲಯ

ಮಂಡ್ಯ–571 401, ಕರ್ನಾಟಕ

(ವಿ.ಟಿ.ಯು, ಬೆಳಗಾವಿ ಅಡಿಯಲ್ಲಿನ ಸ್ವಾಯತ್ತ ಸಂಸ್ಥೆ)

Ph : 08232- 220043, Fax : 08232 ï 222075,Web : www.pescemandya.org

VISION

"PESCE shall be a leading institution imparting quality Engineering and Management education developing creative and socially responsible professionals."

MISSION

- *Provide state of the art infrastructure, motivate the faculty to be proficient in their field of specialization and adopt best teaching-learning practices.*
- *Impart engineering and managerial skills through competent and committed faculty using outcome based educational curriculum.*
- *Inculcate professional ethics, leadership qualities and entrepreneurial skills to meet the societal needs.*
- *Promote research, product development and industry-institution interaction.*

QUALITY POLICY

Highly committed in providing quality, concurrent technical education and continuously striving to meet expectations of stake holders.

CORE VALUES

***P**rofessionalism*
***E**mpathy*
***S**ynergy*
***C**ommitment*
***E**thics*

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

About the Department

The Department of Information science and Engineering takes pride in producing quality engineers over the past 19 years. The credit for all the flowery results goes to the highly motivating staff, from whom all students draw inspiration. The Department was started in the year 2000. The present intake of the undergraduate program is 60. The department has well equipped classrooms, computer laboratories with high-end systems, department library. We are proud to produce the first PhD student in our college. Faculty members of the department are involved in research activities in different fields such as Medical Image Processing, Pattern Recognition, and Data Mining etc. The department is using Outcome-based education (OBE), which is a recurring education reform model, and it is affiliated to Visvesvaraya Technological University (VTU). The department has achieved good Placement, conducted International Conferences and other sponsored short-term courses, workshops, National seminars and symposia. The laboratory facilities and the Internet access are available to the staff and students of the Information Science and Engineering

Vision

"The department strives to equip our graduates with Knowledge and Skills to contribute significantly to Information Science & Engineering and enhance quality research for the benefit of societyò.

Mission

M1: To provide students with state of art facilities and tools of Information Science & Engineering to become productive, global citizens and life-long learners.

M2: To prepare students for careers in IT industry, Higher education and Research.

M3: To inculcate leadership qualities among students to make them competent Information Science & Engineering professionals or entrepreneurs.

1.2. State the Program Educational Objectives (PEOs)

Graduates of the program will be able to

PEO1: Establish a productive Information Science & Engineering career in industry, government or academia.

PEO2: Interact with their peers in other disciplines by exhibiting professionalism and team work to contribute to the economic growth of the country.

PEO3: Promote the development of solutions to the problems in Information Science using hardware and software integration.

PEO4: Pursue higher studies in Engineering, Management or Research.

Knowledge and Attitude Profile(WK)

WK1: A systematic, theory-based understanding of the natural sciences applicable to the discipline and awareness of relevant social sciences.

WK2: Conceptually-based mathematics, numerical analysis, data analysis, statistics and formal aspects of computer and information science to support detailed analysis and modelling applicable to the discipline.

WK3: A systematic, theory-based formulation of engineering fundamentals required in the engineering discipline.

WK4: Engineering specialist knowledge that provides theoretical frameworks and bodies of knowledge for the accepted practice areas in the engineering discipline much is at the forefront of the discipline**.**

WK5: Knowledge, including efficient resource use, environment a impacts, whole-life cost, re- use of resources, net zero carbon, and similar concepts, that supports engineering design and operations in a practice area**.**

WK6: Knowledge of engineering practice(technology) in the practice areas in the engineering discipline.

WK7: Knowledge of the role of engineering in society and identified issues in engineering practice in the discipline, such as the professional responsibility of an engineer to public safety and sustainable development.

WK8: Engagementwithselectedknowledgeinthecurrentresearchliteratureofthediscipline, awareness of the power of critical thinking and creative approaches to evaluate emerging issues.

WK9:Ethics, inclusive behaviour and conduct. Knowledge of professional ethics, responsibilities,andnormsofengineeringpractice.Awarenessoftheneedfordiversity by reason of ethnicity, gender, age, physical ability etc. with mutual understanding and respect, and of inclusive attitudes.

A. List of Program Outcomes (PO's)

Engineering Graduates will be able to:

PO1: Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified in WK1 to WK4 respectively to develop to the solution of complex engineering problems.

PO2: Problem Analysis: Identify, formulate, review research literature and analyze complex engineering problems reaching substantiated conclusions with consideration for sustainable development. (WK1 to WK4).

PO3: Design/Development of Solutions: Design creative solutions for complex engineering problems and design/develop systems/components/processes to meet identified needs with consideration for the

public health and safety, whole-life cost, net zero carbon, culture, society and environment as required. (WK5)

PO4: Conduct Investigations of Complex Problems: Conduct investigations of complex engineering problems using research-based knowledge including design of experiments, modelling, analysis & interpretation of data to provide valid conclusions. (WK8).

PO5: Engineering Tool Usage: Create, select and apply appropriate techniques, resources and modern engineering & IT tools, including prediction and modelling recognizing their limitations to solve complex engineering problems. (WK2 and WK6)

PO6:The Engineer and The World: Analyze and evaluate societal and environmental aspects while solving complex engineering problems for its impact on sustainability with reference to economy, health, safety, legal framework, culture and environment. (WK1, WK5, and WK7).

PO7: Ethics: Apply ethical principles and commit to professional ethics, human values, diversity and inclusion; adhere to national & international laws. (WK9)

PO8: Individual and Collaborative Teamwork: Function effectively as an individual, and as a member or leader in diverse/multi-disciplinary teams.

PO9:Communication: Communicate effectively and inclusively within the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations considering cultural, language, and learning differences

PO10:Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to oneôs own work, as a member and leader in a team, and to manage projects and in multi disciplinary environments.

PO11:Life-Long Learning: Recognize the need for, and have the preparation and ability for

i) independent and life-long learning
ii) adapt ability to new and emerging technologies and
iii) critical thinking in the broadest context of technological change. (WK8)

B. List of Program Specific Outcomes (PSOs)

Information Science & Engineering Graduates will have

PSO1- The Knowledge to excel in IT profession by utilizing mathematical concepts, programming paradigms and software development practices for successful career.

PSO2- The ability to continuously learn and develop solutions in IT world by applying the emerging technologies in multidisciplinary environment

B.E-P24 SCHEME- THIRD SEMESTER CREDITS & COMPONENTS												
Sl. No.	**Course Code**	**Course Title**	**Teaching Department**	**Hrs / Week**			**Credits**	**Examination Marks and Duration**				
				L	**T**	**P**		Max. Marks CIE	Duration CIE (hours)	SEE	Duration SEE (hours)	Total Marks CIE +SEE
1.	P24MA301C	Statistics and Probability	Mat	2	2	-	3	50	1.5	50	3	100
2.	P24IS302	Data Structures	IS	3	-	-	3	50	1.5	50	3	100
3.	P24IS303	Computer Organization	IS	3	-	-	3	50	1.5	50	3	100
4.	P24IS304	Foundations of Information Science	IS	2	2	-	3	50	1.5	50	3	100
5.	P24IS305	Object Oriented Programming With JAVA	IS	3	-	-	3	50	1.5	50	3	100
6.	P24IS306	Digital Systems Design	IS	3	-	-	3	50	1.5	50	3	100
7.	P24ISL307	Data Structure Laboratory	IS	-	-	2	1	50	2	50	3	100
8	P24ISL308	Object Oriented Programming with JAVA Laboratory	IS	-	-	2	1	50	2	50	3	100
9	P24ISL309	Digital Systems Design Laboratory	IS	-	-	2	1	50	2	50	3	100
10.	P24HSMC310A	Employability enhancement Skills - III	IS	1	-	-	1	50	1 (MCQ)	50	2 (MCQ)	100
11.	P24NSS311	National Service Scheme	IS	-	-	-	-	50	1 (MCQ)	50	2 (MCQ)	PP/NP
	P24YOG311	Yoga										
	P24PED311	Physical Education										
12.		AICTE Activity Points	(students have to earn 100 activity points between 01 to 08 semester)					Compulsory requirement for the award of a degree				
Total							**22**	**550**		**550**		**1100**
		BRIDGE COURSE	**B.E [Lateral Entry Students]**									
13.	P24MADIP301	Basic Engineering Mathematics – I		2	2	-	-	50	1.5	50	3	100
14.	P24HDIP308	Additional Communicative English - I			2	-	-	50	1.5	50	3	100

		B.E-P24 SCHEME- FOURTH SEMESTER CREDITS & COMPONENTS										
Sl. No.	**Course Code**	**Course Title**	**Teaching Department**	**Hrs / Week**			**Credits**	**Examination Marks and Duration**				
				L	**T**	**P**		Max. Marks CIE	Duration CIE (hours)	SEE	Duration SEE (hours)	Total Marks CIE +SEE
1.	P24MA401C	Linear Algebra	Mat	2	2	-	3	50	1.5	50	3	100
2.	P24IS402	Theory of Computation	IS	3	-	-	3	50	1.5	50	3	100
3.	P24IS403	Design and Analysis of Algorithm	IS	3	-	-	3	50	1.5	50	3	100
4.	P24IS404	Software Engineering	IS	3	-	-	3	50	1.5	50	3	100
5.	P24IS405	Database Management System	IS	3	-	-	3	50	1.5	50	3	100
6.	P24IS406	Operating System	IS	3	-	-	3	50	1.5	50	3	100
7.	P24ISL407	Design and Analysis of Algorithms Laboratory	IS	-	-	2	1	50	2	50	3	100
8.	P24ISL408	Database Management System Laboratory	IS	-	-	2	1	50	2	50	3	100
9.	P24ISL409	Operating System Laboratory	IS	-	-	2	1	50	2	50	3	100
10.	P24HSMC410A	Employability enhancement Skills - IV	IS	1	-	-	1	50	1 (MCQ)	50	2 (MCQ)	100
11.	P24NSS411	National Service Scheme	IS	-	-	-	-	50	1 (MCQ)	50	2 (MCQ)	PP/NP
	P24YOG411	Yoga										
	P24PED411	Physical Education										
12.		AICTE Activity Points	(students have to earn 100 activity points between 01 to 08 semester)					Compulsory requirement for the award of a degree				
TOTAL							**22**	**550**		**550**		**1100**
BRIDGE COURSE			**B.E [Lateral Entry Students]**									
13.	P24MADIP401	Basic Engineering Mathematics ï II		2	2	-	-	50	1.5	50	3	100
14.	P24HDIP408	Additional Communicative English - II			2	-	-	50	1.5	50	3	100

Academic Year: **2025-26**	Semester: **III**	Scheme: **P24**
Course Title: **Statistics and Probability (Common to CSE Streams)**		
Course Code: **P24MA301C**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P): **2:2:0**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy: **40 Hours**	Exam Hours: **3 Hrs**	
Credits: **03**		

Course Learning Objectives:

1	**Understand** the basic concepts of Statistics and Probability.
2	Categorize and **analyse** the given data using statistical tools.
3	Identify and **apply** the appropriate statistical method to solve given problems

Unit	Syllabus content	No. of hours	
		Theory	**Tutorial**
I	**Introduction to Statistics:** Introduction, frequency distributions, Measure of central tendency-mean, median and mode - for grouped and ungrouped data, illustrative examples. Measure of dispersion-quartile and mean deviation - for grouped and ungrouped data. Moments, method of moments. Fitting of the curves $y=ax+b$, $y=ax^2+bx+c$, $y=ab^x$, $y=ax^b$ by using the method of least squares. **Self – Study content:** Coefficients of Dispersion. Linear regression-angle between two lines of regression.	06	02
II	**Probability distribution**: Introduction to probability, Random Variables, Distribution function, Probability mass function and Probability density function. Discrete Probability Distributions-Introduction and Motivation, Binomial and Poissonôs distribution. Continuous Probability Distributions-Exponential and Normal Distribution. **Self-study**: Geometric distribution and their properties**.**	06	02
III	**Joint Probability and Markov chain:** Joint probability distribution - for two discrete random variables, expectation, covariance and correlation. Markov Chain: Introduction to Stochastic Process, Probability vector, Stochastic matrix, regular stochastic matrices, Markov chains, higher transition probabilities, Stationary distribution of Regular Markov chains and absorbing states. **Self-study component** ï Joint Probability distribution for two continuous random variables.	06	02
IV	**Sampling theory:** Sampling Theory ï Introduction, Random sampling. Sampling from finite and infinite populations, Sampling distributions, Statistical hypotheses, Null Hypotheses, Tests of hypotheses and significance, Type-I and Type-II errors, level of significance, one tailed and two tailed tests, tests of significance for large and small samples- Students 't' test and Chi-square test. **Self-study**: **Self-Study Content:** F-test, Fisherôs z-distribution.	06	02

V	**Statistical Modelling:** Basics of Time series analysis-semi average and moving average methods. Correlation and regression, Karl Pearson's coefficient, lines of regression, multiple regression, non-linear correlation. Introduction to R, Functions, Control flow and Loops, working with vectors and matrices, reading of data, writing data, working with data, manipulating data, simulation. **Self-study**: Multiple Correlation and Regression.	06	02

COURSE OUTCOMES: On completion of the course, student should be able to:

CO1: Understand the basic principles of statistics and probability.

CO2: Analyze the given data using statistical techniques.

CO3: Apply various statistical tests for solving the given problem.

CO4: Understand the basic concepts of R – programming to solve statistical problems.

TEXTBOOKS

1. V. K. Kapoor and S. C. Gupta, Fundamentals of Mathematical Statistics, 2020 & 12th Edition, Sultan Chand & Sons, New Delhi.
2. Kapur J. N. and Saxena H. C., Mathematical Statistics, 2010 & 2nd Edition, Sultan Chand & Sons, New Delhi.

REFERENCE BOOKS

1. E. Kreysizig, Advanced Engineering Mathematics, John Wiley and sons, 10th Ed. (Reprint) 2016.
2. R. Miller, J. E. Freund and R. Johnson, Probability and Statistics for Engineers, 2017 & 9th Edition, PHI, New Delhi.
3. A. Goon, M. Gupta and B. Dasgupta, Fundamentals of Statistics, World Press.

Active Based Learning (Suggested Activity in Class)/ Practical Based Learning (Example)
1. Flip Class 2. Seminar/ poster Presentation 3. Individual Role play/Team Demonstration/ Collaborative Activity 4. Case study 5. Learn by Doing

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2			2							
CO2	2	3			2							
CO3	3	2			1							
CO4	2	3			1							
Strength of correlation: Low-1, Medium-2, High-3												

Academic Year: 2025-26	**Semester: III**	**Scheme: P24**
Course Title: **Data Structures**		
Course Code: **P24IS302**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P): **3:0:0**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy:**40**	Exam Hours: **3 Hrs**	
Credits:0**3**		

Prerequisite: Basics of C programming

Course learning Objectives:

CLO1: To become familiar with the concept of pointers and its usage in data structure.
CLO2: To study and understand the representation and implementation of linear & non-linear data structures.
CLO3: To identify the appropriate data structure while solving real-time applications.

Unit 1	**8 Hours**

Basic Concepts: System Life Cycle, Algorithm Specification: Introduction, Performance Analysis
Pointers: Review of Pointers, Pointers and arrays, Arrays of Pointers.
Structures: Arrays of Structures, Structures and Functions-Passing Individual Members, Passing the Entire Structure, Passing Structures through Pointers, Self-referential Structures.
Introduction: Basic Terminology-Elementary Data Structure Organization, Classification of Data Structures.

Self-Study Content: Pointers and Two-dimensional Arrays, Operations on Data Structures

Text book Map: Text Book 1: Chapter 1: 1.1, 1.2, 1.4
Text Book 2: Chapter 1: 1.11; **Chapter 2:** 2.1, 2.2; **Chapter 3:** 3.7, 3.8;
Chapter 5: 5.3, 5.4, 5.5

Unit 2	**8 Hours**

Stacks: Introduction to Stacks, Operations on Stack, Applications of Stacks: Implementing Parentheses Checker, Conversion of Expression: infix to postfix, Evaluation of Expressions: prefix expression, postfix expression.

Self-Study Content Conversion of Expressions: infix to prefix, Prefix to postfix, prefix to infix, Postfix to infix.

Text book Map: Text Book 1: Chapter 3
Text Book 2: Chapter 7

Unit 3	**8 Hours**

Recursion: Introduction, Factorial of a number, Fibonacci series, Tower of Hanoi, GCD of two numbers.
Queues: Introduction to Queues, Operations on Queue.
Types of Queues: Circular Queues, Deques, Priority Queues, Multiple Queues.

Self-Study Content: Types of recursion with examples (Linear Search, Binary Search)
Applications of Queues: Josephus Problem.

Text book Map: Text Book 2: Chapter 7, Chapter 8

Unit 4	8 Hours
Linked Lists: Dynamic memory Allocation, Introduction, Operations on lists, Singly linked lists, Circular linked lists, Doubly Linked lists, Applications of linked lists-Polynomial Representation, Evaluation of polynomials	
Self-Study Content: Doubly circular linked lists, Header linked list	
Text book Map: Text Book 1: Chapter 4 **Text Book 2: Chapter 6** **Appendix A**	
Unit 5	**8 Hours**
Trees: Introduction, Basic Terminology, Types of Trees, Traversing a Binary Tree, Huffmanôs tree. Applications of Trees, Binary Search Trees, Operations on Binary Search Trees, Threaded Binary Trees.	
Self-Study Content: Expression Trees	
Text book Map: Text Book 1: Chapter 5 **Text Book 2: Chapter 9, 10**	
Teaching Learning Process: Chalk and Board, PPT	

Course Outcomes: At the end of the course students should be able to:	
CO1	**Apply** the concepts of pointers in data structures.
CO2	**Analyze** and represent various data structures and its operations.
CO3	**Design** algorithms using different data structures like List, Stack, Queue and Trees.
CO4	**Develop** programs with suitable data structure based on the requirements of the real-time Applications.

Suggested Learning Resources:				
Textbooks:				
1	Fundamentals of Data Structures in C	E. Horowitz and S. Sahani, Anderson-Freed	2nd Edition 2011	University Press
2	Data Structures using C	Reema Thareja	3rd Edition 2023	Oxford University Press
Reference Books:				
1.	Data Structures using C	Aaron M Tenenbaum , Yedidyah Langsam and Moshe J Augenstein	2014	Low Price Edition, Pearson Education
2	Data Structures with C (Schaum's Outline Series)	Seymour Lipschutz	July 2017	McGraw Hill Education

Web links and Video Lectures (e-resources)
1. https://nptel.ac.in/courses/106102064/
Active Based Learning (Suggested Activity in Class)/Practical Based Learning
1. **https://www.academia.edu/28758384/**
Active Based Learning (Suggested Activity in Class)/Practical Based Learning
1.Flip Class **2**.Individual Role Play/Team Demonstration /Collaborative Activity **3**.Case Study **4**.Learn by Doing

CO-PO Mapping:

CO	Statement	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO1	PSO 2
CO1	**Apply** the concepts of pointers in data structures.	3		2									2	
CO2	**Analyze** and represent various data structures and its operations.	3	3	2	1	1							2	
CO3	**Design** algorithms using different data structures like List, Stack, Queue and Trees.	3	3	3	1	1			1	2		2	2	
CO4	**Develop** programs with suitable data structure based on the requirements of the real-time applications.	3	3	3	1	1		2	2	2	1	2	2	

Academic Year:**2025-26**	Semester:**III**	**Scheme:P24**
Course Title: **Computer Organization**		Credits:0**3**
Course Code:**P24IS303**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week(L:T:P):**3:0:0**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours o Pedagogy:**40**	ExamHours:**3 Hrs**	

Course Learning Objectives:

CLO1:Conceptualize the basics of Organizational issues of a digital computer and compare the performance of machine instruction.

CLO2:Expose different ways of communication with I/O Devices.

CLO3: Notice how to perform computer arithmetic operation.

CLO4:Understand working of processing unit using different bus structures.

CLO5: Illustrate different Types of memory devices with their principles.

UNIT – I	**8 Hours**

BASIC STRUCTURE OF COMPUTERS: Basic operational Concepts, Performance.
INSTRUCTION SET ARCHITECTURE: Memory Location and Addresses, Memory Operations, Instruction and Instruction Sequencing, Addressing Modes, Assembly Language.

Self-study content: Functional Units of Computer, Number Representation and Arithmetic Operations, Character representation.

Text book Mapping: Text Book 1: Chapter1-1.3,1.6, Chapter 2 ï 2.1-2.5

UNIT – II	**8 Hours**

INSTRUCTION SET ARCHITECTURE (Continued): Subroutines, Additional instructions.

BASIC INPUT/OUTPUT: Accessing I/O Devices- I/O Device Interface, Program Controlled I/O,An Example of a RISC-Style I/O program, Interrupts-Enabling and Disabling Interrupts, Handling Multiple Devices, Exceptions.

INPUT/OUTPUT ORGANIZATION: Bus Structure, Bus Operation -Synchronous Bus, Asynchronous Bus.

Self-study content: Stacks, Interface Circuits.

Text book Mapping: Text Book 1: Chapter 2 ï 2.7,2.8, Chapter 3 ï 3.1,3.2, Chapter 7 ï 7.1,7.2

UNIT – III	**8 Hours**

BASIC PROCESSING UNIT: Some Fundamental Concepts, Instruction Execution, Hardware Components, Instruction Fetch and Execution Steps, Control Signals, Hardwired Control.

Self-study content: CISC Style Processors.

Text book Mapping: Text Book 1: Chapter 5 ï 5.1,5.6

UNIT – IV	**8 Hours**

ARITHMETIC: Multiplication of Signed Numbers, Fast Multiplication-Bit Pair Recoding of Multipliers, Carry-Save Addition of Summands, Integer Division, Introduction to Floating point Numbers and Operations.

Self-study content: Design of Fast Adders, Multiplication of Unsigned numbers.	
Text book Mapping: Chapter 9ï 9.4-9.7	
UNIT – V	**8 Hours**
MEMORY SYSTEM: Basic Concepts, Semiconductor RAM Memories, Memory Hierarchy, and Cache Memories ï Mapping Functions.	
Self-study content: Read Only Memories, Direct Memory Access	
Text book Mapping: Chapter 8ï 8.1,8.2,8.5,8.6	
Course Outcomes: At the end of the course students should be able to:	
CO1	**Understand** the operation and organization of a digital computer system.
CO2	**Apply** the knowledge of assembly language / algorithmic techniques to solve the given problem.
CO3	**Analyze** the given assembly language code snippet.
CO4	**Design** Memory Modules.

Suggested Learning Resources:				
Textbooks:				
1	Computer Organization and Embedded Systems.	Carl Hamacher ZvonkoVranesic, SafwatZaky	6th Edition	Tata Mc GrawHill.
Reference Books:				
1.	Computer Organization & Architecture.	William Stallings	9th Edition	PHI,2013
2	Computer Systems Design and Architecture.	Vincent P.Heuring & Harry F. Jordan	2nd Ed.Pearson Education	2004

Web and Video link(s) Lectures (e-resources):

1. https://nptel.ac.in/courses/106/103/106103068/
2. https://nptel.ac.in/content/storage2/courses/106103068/pdf/coa.pdf
3. https://nptel.ac.in/courses/106/105/106105163/
4. https://nptel.ac.in/courses/106/106/106106092/
5. https://nptel.ac.in/courses/106/106/106106166/
6. http://www.nptelvideos.in/2012/11/computer-organization.html

CO-PO Mapping

CO	Statement	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO1	PSO 2
CO1	**Understand** the operation and organization of a digital computer system.	2							1				2	
CO2	**Apply** the knowledge of assembly language / algorithmic techniques to solve the given problem.	2	2	1	2	2		1	1	1		1	2	
CO3	**Analyze** the given assembly language code snippet.	2	2	1	2	2		1	1	1		1	1	
CO4	**Design** memory modules.	2	2	2					1				1	

Academic Year:**2025-26**	Semester:**III**	Scheme:**P24**
Course Title: **Foundations of Information Science**		
Course Code: **P24IS304**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching Hours/Week(L:T:P):**2:2:0**	SEEMarks:**50**	SEE Weightage:**50%**
Total Number of Teaching Hours: **40**	ExamHours:**3 Hrs**	Credits:**3**

Course Learning Objectives:

CLO1: Understand the foundational concepts, philosophies, and historical development of Information Science.

CLO2: Analyze information behaviors, organization, and retrieval systems within various domains.

CLO3: Evaluate the impact of digital technologies, ethical considerations, and future trends in the field of Information Science.

Module-Wise Topics	
Module 1:	**8 Hours**

Introduction to Information Science: The nature and scope of Information Science, Historical evolution: From documents to digital, Philosophical underpinnings and paradigms.

Self-study component:	Information use and users.

Text book Chapters: 1, 2, 3, 4

Module 2:	**8 Hours**

Information Behavior and Domain Analysis: Data, information, knowledge, and wisdom, Understanding information needs and behaviors, Models of information- seeking behavior, Domain analysis and its significance, Information behavior in digital environments.

Self-study component:	Pick a domain (e.g., Engineering, Music) and describe how professionals in that field find information.

Text book Chapters: 5,7,12

Module 3:	**8 Hours**

Organizing and Retrieving Information: Principles of information organization, Metadata standards and resource description, Classification systems and taxonomies, Information retrieval systems and infometrics.

Self-study component:	Applications of infomatrics.

Text book Chapters: 6, 8, 11

Module 4:	**8 Hours**

Digital Technologies and Information Systems: Digital technologies in information science, Data systems and digital infrastructures, Design and evaluation of information systems, Emerging technologies and their applications.

Self-study component:	Applications of Information Systems.

Text book Chapters: 9, 10, 13

Module 5:	**8 Hours**

Ethics, Policy, and Future Directions: Information management and organizational policies, Legal and ethical issues in information science, The role of information in society, Digital literacies and future trends.

Self-study component:	The future of information science

Text book Chapters: 14, 15, 16, 17, 19

COs	**Course Outcomes** with ***Action verb*** for the Course topics
CO1	Describe the foundational theories and historical development of Information Science.
CO2	Identify user information needs and behaviors across different domains.
CO3	Apply principles of information organization and retrieval using appropriate tools and standards.
CO4	Analyze ethical, legal, and technological challenges in information systems

Text Book(s):

- Bawden, David, and Lyn Robinson. Introduction to Information Science.2nd Edition, Facet Publishing, 2022.

Reference Book(s):

1. Saracevic, Tefko (Ed.). Information Science: Integration in a Cyber Age. Information Today, Inc.
2. Rainer, R. Kelly, and Brad Prince. Introduction to Information Systems, 10th Edition, Wiley, 2023.
3. Gregory, Vicki L. Collection Development and Management for 21st Century Library Collections. ALA Editions, 2019.
4. Witten, Ian, David Bainbridge & David Nichols. How to Build a Digital Library. Morgan Kaufmann, 2009.

CO-PO Mapping

CO	Statement	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2
CO1	**Describe** the foundational theories and historical development of Information Science.	3	2				1	1				2		1
CO2	**Identify** user information needs and behaviors across different domains.	3	3	2								2		2
CO3	**Apply** principles of information organization and retrieval using appropriate tools and standards.	3	2	3								2		2
CO4	**Analyze** ethical, legal, and technological challenges in information systems	2	2				2	3				2		2

AcademicYear:2025-26	**Semester:** III	**Scheme:**P24
Course Title: Object Oriented Programming with JAVA	**Course Type:** Professional Core Course	
CourseCode:P24IS305	**CIE Marks:**50	**CIE Weightage:**50
Teaching hours/week(L:T:P) 3:0:0	**SEE Marks:**50	**SEE Weightage:**50
Teaching hours of Pedagogy:40	**Exam Hours:**03	
Credits:03		

Course learning Objectives:

CLO1: Explain the syntax, data types, operators, and control structures of Java.
CLO2: Develop Java programs using classes and objects to represent real-world entities, incorporating inheritance, polymorphism, abstraction, and encapsulation.
CLO3: Design multithreaded Java applications using thread life cycle control and synchronization mechanisms to manage concurrency.
CLO4: Construct type-safe code using generics and boxing.
CLO5: Implement robust runtime error management in Java by utilizing exception handling.
CLO6: Examine Java code to detect and correct errors.

Unit1	**8Hours**

Creating Java Programs: Comparing Procedural and Object-Oriented Programming Concepts, Features of the Java Programming Language, Understanding the First Class, Understanding the main() Method, Saving a Java Class, Adding Comments to a Java Class.

Using Data: Declaring and Using Constants and Variables, Using the Boolean Data Type, Learning About Floating-Point Data Types, Using the char Data Type, Using the Scanner Class to Accept Keyboard Input. Making Decisions and looping: The if and ifé else Statements, Nesting if and ifé else Statements, Using Logical AND and OR operators, Using the switch Statement, Creating while Loops, Creating a for Loop.

Arrays: Declaring an Array, Initializing an Array, Using Variable Subscripts with an Array, Passing Arrays to and Returning Arrays from Methods.

Self-Study Content: Understanding Type Conversion, Nested loops.

Text book mapping

Text book1: Chapter1, Chapter 2, Chapter 5,Chapter6, Chapter 8.

Unit2	**8Hours**

Introducing Classes, Objects, and Methods: Class Fundamental, How Objects are Created, Reference Variables and Assignment, Methods, Returning from a Method, Returning a Value, Using Parameters Constructors, Parameterized Constructors, The new Operator Revisited, The this Keyword. Controlling Access to Class Members, Pass Objects to Methods, How Arguments are Passed, Method Overloading, Overloading Constructors, Understanding static, Var args: Variable-Length Arguments.

Inheritance: Inheritance Basics, Member Access and Inheritance, Constructors and Inheritance, Using super to Call Super class Constructors, Using super to Access Super class Members, Creating a Multilevel Hierarchy, When are Constructors Executed?, Super class References and Subclass, Objects, Method Overriding, Using Abstract Classes.

Self-Study Content: Nested and Inner Classes, Garbage Collection and Finalizers, Using final, final Prevents Overriding, final Prevents Inheritance, Using final with Data Members.	
Text book mapping Text book2: Chapter 4, Chapter 6,Chapter7.	
Unit3	**8Hours**
Interface Fundamentals: Creating an Interface, Implementing an Interface, Using Interface References, Implementing Multiple Interfaces, Constants in Interfaces, Interfaces can be Extended. **Packages:** Package Fundamentals, Packages and Member Access, Importing Packages. **Multithreaded Programming:** Multithreading Fundamentals, the Thread Class and Runnable Interface, Creating a Thread, Creating Multiple Threads, Determining When a Thread Ends, Thread Priorities.	
Self-Study Content: Nested Interfaces, Synchronization, Using Synchronized Methods, The Synchronized Statement.	
Text book mapping Text book2: Chapter 8, Chapter 9, Chapter12.	
Unit4	**8Hours**
Character and Strings: Understanding String Data Problems, Using Character Class Methods, Declaring and Comparing String Objects, Using a Variety of String Methods. **Enumerations, Autoboxing, and Annotations:** Enumerations, Java Enumerations are Class Types, The values ( ) and valueOf( ) Methods, Constructors, Methods, Instance Variables and Enumerations, Enumerations Inherit Enum, Autoboxing, Annotations (Metadata). **Exception Handling:** Learning About Exceptions, Trying Code and Catching Exceptions, Throwing and Catching Multiple Exceptions, Using the finally Block, Understanding the Advantages of Exception Handling, Specifying the Exceptions that a Method Can Throw.	
Self-Study Content: Tracing Exceptions Through the Call Stack, Creating Your Own Exception Classes.	
Text book mapping Text book1: Chapter7, Chapter12. **Text book2:**Chapter13.	
Unit5	**8Hours**
Generics: Generics Fundamentals, A Simple Generics Example, Generics Work Only with Objects, Generic Types Differ Based on Their Type Arguments, A Generic Class with Two Type Parameters, The General Form of a Generic Class, Bounded Types, Using Wildcard Arguments, Bounded Wildcards, Generic Methods. **Applets:** Applet Basics, A Complete Applet Skeleton, Applet Initialization and Termination, A Key Aspect of an Appletõs Architecture, Requesting Repainting, Using the Status Window, Passing Parameters to Applets.	
Self-Study Content: Generic Constructors, Generic Class Hierarchies	
Text book mapping: Text book 2: Chapter 14, Chapter 15.	
Course Out comes with Action verbs for the Course topics	
CO1: Understand the fundamentals of Java programming concepts.	

CO2: Apply object-oriented programming concepts to solve real world problems.
CO3: Develop Java applications using multithreading, generics and exceptions.
CO4: Analyze code snippets or programming scenarios to identify issues and propose solutions.

Suggested Learning Resources:

Textbooks:

Sl.No.	Title	Author	Year& Edition	Publisher
1	Java Programming	Joyce farell	9th Edition,2018 ISBN:978-1-337-39707-0	Cengage
2	Java Programmingï A comprehensive Introduction.	Herbert Schildt and Dale Skrien	1st Edition,2013 ISBN:978-0-07-802207-4	McGrawHill

Reference Books:

1	The Complete Reference-Java	Herbert Schildt	12th Edition, 2022	McGraw Hill Education
2	Learning Java: An Introduction to Real-World Programming with Java.	MarcLoy, Patrick Niemeyer, Daniel Leuck	6th Edition,2023	O'Reilly Media
3	Guide to Java: A Concise Introduction to Programming	James T.Streib, Takako Soma	2nd Edition,2023	Springer

Web links and Video Lectures(e-resources)

1. https://www.youtube.com/watch?v=OjdT2l-EZJA&list=PLfn3cNtmZdPOe3R_wO_h540QNfMkCQ0ho
2. https://www.youtube.com/watch?v=VHbSopMyc4M&list=PLBlnK6fEyqRjKA_NuK9mHmlk0dZzuP1P5
3. https://www.youtube.com/watch?v=bm0OyhwFDuY&list=PLsyeobzWxl7pe_IiTfNyr55kwJPWbgxB5
4. https://www.youtube.com/watch?v=GoXwIVyNvX0

Active Based Learning(Suggested Activity in Class)/Practical Based Learning(Example)

1. Debugging Exercises.
2. Problem Based Learning.
3. Student-Led Demonstrations.

COs and POs Mapping:

CO's	Statement	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2
CO1	Understand the fundamentals of Java programming concepts.	2												
CO2	Apply object- oriented Programming concepts to solve real world problems	3	2	2	1	2		1					1	
CO3	Develop Java applications using multithreading, generics and exceptions.	3	2	2	1	2		1					1	
CO4	Analyze code snippets or programming scenarios to identify issues and propose solutions.	2	2	2	1	2		1	2	1			1	

Academic Year: **2025-26**	Semester: **III**	Scheme: **P24**
Course Title: **Digital Systems Design**		
Course Code: **P24IS306**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P):**3:0:0**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy:**40**	Exam Hours: **3 Hrs**	
Credits:**03**		

Prerequisite: **Number Systems, Basic Gates.**

Course learning Objectives:

CLO1: Understand Boolean laws and minimization techniques and fundamental gates.

CLO2: Design of combinational logic circuits using minimum number of gates, Decoders and Multiplexers.

CLO3: Understand the Sequential logic components and Design of sequential circuits.

CLO4: Understand and use high-level hardware description languages (VHDL) to design combinational / sequential circuits.

CLO5: Conduct and Simulate practical experiments of combinational and sequential circuit.

Unit 1	**Hrs:8 Hours**

Boolean Algebra: Introduction, Logic gates , Boolean Laws, Duality, Boolean expression in standard SOP and POS , Realization using basic gates and universal gates.

Minimization Of Switching Functions: Introduction, K-Map: Two-variable, Three-variable and ,Four-variable K-map, Don't care combinations, Map entered variable(VEM), Limitation of K-map, Code converters: Binary to gray , BCD to Excess 3, Quine-Mc-Clusky method- 3 variable.

Self-Study Content: Quine-Mc-Clusky method- 4,5 variable

Text book mapping:
Text book1- Chapter 3 – 3.3-3.6, Chapter 4-4.1-4.7,4.9, , Chapter 5-5.1-5.6,5.8,5.10, Chapter 6-6.1-6.4, Chapter 7-7.9-7.13.

Unit 2	**Hrs: 8 Hours**

Combinational Logic Design: Introduction to combinational circuits, Adders, Subtractors, ripple carry adder, Look ahead carry adder**, Comparators:**1-bit and 2bit magnitude comparator**, Encoders:** octal to Binary and Decimal to BCD encoder, Priority encoders, **Decoders:** 2 to 4, 3 to 8 line decoder, **Multiplexers: 2:1,4:1, 8:1,16:1 ,** Design combinational circuits using Decoders and Multiplexers.

Self-Study Content: 7 Segment Decoder, Demultiplexer.

Text book mapping:
Text book1- Chapter 7 – 7.1-7.6,7.10, Chapter 8-8.1-8.6,8.7 , Chapter 9-9.1-9.5,9.8-9.12

Unit 3	**Hrs: 8 Hours**

Introduction to Sequential Circuits: Classification of sequential circuits: Asynchronous and Synchronous, NAND and NOR latches and flip flops: Excitation tables, State diagram and Characteristic equation of SR, JK, Race around condition, Master slave JK flip flops , Excitation tables, State diagram and Characteristic equation of D and T flip flops, Conversion of SR to JK, JK to D, T to D Flip flops.

Self-Study Content: Conversion of JK to SR, D to JK and D to T Flip flops	
Text book mapping: **Text book1-** Chapter 10 ï 10.1-10.13	
Unit4	**Hrs: 8 Hours**
Introduction to Shift Registers and Counters: Data Transmission In Shift Registers, Serial In Serial Out Shift Register, Serial In Parallel Out Shift Register, Parallel In Serial Out Shift Register, Parallel In Parallel Out Shift Register, Design of shift registers using JK and D flip Flopõs, Application Of Shift Registers: Ring Counter , Johnson Counter Up/Down Synchronous and Asynchronous Introduction, Design counters using JK and T Flip flop.	
Self-Study Content: Effects of propagation delay in ripple counters, Sequence detector design.	
Text book mapping: **Text book1-** Chapter 11 ï 11.1-11.12, Chapter 12 ï 12.1-1.0,12.11, Chapter 13 ï 13.1-13.2,13.8	
Unit 5:	**Hrs: 8 Hours**
Hardware description languages, VHDL description of combinational circuits, VHDL models for multiplexers, VHDL modules, Sequential statements and VHDL processes, Modeling Flip-flops using VHDL Processes, VHDL Modelling registers and counters using VHDL processes.	
Self-Study Content: Compilation, simulation and synthesis of VHDL code, Simple synthesis examples.	
Text book mapping: **Text book 2-** Chapter 1 ï 1.3, Chapter 2 ï 2.1-2.5, Chapter 4 ï 4.1-4.9, Chapter 8 ï 8.2-8.3,8.7	

Course Outcomes: At the end of the course students should be able to:

CO1: Apply Boolean algebra and simplification methods to minimizing Logic function.
CO2: Analyze Combinational and Sequential circuits.
CO3: Design Combinational and sequential circuit for the given problem.
CO4: Develop Combinational/ Sequential logic circuit using VHDL code.

Suggested Learning Resources:

Text books:

SL. NO	Title	Author	Year & Edition (Latest)	Publisher
1.	Fundamentals of Digital Circuits.	A. Anand Kumar	4th Edition, 2016	PHI Learning
2.	Digital Systems Design using VHDL.	Charles H.Roth, Jr., LizyKurian John	2nd Edition, 2017	McGraw-Hill Education
Reference Books:				
1.	Digital Design with an introduction to the verilog HDL, VHDL and system verilog.	M.Morris Mano, Michael D.Ciletti	6th edition,2020	Pearson Publication
2.	Digital Principles and applications.	Donald P Leach,	8th edition,2017	McGraw-Hill

		Albert Paul Malvino, GoutamSaha		Education

Web links and Video Lectures (e-resources)
https://nesoacademy.org/ec/05-digital-electronics https://dvikan.no/ntnu-studentserver/kompendier/digital-systems-design.pdf https://drive.google.com/file/d/1lw9LhePHIhwBljiWSXrmEJgXj5RE05j4/view?usp=sharing
Active Based Learning (Suggested Activity in Class)/ Practical Based Learning (Example)
• Flip Class • Seminar/ poster Presentation • Individual Role play/Team Demonstration/ Collaborative Activity • Case study • Learn by Doing

CO-PO Mapping:

CO	Statement	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2
CO1	**Apply** Boolean algebra and simplification methods to minimizing Logic function.	3	3		1	1	1		1		1		1	
CO2	**Analyze** Combinational and Sequential circuits	2	3	1	2	1	1		1	1	2		1	
CO3	**Design** Combinational and sequential circuit for the given problem.	2	2	3	1	2	2		1	2	2	1	1	
CO4	**Develop** Combinational/ Sequential logic circuit using VHDL code	2	2	3	3	3	1		1	2	2	1	1	

Academic Year: 2025-26	**Semester: III**	**Scheme: P24**
Course Title: **Data Structure Laboratory**		
Course Code: **P24ISL307**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P):**0:0:2**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy:**24**	Exam Hours: **3**	
Credits:**1**		

Sl No.	**Data Structure Laboratory Programs.**
1.	In a travel management system, distances between various travel checkpoints are recorded. Create a structure DISTANCE with members kms and meters to store these values. Write a C program that calculates the total and remaining distance between two points by performing addition and subtraction of distances using functions that accept pointers to the structure.
2.	Design a basic task management system that uses a stack (implemented using an array with a fixed maximum size MAX) to manage tasks based on their arrival time following a Last-In-First-Out (LIFO) strategy. Each task is represented by an integer (e.g., task ID or priority code). Develop a menu-driven C program to support the following operations: • Add (push) a new task onto the stack. If the stack is full, display an appropriate overflow message. • Remove (pop) the most recent task from the stack. If the stack is empty, display an underflow warning. • Display the current status of the stack, showing all tasks waiting to be processed.
3.	Develop a module for a compiler or expression evaluator that converts standard mathematical expressions from **infix notation** (e.g., A + B * C) to **postfix notation** (also known as Reverse Polish Notation, e.g., A B C * +). This conversion is essential for efficient expression evaluation using stacks. Implement a C program to perform the following: • Accept a valid infix expression containing operands and operators (+, -, *, /, ^, and parentheses). • Convert the expression into its corresponding postfix form using stack operations. • Display the resulting postfix expression suitable for evaluation by machines or interpreters.
4.	Design a recursive solution module for solving classic computational problems that frequently arise in mathematical modeling and system simulations. Implement the following operations using recursion in C: • **Disk Movement in Automation Systems (Tower of Hanoi):** Simulate the process of moving disks between pegs in an automated robotic arm system using the Tower of Hanoi logic. The objective is to move n disks from the source peg to the destination peg following the recursive strategy. • **Fault Detection in Signal Processing (GCD of Two Numbers):** Determine the Greatest Common Divisor (GCD) of two frequency values to identify synchronization intervals or signal overlaps using the Euclidean algorithm implemented recursively. • **Recursive Evaluation in Data Streams (Find Largest of 'n' Numbers):** Implement a recursive approach to find the largest value in a dynamic dataset, such as real-time sensor data or test scores, without using loops or built-in sort mechanisms.

5.	Develop a hospital emergency management system using a priority queue where each patient is assigned a priority value - the lower the number, the more critical the case. Implement a C program to: • Add a patient to the emergency queue with name and priority. • Attend to the most critical patient by removing the one with the highest priority. • Display the list of waiting patients along with their priority levels.
6.	Implement an order tracking system using a Singly Linked List (SLL) where each order is represented by an order ID (integer). Develop a C program to: • Insert n new orders either at the front (VIP orders) or at the rear (regular orders). • Delete a specific order ID after serving it, with a proper message if the ID is not found. • Display the list of all pending orders.
7.	Build a ticket booking system for a theatre or event using a queue where each booking request is stored as a string (e.g., user name or booking code). The system must ensure bookings are handled in the order they arrive. Using a linked list implementation in C, perform the following operations: • Insert a new booking request into the queue. • Remove the front request once it's processed. • Display all pending booking requests in the queue.
8.	Develop a patient record management system for a hospital using a Doubly Linked List (DLL). Each patient record should include the following details: PATIENT_ID, NAME, DIAGNOSIS, and ADMISSION_DATE. Implement a menu-driven C program to perform the following operations: • Create an ordered list of N patient records sorted by PATIENT_ID. • Count and display the total number of patients currently admitted. • Delete the patient record at a specified position (e.g., patient discharge). • Display all patient records in order, showing full details.
9.	Design a module of a computer algebra system that adds two polynomial expressions. Each polynomial is represented using a linked list, where each node contains a coefficient and exponent. Implement a C program to: • Add two polynomials by traversing their terms in descending order of exponents and combining like terms. • Display the resulting polynomial in standard mathematical format.
10.	Write a C program to simulate an inventory tracking system using a Binary Search Tree (BST). Each node represents a product, uniquely identified by an integer product_id. Through a menu-driven interface, perform the following operations: • Create the BST by inserting N product IDs. • Traverse the BST in: o Inorder (for sorted view of products), o Preorder (for serialization or storage), o Postorder (for safe deletion or restocking sequence).

Course Outcomes:

1. **Design** algorithms using different data structures like List, Stack, Queue and Trees.
2. **Develop** programs with suitable data structure based on the requirements of the real time applications.

CO-PO Mapping :

CO	Statements	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PS O1	PS O2
CO1	**Design** algorithms using different data structures like List, Stack, Queue and Trees.	3	3	3	1				2	2		2	2	
CO2	**Develop** programs with suitable data structure based on the requirements of the real time applications.	3	3	3	2	2		1	3	2		2	2	

P.E.S. College of Engineering, Mandya

Department of Information Science & Engineering

Academic Year: 2025-26	**Semester: III**	**Scheme: P24**
Course Title: **Object Oriented Programming with JAVA Laboratory**		
Course Code: **P24ISL308**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P):**0:0:2**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy:**24**	Exam Hours: **3**	
Credits:**1**		

	Note: All programs are to be implemented using JAVA Language
1	Write a Java program to simulate a simple shopping cart billing system for a retail store. The program should first prompt the user to enter the number of items they wish to purchase. For each item, collect details such as the item name, quantity, and price per unit. Then calculate total cost of all items. After calculating the total, add a 5% tax to it. Based on the total amount after tax, apply appropriate delivery charges according to the criteria shown in the table below.

Grand Total (after 5% tax)	**Delivery Charge (₹)**
Ò500	50
> 500 and Ò1000	30
> 1000	0

2	Write a Java program to implement a simple multiple-choice quiz system that asks **4 questions** to the user. Each question should have **4 options,** and the user must enter their answer by selecting a number between 1 and 4. After all answers are entered, the program should calculate the **total number of correct answers** and the **percentage score.**
3	Create a Java program for a Bank Account System without using constructors. Define a class with fields for account holder name, account number, and balance. Include methods to set account details, deposit money, withdraw (only if balance remains ₹500 or more after withdrawal), apply simple interest, and display account details. The system must ensure that a minimum balance of ₹500 is maintained at all times.
4	Design and implement a Java program that simulates a basic ticket booking system for an event. Each booking must include a customer name, number of tickets, and a unique booking ID starting from 1000. Use a default constructor to take input for customer name and ticket count from the keyboard, and a parameterized constructor to accept these values directly. Implement a method to cancel tickets with proper validation and update the total number of tickets sold, which should be tracked using a static variable. Include a method to display the total tickets sold. In the main() method, create at least two bookings, cancel some tickets from each, and display the final total tickets sold.
5	Develop a Java application that models an electricity billing system for a power supply company. The system should support billing for two types of consumers: domestic and commercial. All consumers will have shared attributes such as customer name and the number of electricity units consumed. The billing process should vary depending on the consumer type, applying a rate of ₹4 per unit for domestic consumers and ₹7 per unit for commercial consumers. Additionally, include a tax of 5% on domestic bills and 10% on commercial bills. The program should prompt the user to choose the type of consumer, accept the required input, compute the total bill including applicable tax, and display the final payable amount. To achieve this, use **inheritance** by defining a base class for common consumer details and creating two specialized classes for domestic and commercial consumers that extend this base class.

6	Write a Java program to simulate a **Smart Water Tank Monitoring System** using **interfaces** and multiple classes. Define three interfaces: • **MotorControl**ï which includes methods starMotor() and stopMotor() to control the water pump. • **WaterMonitor**ï which declares a constant MAX_LEVEL to represent the tankôs maximum capacity and a method checkLevel(int currentLevel) to monitor the current water level. • **EfficiencyCalculator** ï which defines a method calculateEfficiency(int refillAmount, int timeInSeconds) to calculate the pump's operational efficiency based on how much water was refilled and how long it took. Implement all three interfaces in a class named SmartTank. Based on the input water level from the user, if the level falls below 20% of the tank's capacity, the motor should start, and the refill amount should be computed and displayed. Then, the efficiency of the refill operation should be calculated assuming a fixed refill time. If the water level is above 90% of the capacity, the motor should stop. If the level is in a safe range (between 20% and 90%), the system should take no action. In main() take input from the user and trigger appropriate actions.
7	Develop a Java program to simulate a basic library management system using **packages** and classes. Define a package named library. books that includes a Book class to hold information such as the bookôs title, author, total number of copies, and available copies. Implement methods within the class to perform borrowing and returning operations. Create a second package named library.user containing a Student class, which stores the student's name and the title of the book they have borrowed. In the main class, import both packages and demonstrate the borrowing and returning of a book by the student.
8	Implement a Java program to simulate a patient monitoring system using **multithreading.** In this application, two threads run concurrently to monitor vital signs. The first thread, Temperature Monitor, reads a sequence of temperature values, displays each reading with a delay to mimic real-time monitoring, and then computes the average body temperature. The second thread, Heart Rate Monitor, processes a list of heart rate values, prints each reading with a time delay, identifies the maximum heart rate, and generates a warning message if any value exceeds a critical threshold of 100 beats per minute.
9	Develop a Java application to simulate a basic real-time flight booking system. The program should prompt the user to enter the passengerôs name and the number of tickets to be booked via keyboard input. Use an enumeration to define the ticket status with values like BOOKED and CANCELLED, and create a custom annotation to highlight the booking method as significant. Design a Passenger class to store the user's details and a Flight class to maintain the flight number, ticket status, and associated Passenger information. Include methods to handle booking, cancel the reservation, and display passenger and flight details before and after cancellation.
10	A university is developing a result processing system to compute and display the performance of students based on marks obtained in five subjects. Write a Java program that performs the following and handles the corresponding exceptions: • Accepts the studentôs name - Handle **Null Pointer Exception** if the name is null or not provided (input contains only spaces). • Accepts marks for n subjects, calculates total and percentage - Handle **Arithmetic Exception** for division by zero. • Displays marks of a subject based on userôs choice - Handle **ArrayIndexOutOfBoundsException** for invalid subject number.

11	Develop a Java program for an e-commerce application that displays products sorted in ascending order by **name** or **price**. Use **generics** to implement a reusable sorting method. Allow the user to choose between sorting an array of String (product names) or Double (product prices), and display the sorted results.
12	Design and develop a Java Applet application that simulates a simple digital banner system for a college information display. The applet should continuously scroll the message **"Computer Science"** across the screen. In addition to the scrolling text, the applet must display a static **welcome message** (e.g., ñWelcome to the Appletò) and dynamically show the **current system time** which updates as the banner scrolls. The application should also track how many times the banner has scrolled and display this as a **scroll count** on the applet window.

Course Outcomes:

1. **Apply** the principles of object-oriented programming to design Java programs.
2. **Evaluate** Java programs by validating logic, analysing outputs, and explaining concepts.

CO-PO Mapping

CO's	Statement	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2
CO1	Apply the principles of object-oriented programming to design Java programs.	2		1	1				1					
CO2	**Evaluate** Java programs by validating logic, analysing outputs, and explaining concepts.	2	2	2	1	1			1				2	

Academic Year: 2025-26	**Semester: III**	**Scheme: P24**
Course Title: **Digital Systems Design Laboratory**		
Course Code: **P24ISL309**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P):**0:0:2**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy:**24**	Exam Hours: **3**	
Credits:**1**		

1.	A chemical process gives out a warning signal (W = 1) when the process operates incorrectly. A logic circuit (network) is used to monitor the process and to determine whether W = 1or not. Inputs Binary Values Description of plant status C 1 Chemical rate > 10 litres/sec 0 Chemical rate <= 10 litres/sec T 1 Temperature > 91ºC 0 Temperature <=91ºC X 1 Concentration > 5 M 0 Concentration <= 5M A warning signal (W = 1) will be generated if: (a) Chemical rate < =10 litres/second (b) Temperature > 91ºC and Concentration > 5M (c) Chemical rate <= 10 litres/second and Temperature > 91ºC. Give the truth table to show all the possible situations when the warning signal could be received and design the logic circuit using only NAND gates.
2	Design Logic circuit to convert 3 bit binary to gray code using basic gates.
3	Design Full Subtrator using suitable Decoder.
4	A smart lighting system in a building uses four inputs to decide whether to turn ON the lights in a corridor. The inputs are: A: Motion detected in the corridor B: Time of day (1 = night, 0 = day) C: Light level (1 = dark, 0 = bright) D: Manual override (1 = ON, 0 = OFF) The lights should turn ON if: • Motion is detected and itôs dark • Itôs night and manual override is active Use C, B, and A as select lines and express the logic function in terms of D to connect the inputs of the multiplexer. Construct the truth table and implement using an 8:1 MUX**.**
5	Implement Master slave D FlipFlop using only NAND Gates.
6	Design and demonstrate the conversion of JK flipflop to T Flip Flop.
7.	Design and demonstrate 3-bitserialin parallel out shift register Using D Flip Flopôs.

8	Design and demonstrate 2-bit synchronous counter for the given sequence using JK Flip Flop.
9	Write the VHDL code for 8:1 Mux. Simulate and verify its working.
10	Write the VHDL code for JK and D flip-flop. Simulate and verify its working.
11	Write the VHDL code for 3-bit synchronous down counter. Simulate and verify its working.

COs	**Course Outcomes** with ***Action verbs*** for the Course topics.	**Bloom's Taxonomy Level**	**Level Indicator**
CO1	Design and Conduct experiments to realize various combinational and sequential circuits using IC.	Create	L6
CO2	Simulate using Xilinx to synthesize their designs and perform timing analysis.	Create	L6

CO-PO Mapping

CO	**Statement**	**PO1**	**PO 2**	**PO 3**	**PO 4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO 10**	**PO 11**	**PSO 1**	**PS O2**
CO 1	Design and Conduct experiments to realize various combinational and sequential circuits using IC.	3	2	3	2	2	1	1	1	2	1	1	1	1
CO 2	Simulate using Xilinx to synthesize their designs and perform timing analysis.	3	2	3	3	3	1	1	1	2	2	1	1	1

Academic Year: **2025-26**	Semester: **III**	Scheme: **P24**
Course Title: **Employability Enhancement Skills – III (CSE/ISE/ECE/CSE(AIML)/CSDS/CSBS)**		
Course Code: **P24HSMC310A**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P): **1:0:0**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy: **40 Hours**	Exam Hours: **3 Hrs**	
Credits: **01**		

Course Learning Objectives: This course will enable the students to:

- Calculations involving percentages, profit & loss and discounts.
- Explain concepts behind logical reasoning modules of direction sense and blood relations.
- Prepare students for Job recruitment process and competitive exams.
- Develop Problem Solving Skills.
- Apply programming constructs of C language to solve the real-world problem.

UNIT – I		**06 Hours**
Quantitative Aptitude: Number System ï Divisibility & Remainder, Multiples & Factors, Integers, HCF & LCM, Decimal Fractions, Surds & Indices, Simplification.		
Self-study component:	Linear equations.	
UNIT – II		**06 Hours**
Quantitative Aptitude: Percentages, Profits, Loss and Discounts. **Logical Reasoning:** Blood Relations.		
Self-study component:	Inferred meaning, Chain rule.	
UNIT – III		**06 Hours**
Logical Reasoning: Direction Sense Test. **Verbal Ability:** Change of Speech and Voice, Sentence Correction.		
Self-study component:	Height & distance.	
UNIT – IV	**C-PROGRAMMING - I**	**06 Hours**
Introduction: Keywords and Identifier, Variables and Constants, Data Types, Input/Output, Operators, Simple Programs. **Flow Control:** Ifé else, for Loop, while Loop, break and continue, switché case, goto, Control Flow Examples, Simple Programs. **Functions:** Functions, User-defined Functions, Function Types, Recursion, Storage Class, Programs **Arrays:** Arrays, Multi-dimensional Arrays, Arrays & Functions, Programs.		
Self-study component:	Evaluation of Expression.	

UNIT – V	C-PROGRAMMING - II	06 Hours

Pointers: Pointers, Pointers & Arrays, Pointers and Functions, Memory Allocation, Array & Pointer Examples.

Strings: String Functions, String Examples, Programs.

Structure and Union: Structure, Struct & Pointers, Struct & Function, Unions, Programs.

Programming Files: Files Input/output

Self-study component:	Error handling during I/O operations.

Course Outcomes: On completion of this course, students are able to:

COs	Course Outcomes with *Action verbs* for the Course topics	Bloom's Taxonomy Level	Level Indicator
CO1	Exhibit amplified level of confidence to express themselves in English.	Applying	L3
CO2	Solve the problems based on Number systems, percentages, profit & loss and discounts.	Analyzing	L4
CO3	Solve logical reasoning problems based on direction sense and blood relations.	Analyzing	L4
CO4	Apply suitable programming constructs of C language and / or suitable data structures to solve the given problem.	Applying	L3

Text Book(s):

1. The C Programming Language (2nd edition) by Brian Kernighan and Dennis Ritchie.
2. C in Depth by S K Srivastava and Deepali Srivastava.
3. Quantitative aptitude by Dr. R. S Agarwal, published by S. Chand private limited.
4. Verbal reasoning by Dr. R. S Agarwal, published by S. Chand private limited.

Reference Book(s):

1. E. Balaguruswamy, Programming in ANSI C, 7th Edition, Tata McGraw-Hill. Brian W. Kernighan and Dennis M. Ritchie, The ćCô Programming Language, Prentice Hall of India.
2. Quantitative Aptitude by Arun Sharma, McGraw Hill Education Pvt Ltd.

Web and Video link(s):

1. Problem Solving through Programming in C - https://archive.nptel.ac.in/courses/106/105/106105171/

CO → / PO ↓	**PO1**	**PO2**	**PO3**	**PO4**	**PO 5**	**PO 6**	**PO 7**	**PO 8**	**PO 9**	**PO1 0**	**PO1 1**
CO1: Exhibit amplified level of confidence to express themselves in English.	1								2	3	1
CO2: Solve problems based on number systems, percentages, profit & loss and discounts.	3	3		2	1					2	1
CO3: Solve logical reasoning problems based on direction sense and blood relations.	2	3	2	2	1				1	2	
CO4: Apply suitable programming constructs of C language and / or suitable data structures to solve the given problem.	3	3	3	2	3				1	2	2

Academic Year: **2025-26**	Semester: **III**	Scheme: **P24**
Course Title: **National Service Scheme**		
Course Code: **P24NSS311**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P): **0:0:0**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy: **20-24 Hrs**	Exam Hours: -	
Credits: **00**		

Course Outcomes (COs):

Upon successful completion of this course, students will be able to:

CO1: Analyze Indian agriculture and organic farming: Assess historical and current trends in Indian agriculture, focusing on organic farming's potential for sustainability and market access.

CO2: Design waste management systems: Apply the 5 R's to design and evaluate waste management solutions considering technical, economic, and environmental factors.

CO3: Develop women's empowerment strategies: Create plans for information-sharing platforms to address women's social and economic needs and promote community participation.

CO4: Apply engineering to sustainable development: Integrate engineering knowledge to develop practical solutions for organic farming, waste management, and community development.

CO5: Evaluate sustainable development impacts: Assess the social, economic, and environmental impacts of sustainable development initiatives.

Course Description: This course explores critical aspects of sustainable development, focusing on organic farming practices, effective waste management strategies, and initiatives for empowering women in social and economic spheres. It emphasizes practical application, problem-solving, and community engagement.

Course Content:

- Organic farming and its role in Indian agriculture (historical context, current practices, and future trends). Emphasis on connectivity for marketing organic produce.
- Waste management strategies across public, private, and governmental organizations, with a focus on the 5 R's (Reduce, Reuse, Recycle, Recover, Refuse).
- Establishing information-sharing platforms for women to address social and economic challenges.

Academic Year: **2025-26**	Semester: **III**	Scheme: **P24**
Course Title: **Yoga**		
Course Code: **P24YOG311**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P): **0:0:0**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy: **20-24 Hrs**	Exam Hours: -	
Credits: **00**		

Course Outcomes (COs):

Upon successful completion of this course, students will be able to:

CO1: Understand Yoga's principles and philosophy: Explain the meaning, history, schools, aims, and importance of prayer in Yoga.

CO2: Perform basic Yoga practices safely: Execute Suryanamaskar and selected Asanas with proper technique, breathing, and safety awareness.

CO3: Analyze Yoga's benefits and contraindications: Explain the physiological and Psychological benefits and identify contraindications and precautions for various practices.

CO4: Apply Yoga for stress management and well-being: Integrate Yoga into daily life for Stress reduction, focus enhancement, and improved well-being.

CO5: Evaluate Yoga misconceptions: Identify and debunk common myths, promoting a Scientifically informed understanding of Yoga.

Course Description: This course introduces students to the fundamental principles and practices of Yoga, emphasizing its holistic benefits for physical, mental, and emotional well-being. It explores the philosophical underpinnings of Yoga, various techniques, and their practical application in daily life. The course also addresses common misconceptions and provides guidelines for safe and effective practice.

Course Content:

- **Introduction to Yoga:**
  - Meaning and Definitions of Yoga
  - Historical Overview and Different Schools of Yoga (e.g., Hatha, Raja, Karma, Bhakti)
  - Aim and Objectives of Yoga: Physical health, mental clarity, spiritual growth, stress management.
  - Importance of Prayer and its role in Yoga
- **Yogic Practices for Common Man:**
  - Brief introduction to various yogic practices suitable for beginners.
  - Focus on promoting positive health and stress reduction.
- **Rules and Regulations for Yogic Practices:**
  - Guidelines for safe practice (e.g., appropriate time, place, clothing, empty stomach).
  - Contraindications and precautions for specific conditions.
- **Misconceptions of Yoga:**
  - Addressing common myths and misunderstandings about Yoga.
  - Clarifying the scientific basis of Yoga's benefits.
- **Suryanamaskar (Sun Salutation):**
  - Suryanamaskar prayer and its meaning.
  - Need, importance, and benefits of Suryanamaskar.
  - Detailed breakdown of the 12 counts with proper breathing and movement

coordination.
    - Practice of 2 rounds.
- **Asanas (Postures):**
    - Meaning and importance of Asanas.
    - Detailed study of the following Asanas:
        - **Sitting:** Padmasana (Lotus Pose), Vajrasana (Thunderbolt Pose)
        - **Standing:** Vrikshasana (Tree Pose), Trikonasana (Triangle Pose)
        - **Prone:** Bhujangasana (Cobra Pose), Shalabhasana (Locust Pose)
        - **Supine:** Utthitadvipadasana (Raised Two-Legged Pose), Ardha Halasana (Half Plough Pose)
    - For each Asana:
        - Meaning of the name.
        - Step-by-step technique.
        - Breathing pattern.
        - Benefits.
        - Precautionary measures and contraindications.

Academic Year: **2025-26**	Semester: **III**	Scheme: **P24**
Course Title: **Physical Education**		
Course Code: **P24PED311**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P): **0:0:0**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy: **20-24 Hrs**	Exam Hours: **-**	
Credits: **00**		

Course Outcomes: At the end of the course, the student will be able to

1. Understand the fundamental concepts and skills of Physical Education, Health, Nutrition and Fitness
2. Familiarization of health-related Exercises, Sports for overall growth and development
3. Create a foundation for the professionals in Physical Education and Sports
4. Participate in the competition at regional/state / national / international levels.
5. Create consciousness among the students on Health, Fitness and Wellness in developing and maintaining a healthy lifestyle.
6. Understand and practice of Traditional Games

Module I: Orientation 4 Hours

1. Lifestyle
2. Health & Wellness \
3. Pre-Fitness test.

Module II: General Fitness & Components of Fitness 4 Hours

1. Warming up (Free Hand exercises)
2. Strength ï Push-up / Pull-ups
3. Speed ï 30 Mtr Dash

Module III: Specific games (Any one to be selected by the student) 16 Hours

1. Kabaddi ï Hand touch, Toe Touch, Thigh Hold, Ankle hold and Bonus.
2. Kho-Kho ï Giving Kho, Single Chain, Pole dive, Pole turning, 3-6 Up.

Academic Year: **2025-26**	Semester: **III**	Scheme: **P24**
Course Title: **Basic Engineering Mathematics – I**		
Course Code: **P24MADIP301**	CIE Marks:**100**	CIE Weightage:**100%**
Teaching hours/week (L:T:P): **2:2:0**		
Teaching hours of Pedagogy: **40 Hours**		
Credits: **00**		

Course Learning Objectives: to provide basic concepts of complex trigonometry, vector algebra, differential & integral calculus, vector differentiation and various methods of solving first order differential equations.

UNIT-I	
Complex Trigonometry: Complex Numbers: Definitions & properties. Modulus and amplitude of a complex number, Argandôs diagram, De-Moivreôs theorem (without proof). Vector Algebra: Scalar and vectors. Vectors addition and subtraction. Multiplication of vectors (Dot and Cross products). Scalar and vector triple products-simple problems Self-study components: De-Moivreôs theorem (without proof). Roots of complex number - Simple problems.	12 Hrs
UNIT-II	
Differential Calculus: Polar curves ï angle between the radius vector and the tangent pedal equation- Problems. Taylors series and Maclaurinôs series expansions- Illustrative examples. Partial Differentiation: Elementary problems. Eulerôs theorem for homogeneous functions of two variables. Total derivatives-differentiation of composite and implicit function. Self-study components: Review of successive differentiation. Formulae for n^{th} derivatives of standard functions- Liebnitzôs theorem (without proof). Application to Jacobians, errors & approximations.	10 Hrs
UNIT-III	
Integral Calculus: reduction formulae for $\sin^n x, \cos^n x$ and $\sin^m x \cos^n x$ and evaluation of these with standard limits-Examples. Applications of integration to area, length of a given curve, volume and surface area of solids of revolution. Self-study components: Differentiation under integral sign (Integrals with constants limits)- Simple problems.	10 Hrs

UNIT-IV

Vector Differentiation: Differentiation of vector functions. Velocity and acceleration of a particle moving on a space curve. Scalar and vector point functions. Gradient, Divergence, Curl and Laplacian (Definitions only). Self-study components: Solenoidal and irrotational vector fields-Problems.	10 Hrs
UNIT-V	
Ordinary differential equations (ODEôs): Introduction-solutions of first order and first-degree differential equations: homogeneous, exact, linear differential equations of order one and equations reducible to above types. Self-study components: Applications of first order and first-degree ODEôs - Orthogonal trajectories of Cartesian and polar curves. Newtonôs law of cooling, R-L circuits- Simple illustrative examples from engineering field.	10 Hrs

Course Outcomes: After completing the course, the students will be able to	
CO1	Demonstrate the fundamental concepts–in complex numbers and vector algebra to analyze the problems arising in related area of engineering field.
CO2	Identify–partial derivatives to calculate rate of change of multivariate functions
CO3	Apply-the acquired knowledge of integration and differentiation to evaluate double and triple integrals to compute length surface area and volume of solids of revolution and identify velocity, acceleration of a particle moving in a space
CO4	Find analytical solutions by solving first order ODE's which arising in different branches of engineering.

Text Book:

1. B. S. Grewal, Higher Engineering Mathematics (44thEdition2018), Khanna Publishers, New Delhi.

Reference books:

1. E. Kreyszig: Advanced Engineering Mathematics, John Wiley & Sons, 10th Ed., 2015.
2. N. P. Bali and Manish Goyal: Engineering Mathematics, Laxmi Publishers, 7th Ed., 2007.

Academic Year: **2025-26**	Semester: **III**	Scheme: **P24**
Course Title: **Additional Communicative English – I**		

Course Code: **P24HDIP308**	CIE Marks: **50**	CIE Weightage:**100%**
Teaching hours/week (L:T:P): **0:2:0**	SEE Marks: **100**	SEE Weightage: -
Teaching hours of Pedagogy: **30 Hours**	Exam Hours: **3 Hrs**	
Credits: **00**		

Module-1

Introduction to Communication Skills **6 Hours**

Introduction to communication, Meaning and process, Channels of communication, Elements of communication, Barriers to effective communication. Activities - Making introductions, Sharing personal information, Describing feelings and opinions.

Module-2

Listening Skills I **4 Hours**

Hearing vs. Listening, Types of listening, Determinants of good listening, Active listening process, Barriers to listening, Activities - Listening for pronunciation practice, Listening for personal communication, Listening for communication - language functions

Module-3

Speaking Skills I **6 Hours**

Basics of speaking, Elements and Functions of speaking, Structuring your speech, Focusing on fluency, Homographs and Signpost words. Activities ï Free Speech and Pick and Speak

Module-4

Reading Skills I **4 Hours**

Developing reading as a habit, Building confidence in reading, improving reading skills, Techniques of reading - skimming and scanning. Activities - understanding studentsô attitudes towards reading, countering common errors in reading, developing efficiency in reading.

Writing Skills I **4 Hours**

Improving writing skills, Spellings and punctuation, Letter and Paragraph writing. Activity ï Writing your personal story

Module-5

Body Language and Presentation Skills **6 Hours**

Elements of body language, Types, Adapting positive body language, Cultural differences in body language. 4 Ps in presentations, Overcoming the fear of public speaking, Effective use of verbal and nonverbal presentation techniques. Activity ï Group presentations

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

CO 1: Understand the role of communication in personal and professional success

CO 2: Comprehend the types of technical literature to develop the competency of students to Apprehend the nature of formal communication requirements.

CO 3: Construct grammatically correct sentences to strengthen essential skills in speaking & writing and to develop critical thinking by emphasizing cohesion and coherence

CO 4: Demonstrate effective individual and teamwork to accomplish communication goals.

Textbooks and Reference Books:

1. Communication Skills by Sanjay Kumar and Pushpa Lata, Oxford University Press - 2015.
2. Everyday Dialogues in English by Robert J. Dixson, Prentice-Hall of India Ltd., 2006.
3. Developing Communication Skills by Krishna Mohan& Meera Banerjee (Macmillan)

4. The Oxford Guide to Writing and Speaking, John Seely, Oxford.
5. English Language Communication Skills - Lab Manual cum Workbook by Rajesh Kumar Singh, Cengage learning India Pvt Limited ï 2018

CO – PO – PSO Matrix

CO	PO												PSO		
	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1												2			
CO2										2					
CO3										2					
CO4									2						
CO									**2**	**2**		**2**			

Course Code: **P24MA401C**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P): **2:2:0**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy: **40 Hours**	Exam Hours: **3 Hrs**	
Credits: **03**		

Course Learning Objectives:

1	To build up the knowledge of Matrices and Determinants
2	Understand algebraic structures like Vector space, Inner product space and Fields.
3	To gain the knowledge of interplay between matrices and linear transformations
4	Apply Mathematical methods to solve system of linear equations and to decompose the given matrix using LU, QR and SVD methods.

Unit	**Syllabus content**	**No. of hours**	
		Theory	**Tutorial**
I	**Matrices and Linear Systems:** Introduction to Matrices and Determinants; Special Matrices-Hermitian, Unitary and Orthogonal Matrices. LU Decomposition. Solution to Linear Equations by Gauss elimination method. Applications of linear systems - in Network analysis, Balancing Chemical equation, Polynomial interpolation. Solve System of equations using MATLAB. **Self-Study:** Linear equations in Electrical Networks.	06	02
II	**Vector spaces:** Vector spaces (Axiomatic definition), Subspaces, examples. Linear Combinations, Linear Spans. Linear Dependence and Independence, Basis and Dimension. Problems. Row space, column space and null space of a Matrix-bases and dimension. The Rank theorem. Application to System of Equations ï Illustrate using MATLAB. **Self – Study:** Change of bases ï applications to differential equations, signal processing.	06	02
III	**Linear Transformations:** Linear Transformation, Geometric Linear Transformations of R^2, Kernel and Image of a linear transformation, Singular and Non-singular linear transformations. Rank-Nullity Theorem (No proof). Matrix representation of linear transformations. Change of basis-Problems. Visualize properties of Linear transformations through MATLAB. **Self – Study:** Change of bases-applications to differential equations, signal processing.	06	02
IV	**Diagonalization and quadratic forms:** Eigenvalues and Eigenvectors, Diagonalization of a matrix using eigen vectors. Inner products, inner product space, length and norm, Orthogonality. Quadratic forms and Nature of the Quadratic Forms, Positive definite matrices, Reduction of Quadratic form to canonical forms by Orthogonal Transformation.	06	02

	Determine the Eigen values and Eigen vectors using MATLAB. **Self-Study**: Iterative estimate for Eigen values and eigen vectors-Power and Inverse power method.		
V	**Matrix Decomposition:** Gram-Schmidt orthogonalization and QR decomposition. Singular value decomposition. Least Square solution of $AX = B$. Introduction to their applications in Image Processing and Machine Learning. **Self-study:** Applications to Linear Models. Principal Component Analysis. Illustrate SVD through MATLAB.	06	02

COURSE OUTCOMES: On completion of the course, student should be able to:

CO1: Understand and develop a working model in the language of matrices.

CO2: Understand the concepts of Vector spaces, linear independence, bases, dimension and linear Transformation.

CO3: Analyze and apply techniques of matrix decomposition and their applications in data analysis.

CO4: Solve problems on linear equations, matrices using MATLAB.

TEACHING – LEARNING PROCESS: Chalk and Talk, power point presentation, animations, videos.

TEXTBOOKS

1. B. S. Grewal, Higher Engineering Mathematics (44th Edition 2018), Khanna Publishers, New Delhi.
2. Seymour Lipschutz, Linear Algebra, 4th Edition, McGraw-Hill Companies, Inc., New Delhi.
3. David C. Lay, Steven R. Lay, Judi J Mc. Donald, Linear Algebra and its Applications, 6th Edition, 2021, Pearson Education.

REFERENCE BOOKS

1. P. N. Wartikar and J. N. Wartikar, Applied Mathematics, Vol I & II, Vidyarthi Prakashan.
2. Gilbert Strang, Linear Algebra and its Applications, 4th edition, 2005, Brooks Cole.3.
3. Richard Bronson & Gabriel B. Costa, Linear Algebra: An Introduction, 2nd edition, Academic Press.

Active Based Learning (Suggested Activity in Class)/ Practical Based Learning (Example)
1. Flip Class 2. Seminar/ poster Presentation

3. Individual Role play/Team Demonstration/ Collaborative Activity
4. Case study
5. Learn by Doing

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	2										
CO2	2	3										
CO3	3	2										
CO4	2	3										
Strength of correlation: Low-1, Medium-2, High-3												

Academic Year:**2025-26**	Semester: **IV**	Scheme:**P24**
Course Title: **Theory Of Computation**		
Course Code: **P24IS402**	CIE Marks: **50**	CIE Weightage:**50%**

Teaching Hours/Week(L:T:P):**3:0:0**	SEE Marks: **50**	SEE Weightage:**50%**
Total Number of Teaching Hours**:40**	Exam Hours:**3 Hrs**	
Credits:**3**		

Course Learning Objectives:

CLO1: Gain the knowledge of basic kinds of finite automata and their capabilities.

CLO2: To understanding of regular and context-free languages

CLO3: Constructing the Pushdown automata and Turing machine for Recursive languages.

UNIT– I	**8 Hours**

Finite Automata: Chomsky Hierarchy, Deterministic finite automata, Non deterministic finite automata, Finite automata with Epsilon transitions, Application of finite automata.

Self-study component:	Extended transitions and languages for DFA,NFA and ε-NFA

UNIT– II	**8 Hours**

Regular Expressions, Languages And Properties: Regular expressions, Finite Automata and Regular Expressions, Pumping Lemma for regular Languages, Equivalence and minimization of automata, Applications.

Self-study component:	Closure properties; Decision properties

UNIT– III	**8 Hours**

Context Free Grammers, Languages And Properties: Contextï free grammars, Parse trees, Ambiguity in CFG**,** The pumping lemma for CFLs, Normal forms: Chomskyôs Normal Forms, GNF, Applications.

Self-study component:	Closure properties of CFLs.

UNIT– IV	**8 Hours**

Push Down Automata: Definition of the Push down automata, the languages of a PDA, Deterministic Pushdown Automata, Equivalence of PDAôs and CFGôs, CFG to PDA.

Self-study component:	PDA to CFG

UNIT – V	**8 Hours**

Turing machines: The turing machine; Programming techniques for Turing Machines; Extensions to the basic Turing Machines, Undecidable problem that is RE, Postôs Correspondence problem.

Self-study component:	Problems that Computers cannot solve, Turing Machine and Computers.

Course Outcomes: At the end of the course students should be able to:

CO	**Course Outcomes**	**Highest Level of Cognitive Domain**

CO1	To Design Finite Automataõs for different Regular Expressions and Languages.	L3
CO2	To Construct context free grammar for various languages.	L3
CO3	To solve various problems of applying normal form techniques, push down automata and Turing Machines.	L4

Suggested Learning Resources:				
Text books:				
1	Introduction to Automata Theory. Languages. And Computation	John E.Hopcroft, Rajeev Motwani and Jeffrey D.Ullman,	3rd Edition(2013).	**Pearson Edition**
Reference Books:				
1.	Introduction to Languages and Automata Theory.	John C Martin:	3rd Edition, 2007.	Tata McGraw Hill
2.	Introduction to Computer Theory.	Daniel I.A. Cohen:	2nd Edition,2004.	John Wiley & Sons

Web links and Video Lectures (e-resources)

Web and Video link(s):

1. https://www-2.dc.uba.ar/staff/becher/Hopcroft-Motwani-Ullman-2001.pdf
2. https://www.mog.dog/files/SP2019/Sipser_Introduction.to.the.Theory.of.Computation.3E.pdf

E-Books/Resources:

3. https://tinyurl.com/bdfst7kn

Active Based Learning(Suggested Activity in Class)/Practical Based Learning(Example)

1. Flip Class
2. Collaborative Activity
3. Case study
4. Learn by Doing

CO-PO Mapping :

CO	Statement	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2
CO1	To **Design** Finite Automataõs for different Regular Expressions and Languages.	3	2	2	1	2						1	2	2
CO2	To **Construct** context free grammar for various languages.	3	2	2	1	2						1	2	2
CO3	To **Solve** various problems of applying normal form techniques, push down automata and Turing Machines.	3	2	2	1	2						1	2	2

Academic Year: 2025-26	**Semester: IV**	**Scheme: P24**
Course Title: **Design & Analysis of Algorithms**		

Course Code: **P24IS403**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P): **3:0:0**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy: **40**	Exam Hours: **3 Hrs**	
Credits: **3**		

Course learning Objectives:

CLO1: Explain various computational problem-solving techniques.
CLO2: Apply appropriate method to solve a given problem.
CLO3: Describe various methods of algorithm analysis.

Unit 1	**8 Hours**

Introduction: Algorithm, Fundamentals of Algorithmic problem solving, Important Problem Types, Fundamental Data Structures ï Graphs.

Fundamentals of the Analysis of Algorithm Efficiency: Analysis Framework, Asymptotic Notations and Basic Efficiency Classes, Mathematical analysis of Non-Recursive Algorithms with Examples [Max Element, Unique Elements]and Recursive Algorithms with Examples [Factorial, Tower of Hanoi].

Self-Study Content: Additional Examples of Mathematical analysis of Non-Recursive & Recursive Algorithms.

Text book Map : Text book1:Chapter 1 ; Chapter 2:2.1,2.2,2.3,2.4

Unit 2	**8 Hours**

Brute Force and Exhaustive Search: Selection Sort, Brute-Force String Matching, Exhaustive Search [Travelling Salesman Problem and Knapsack Problem],Depth First Search, Breadth First Search.

Decrease and Conquer: Introduction, Insertion Sort, Topological Sorting , Algorithms for generating Combinatorial objects.

Self-Study Content: Bubble Sort and Sequential Search.

Text book Map : Text book1: Chapter 3: 3.1,.3.2,3.4,3.5
Chapter 4: 4.1,4.2,4.3

Unit 3	**8 Hours**

Divide and Conquer: General Method, Merge sort, Quick Sort, Binary Search, Strassenôs Matrix Multiplication.

Transform and Conquer: Presorting, Balanced Search Trees, Heaps and Heap sort.

Self-Study Content: Binary Tree Traversals and Related Properties.

Text Book 1: Chapter 6: 6.1,6.3,6.4
Text Book 2:Chapter 3: 3.1,3.3,3.5,3.6,3.8

Unit 4	**8 Hours**

Space and Time Tradeoffs: Sorting by counting (comparison counting sort), Input Enhancement in String Matching (Horspoolôs), Hashing.

Greedy Technique: General Method, Job Sequencing with Deadlines, Primôs Algorithm, Kruskalôs Algorithm, Single Source Shortest path (Dijikstraôs Algorithm),Huffman Trees and codes.

Self-Study Content: B-Trees, Optimal Binary Search Trees.

Text Book 1: Chapter 7: 7.1,7.2,7.3
Chapter 9: 9.1,9.2,9.3,9.4
Text Book 2: Chapter 4: 4.1,4.5

Unit 5	**8 Hours**

Dynamic Programming: General Method, The Knapsack Problem, Warshallôs and Floydôs Algorithms.

Limitations of Algorithm Power: P, NP and NP- Complete Problems.

Coping with the Limitations of Algorithm Power:

Backtracking: n-Queens Problem, Subset-Sum Problem,

Branch and Bound: Knapsack Problem.

Approximation Algorithms for NP –Hard Problems : Travelling Salesperson Problem

Self-Study Content: Lower Bound Arguments, Decision trees.

Text Book 1: Chapter 8: 8.1,8.2,8.4,11.3,12.1,12.2,12.3
Text Book 2: Chapter 5: 5.1

Course Outcomes: On completion of this course, students are able to:

COs	**Course Outcomes** with ***Action verbs*** for the Course topics.
CO1	Understand the basic concepts of various algorithmic techniques
CO2	Analyze the asymptotic performance of algorithms.
CO3	Design solutions for the given problem using algorithmic technique.

Suggested Learning Resources:

Text books:

1	Introduction to the Design and Analysis of Algorithms	Anany Levitin	3rd Edition, 2012	Pearson
2	Fundamentals of Computer Algorithms	Ellis Horowitz, Satraj Sahni and Rajasekaran	2nd Edition, 2014	Universities Press

Reference Books:

1.	Introduction to Algorithms.	Thomas H.Cormen, Charles E.Leiserson, Ronal L.Rivest, Clifford Stein	3rd Edition	PHI
2	The Design and Analysis of Algorithms.	Aho, J.Hopcroft, Ullman	1st Edition, 1974	Addison-Westey

Web links and Video Lectures (e-resources):

- https://www.mooc-list.com/course/algorithms-design-and-analysis-part-1-coursera
- https://onlinecourses.nptel.ac.in/noc15_cs02/preview
- http://www.digimat.in/nptel/courses/video/106101060/L01.html

CO- PO Mapping

CO	Statements	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO1	PSO2
CO1	**Understand** the basic concepts of various algorithmic techniques.	3											2	
CO2	**Design** solutions for the given problem using algorithmic technique.	3	3	3	2				2	1		2	2	
CO3	**Analyze** the asymptotic performance of algorithms.	3	3	3	1				1				2	

Academic Year: **2025-26**	Semester: **IV**	Scheme: **P24**
Course Title: **Software Engineering**		
Course Code: **P24IS404**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P): **3:0:0**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy:**40**	Exam Hours: **3 Hrs**	
Credits:**3**		

Course learning Objectives:

CLO1:Demonstration understanding of the principles and techniques of Software Engineering.
CLO2:Analyze the various steps involved in the design process and the different design approaches which include function-oriented design and object-oriented design.
CLO3:Understand the activities in project management, requirement engineering process and to identify the different types of system models.
CLO4:Apply the knowledge of design engineering in software development.
CLO5:Provide an understanding of the principles of software engineering in a broader system context. And the notions of software engineering process and management.

Unit 1:	**8 Hours**

Overview and Requirements:

Introduction: FAQ's about software engineering, Professional and ethical responsibility; software process models, software specification, software design and implementation, software validation, software evaluation; Software Requirements: Functional and Non-functional requirements; User requirements; System requirements; software requirements document; requirements engineering processes: feasibility studies, requirements elicitation and analysis process, requirement validation and management.

Self-Study Content: Agile Process Model.

Text book Map : Text book1:Chapter 1 ,2,4

Unit 2:	**8 Hours**

Software Design:

Architectural Design: system structuring, control models, modular decomposition, domain- specific architectures; Object Oriented Design: Objects and Object Classes, An Object-Oriented Design process.

Self-Study Content: Design Evolution.

Text book Map : Text book1:Chapter 6,7

Unit 3:	**8 Hours**

Critical System, Verification and Validation:

Dependability: critical systems, availability and reliability, safety, security; critical system specification, Verification and Validation: Planning; Software inspections; clean room software development; software testing: defect testing, integration testing, system testing, workbenches.

Self-Study Content: Object Oriented Testing.

Text book Map : Text book1:Chapter 8,11,12

Unit4:	8 Hours
Management: Managing People: limits to thinking, group working, choosing and keeping people, the people capability maturity model; software cost estimation: productivity, estimation techniques, algorithmic cost modeling, project duration and staffing; quality management: quality assurance and standards, quality planning, quality control.	
Self-Study Content: Change Management.	
Text book Map : Text book1:Chapter 22,23,24	
Unit 5:	**8 Hours**
Evolution: Software change: program evolution dynamics, software maintenance, architectural evolution; software Re-engineering: source code translation, reverse engineering, program structure improvement, program modularization, data re-engineering.	
Self-Study Content: Reverse Engineering Process.	
Text book Map : Text book1:Chapter 9.3.2,25,26	
Course Outcomes: On completion of this course, students are able to:	

COs	Course Outcomes with *Action verbs* for the Course topics.
CO1	**Understand** the principles of large scale software systems, and the processes that are used to build them.
CO2	**Apply** the process of analysis and design using object oriented approach.
CO3	**Analyzing** and Identify the current trends in the area of software engineering.
CO4	**Identify** the importance of testing in assuring the quality of software with an understanding of managing risks during the progress of the project.
CO5	**Discuss** the software evolution & related issues such as version management.

Suggested Learning Resources:

Text books:

1	Software Engineering	Ian Somerville	9th Edition, 2007	Pearson Education

Reference Books:

1.	Software Engineering	A Practitioners Approach- Roger S. Pressman	7th Edition, 2007	McGraw-Hill
2	Software Engineering Theory and Practice	Shari Lawrence P- fleeger, Joanne M. Atlee	3rdEdition, 2006	Pearson Education
3	Software Engineering Principles and Practice	Waman S Jawadekar, Tata McGraw Hill	2004	-
4	Software Engineering	Pankaj Jalote, Tata McGraw Hill	-	-

CO- PO Mapping

CO	Statement	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PS O1	PS O2
CO1	**Understand** the principles of large scales of software systems, and the processes that are used to build them.	3	1	2			1	1		1			1	1
CO2	**Apply** the process of analysis and design using object oriented approach.	2		3			1			2		1		1
CO3	**Analyzing** and Identify the current trends in the area of software engineering.	2	1		1			1		1		2		1
CO4	**Identify** the importance of testing in assuring the quality of software with an understanding of managing risks during the progress of the project.	3		3			2			2	1	2		1
CO5	**Discuss** the software evolution & related issues such as version management.	2	3	3					1	2	1	2		

Academic Year: 2025-26	**Semester: IV**	**Scheme: P24**
Course Title: **Database Management Systems**		
Course Code: **P24IS405**	CIE Marks:**50**	CIE Weight age: **50%**
Teaching hours/week (L:T:P): **3:0:0**	SEE Marks:**50**	SEE Weight age: **50%**
Teaching hours of Pedagogy: **40**	Exam Hours: **3 Hrs**	
Credits: **3**		

Course Learning Objectives (CLOs):

CLO1: Understand the basic concepts of different models to design a relational database.

CLO2: Formulate SQL queries on data and improve the database design by Normalization.

CLO3: Describe the basic issues of transaction processing and concurrency control.

CLO4: Understand the advanced databases and database security.

Unit 1	**8 Hours**

Introduction to Database, Database system concepts and architecture: Databases Introduction, Characteristics of the database approach, Advantages of DBMS, Schemas, and Instances, Three Schema Architecture and Data Independence. ER model: Entity Types, Entity Sets, attributes and keys, Relation Types, Relationship Sets, roles, and structural constraints, Weak Entity Types, ER Diagrams.

Self-Study Content: Network model, Object-Oriented data models.

Teaching-Learning Process: Chalk and board, Active Learning, Problem based learning.

Unit 2	**8 Hours**

Relational Model: Relational Model Concepts, Relational Model Constraints, update operations dealing with constraint violations, Relational Database Design using ER-to-Relational mapping.

Relational Algebra: Unary and Binary relational operations, Examples of simple queries in relational algebra.

Creation of table in SQL: SQL Data Definition and Data types.

Self-Study Content: Constraint violation problems

Teaching-Learning Process: Chalk and board, Active Learning, Problem based learning.

Unit 3	**8 Hours**

SQL: Specifying constraints in SQL, retrieval queries in SQL, INSERT, DELETE, and UPDATE statements in SQL, More Complex SQL Retrieval Queries, Specifying Constraints as Assertions and Triggers, Views in SQL.

Self-Study Content: EXPLAIN command in SQL

Teaching-Learning Process: Chalk and board, Active Learning, Problem based learning.	
Unit 4	**8 Hours**
Basics of Functional Dependencies and Normalization for Relational Databases: Informal design guidelines for relation schema, Functional Dependencies: Inference rules, Normal Forms based on Primary Keys: First, Second and Third Normal Forms, Boyceï Codd Normal Form.**Transaction processing**: Introduction to Transaction processing, Transaction and System concepts, ACID property.	
Self-Study Content: Dependency preservation**.**	
Teaching-Learning Process: Chalk and board, Active Learning, Problem based learning.	

Unit 5	**8 Hours**
Transaction processing (cont.): characterizing schedules based on Serializability: Serial, Non-serial and conflict-Serializable, Testing for conflict serializability of a schedule. **Concurrency Control:** Two ï phase locking techniques, Control based on time stamp ordering. **Database Recovery:** Techniques based on Update, Shadow paging. **Database Security:** Authentication, Authorization and access control, DAC, MAC and RBAC models, Intrusion detection.	
Self-Study Content: Logical databases, Web databases, SQL injection**.**	
Course Outcomes : On completion of this course, students are able to:	

COs	**Course Outcomes** with **Action verbs** for the course topics**.**	**Bloom's Taxonomy Level**	**Level Indicator**
CO1	**Apply** the database concepts to create the relations by specifying various constraints.	Apply	L3
CO2	**Design** ER diagrams for given scenario using draw.io tool and transforms it to a relational model.	Design	L5
CO3	**Apply** suitable normalization technique to improve relational database design.	Apply	L3
CO4	**Implement** simple and complex queries for the given context using relational algebra and SQL.	Implement	L5
CO5	**Demonstrate** knowledge of concurrency control and recovery techniques in database systems.	Demonstrate	L3

Suggested Learning Resources:				
Text books:				
1	Fundamentals of Database Systems	Elmasri and Navathe	6th Edition, 2011	Addison-Wesley
Reference Books:				
1.	Data Base System Concepts	Silberschatz, Korth and Sudharshan	5th Edition, 2006	Mc-Graw Hill

2	An Introduction to Database Systems	C.J. Date, A. Kannan, S. Swamynatham	8th Edition, 2006	Pearson Education

CO-POMapping

CO	Statement	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2
CO1	**Apply** the database concepts to create the relations by specifying various constraints.	3	1	2									3	3
CO2	**Design** ER diagrams for given scenario using draw.io tool and transforms it to a relational model.	3	2	3		2			2	2	2		3	3
CO3	**Apply** suitable normalization technique to improve relational database design.	2	1	2									2	2
CO4	**Implement** simple and complex queries for the given context using relational algebra and SQL.	3	2	2	1	2							3	3
CO5	**Demonstrate** knowledge of concurrency control and recovery techniques in database systems.	2	1	2									2	2

AcademicYear**:2025-26**	Semester: **IV**	Scheme:**P24**
Course title: **Operating System**		
Course Code: **P24IS406**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week(L:T:P):**3:0:0**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy:**40**	Exam Hours:**3Hrs**	
Credits:**3**		

Course Learning Objectives:

- To familiarize the operations performed by OS as a resource Manager.
- To impart various scheduling policies of OS.
- To teach different memory management techniques.

UNIT – I	**8 Hours**

Introduction: Purpose of Operating System, Computer System Architecture, Operating System Structure, Operating System Operations

System Structures: Operating System Services, User and Operating system interface, System Calls, Types of System calls, System programs.

Processes: Process Concept, Process Scheduling, Operations on Processes, Inter-process Communication.

Self-study component:	Computer system Organization, Computing Environments, Operating System Structure(chapter 2)

UNIT – II	**8 Hours**

Threads: Overview, Multicore Programming, Multithreading Models.

File-system Implementation: File-System Structure, File-System Implementation, Directory Implementation, Allocation methods.

Self-study component:	Threading Issues, Free Space Management

UNIT – III	**8 Hours**

Process Synchronization: Critical Section Problem, Petersonôs solution, Mutexlocks, Semaphores, Classic Problems of Synchronization.

CPU Scheduling: Basic concepts, Scheduling Criteria, Scheduling Algorithms-FCFS, SJF, RR, priority.

Self-study component:	Synchronization Hardware ,Multiple-Processor Scheduling

UNIT – IV	**8 Hours**

Deadlocks: System Model, Deadlock characterization, Methods for handling deadlocks, Deadlock prevention, Deadlock avoidance, Deadlock Detection.

Main Memory: Background, Contiguous Memory Allocation, Segmentation, Paging.

Self-study component:	Recovery from deadlock, Structure of Page Table

UNIT – V	**8 Hours**

Virtual Memory: Background, Demand paging, Copy on write, Page replacement algorithms-. FIFO page replacement, Optimal page replacement, LRU page replacement.

Mass-storage structure: Disk Structure, Disk Scheduling.

Self-study component:	Thrashing, Disk Attachment.

Course Outcomes: On completion of this course, students are able to:

COs	**Course Outcomes** with ***Action verbs*** for the Course topics.
CO1	**Apply** Various Process Scheduling Algorithms, Disk Scheduling algorithms, Page replacement algorithms and Deadlock detection and avoidance techniques for providing Operating System functionalities.
CO2	**Analyze** and interpret operating system concepts to acquire a detailed understanding of the course.
CO3	**Understand** and explore the fundamental concepts of various operating system services.
CO4	**Conduct** experiments using Programming Language to demonstrate the Basic features of Operating System.

Text Book(s): Operating System Concepts Abraham Silberschatz, Peter Baer Galvin and Greg Gagn, 9th edition, John Wiley &Sons, Inc.

Reference Book(s):

1. Ann McHoes Ida M Flynn, Understanding Operating System, Cengage Learning, 6th Edition
2. D.M Dhamdhere, Operating Systems: A Concept Based Approach 3rd Ed, McGraw- Hill, 2013.
3. P.C.P. Bhatt, An Introduction to Operating Systems: Concepts and Practice 4th Edition, PHI (EEE), 2014.
4. William Stallings Operating Systems: Internals and Design Principles, 6th Edition, Pearson.

Web and Video link(s):

1. https://www.youtube.com/watch?v=vBURTt97EkA&list=PLBlnK6fEyqRiVhbXDGLXDk_OQAeuVcp2O.
2. https://www.youtube.com/watch?v=783KAB-tuE4&list=PLIemF3uozcAKTgsCIj82voMK3TMR0YE_f

E-Books/Resources:

1 **https://www.researchgate.net/publication/354665053_Operating_System_Concepts_9th20121**

CO-PO Mapping

CO	Statement	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2
CO1	**Apply** Various Process Scheduling Algorithms, Disk Scheduling algorithms, Page replacement algorithms and Deadlock detection and avoidance techniques for providing Operating System functionalities.	2	2	1									2	
CO2	**Analyze** and interpret operating system concepts to acquire a detailed understanding of the course.	2	2										2	
CO3	**Understand** and explore the fundamental concepts of various operating system services.	2	1										2	
CO4	**Conduct** experiments using Programming Language to demonstrate the Basic features of Operating System.	2	2	1	1								2	

Academic Year: 2025-26	**Semester: IV**	**Scheme: P24**
Course Title: **Design & Analysis of Algorithms Laboratory**		
Course Code: **P24ISL407**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P):**0:0:2**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy:**24**	Exam Hours: **3**	
Credits:1		

Note: All programs are to be implemented using C Language	
1.	Develop a graph traversal module that uses the Breadth-First Search (BFS) algorithm to identify and list all nodes reachable from a specified starting node within a directed graph.
2.	Implement Depth-First Search (DFS) based algorithm to compute the topological ordering of vertices in a directed acyclic graph (DAG) .
3.	Implement Merge sort algorithm to sort the given ónô unordered elements. Determine the time taken to sort the elements for different values of ónô and plot a graph of the time taken versus ónô.
4.	Implement Quick sort algorithm to sort the given ónô unordered elements. Determine the time taken to sort the elements for different values of ónô and plot a graph of the time taken versus ónô.
5.	Develop a text search module using Horspoolôs String Matching Algorithm to locate a specific pattern within a larger text.
6.	Implement Heap Sort algorithm to sort a list of unordered elements.
7.	Implement a dynamic programming algorithm for 0/1 Knapsack problem and determine the objects that are part of optimal solution.
8.	Implement Dijikstraôs algorithm to find shortest paths from a given vertex to all other vertices in a weighted connected graph.
9.	Implement Kruskalôs Algorithm to find the minimum cost spanning tree for a given undirected graph.
10.	Implement Travelling Salesperson Problem (TSP) using approximation algorithms, aiming to compute near-optimal tour paths for visiting all cities with minimal total travel cost.

Course Out comes: On completion of this course, students are able to:	
COs	**Course Out comes** with ***Action verbs*** for the Course topics
CO1	Implement the algorithms based on various algorithm design techniques.
CO2	Analyze the efficiency of various algorithms.

CO-PO Mapping:

CO	Statements	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2
CO1	**Implement** the algorithms based on various algorithm design techniques.	3	3	3	2	3		1	2	2		2	2	
CO2	**Analyze** the efficiency of various algorithms.	3	3	3	2				1	1		1	2	

Academic Year: 2025-26	**Semester: IV**	**Scheme: P24**
Course Title: **Database Management System Laboratory**		
Course Code: **P24ISL408**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P):**0:0:2**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy:**24**	Exam Hours: **3**	
Credits: 1		

1.	**Consider the following Company Database** EMPLOYEE (Fname: String, MINIT: STRING, LNAME: string, SSN: int, Bdate: date, Address: string, Sex: string, Salary: int, super_ssn: int, DNO: int) DEPARTMENT (Dname: string, Dnumber:int, mgr_ssn:int, mgr_strat_date:date) DEPT_LOCATION (Dnumber: int, Dlocation: string) PROJECT (Pname: string, Pnumber: int, Plocation:string, Dnum:int) WORKS_ON (ESSN: int, Pno:int, hours:int) DEPENDENT (essn:int, Dependent_name: string, sex: string, Bdate:date, Relationship: string) **Write the SQL Queries of the following:** 1) Retrieve the name and address of all employees who work for the 'Research' department. 2) For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birthdate. 3) For each employee, retrieve the employee's name, and the name of his or her immediate supervisor.
2.	**Consider the following Company Database** EMPLOYEE (Fname: String, MINIT: STRING, LNAME: string, SSN: int, Bdate: date, Address: string, Sex: string, Salary: int, super_ssn: int, DNO: int) DEPARTMENT (Dname: string, Dnumber:int, mgr_ssn:int, mgr_strat_date:date) DEPT_LOCATION (Dnumber: int, Dlocation: string) PROJECT (Pname: string, Pnumber: int, Plocation:string, Dnum:int) WORKS_ON (ESSN: int, Pno:int, hours:int) DEPENDENT (essn:int, Dependent_name: string, sex: string, Bdate:date, Relationship: string) **Write the SQL Queries of the following:** 1) Retrieve the name of each employee who has a dependent with the same first name and same sex as the employee.

	2) Retrieve the employee numbers of all employees who work on project located in Bellaire, Houston, or Stafford. 3) Find the sum of the salaries of all employees, the maximum salary, the minimum salary, and the average salary. Display with proper headings.
3.	**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 ï 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. Partition the BOOK table based on year of publication. Demonstrate its working with a simple query.
4.	**Consider the following database for a Banking enterprise:** BRANCH (branch-name: string,branch-city: string,assets: real) ACCOUNT (accno:int,branch-name: string,balance: real) DEPOSITOR (customer-name: string,accno:int) CUSTOMER (customer-name: string,customer-street: string,city:string) LOAN (loan-number:int,branch-name: string,loan- number-int) BORROWER (customer-name: string,customer-street: string,city: string) **Write SQL queries to** 1. Create the above tables by properly specifying the primary and foreign keys. 2. Enter 5 tuples for each relation. 3. Find all the customers who have atleast two accounts at the main branch. 4. Demonstrate how you delete all account tuples at every branch located in a specified city.
5.	**Consider the following database for a Sports League Management System:** TEAMS (team_id: int, team_name: string, city: string) PLAYERS (player_id: int, player_name: string, age: int, position: string, team_id: int) MATCHES (match_id: int, match_date: date, home_team_id: int, away_team_id: int, home_score: int, away_score: int) STATS (stat_id: int, player_id: int, match_id: int, goals: int, assists: int, yellow_cards: int, red_cards: int)

	COACHES (coach_id: int, coach_name: string, team_id: int, experience_years: int) Create the above tables by properly specifying the primary and foreign keys. 1. List all players in a specific team (e.g., team_id = 1) 2. Get the result of all matches where a specific team (e.g., team_id = 2) played Find top 5 players with the most goals.
	PART-B **Mini-Project**

Course Outcomes: On completion of this course, students are able to:			
CO's	**Course Outcomes** with ***Action verbs*** for the Course topics	**Bloom's Taxonomy Level**	**Level Indicator**
CO1	**Design** ER diagrams for given scenario using draw.io tool and transforms it to a relational model.	Design	L5
CO2	**Implement** simple and complex queries for the given context using SQL.	Implement	L5

CO-PO Mapping

CO	Statements	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO 1	PSO 2
CO1	**Design** ER diagrams for given scenario using draw.io tool and transforms it to a relational model.	3	2	3		2			2	2	2		3	3
CO2	**Implement** simple and complex queries for the given context using SQL.	3	2	2	1	2			2	2	2		3	3

Academic Year**: 2025-26**	Semester**: IV**	Scheme**: P24**
Course Title: **Operating System Laboratory**		
Course Code: **P24ISL409**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P):**0:0:2**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy:**24**	Exam Hours: **3**	
Credits:**1**		

1.	Program to implement the Process system calls.
2.	Program to create a Process using API.
3.	Program to implement Sequential file allocation method.
4.	Program to simulate Single level directory file organization technique.
5.	Program to simulate the concept of Dining-Philosopherôs problem.
6.	Program to implement CPU scheduling algorithm for Shortest Job First.
7.	Simulate Bankerôs algorithm for Dead Lock Avoidance.
8.	Program to implement and simulate the MFT algorithm.
9.	Program to implement FIFO page replacement technique.
10.	Program to simulate FCFS Disk scheduling algorithm.

COs	**Course Outcomes** with ***Action verbs*** for the Course topics.
CO1	Implement OS concepts in process, memory, file, CPU, and disk management.
CO2	Analyse and compare OS algorithms for efficiency and performance.

CO-PO Mapping:

COs	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PSO1	PSO2
CO1: Implement OS concepts in process, memory, file, CPU, and disk management.	3				3						1	2	
CO2: Analyse and compare OS algorithms for efficiency and performance.		3			3						1	2	

Academic Year: **2025-26**	Semester: **IV**	Scheme: **P24**
Course Title: **Employability Enhancement Skills – IV (CSE/ISE/ECE/CSE(AIML)/CSDS/CSBS)**		
Course Code: **P24HSMC410A**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P): **1:0:0**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy: **40 Hours**	Exam Hours: **3 Hrs**	
Credits: **01**		

Course Learning Objectives: This course will enable the students to:

- Calculations involving simple and compound interest, averages, allegations & mixtures, proportions, variations and partnership.
- Explain concepts behind logical reasoning modules of series, coding & decoding, seating and data arrangements.
- Develop problem solving skills through Data structures.

UNIT – I		**06 Hours**
Quantitative Aptitude: Simple and Compound Interest, Averages. **Logical Reasoning:** Series, Coding & Decoding.		
Self-study component:	Mensuration	
UNIT – II		**06 Hours**
Quantitative Aptitude: Allegations and Mixtures, Ratios, Proportions and Variations. **Logical Reasoning:** Seating Arrangement, Data Arrangement.		
Self-study component:	Types of cryptarithm	
UNIT – III		**06 Hours**
Quantitative Aptitude: Partnership. **Verbal Ability:** Sentence Completion, Ordering of Sentences.		
Self-study component:	Game based assessments	
UNIT – IV	**DATA STRUCTURES I - Problem Solving Techniques and Object-Oriented Programming**	**06 Hours**

Recursion: Introduction to recursion, Principle of mathematical induction, Fibonacci numbers, Recursion using arrays, Recursion using strings, Recursion using 2D arrays.

Time and Space Complexity: Order complexity analysis, Theoretical complexity analysis, Time complexity analysis of searching and recursive algorithms, Theoretical space complexity, Space complexity analysis of merge sort.

Backtracking: Introduction to Backtracking, Rat In a Maze, N-queen, Word Search.

Basics of OOP: Introduction to oops, Creating objects, Getters, and setters, Constructors and related concepts, Inbuilt constructor and destructor, Example classes.

Advance Concepts of OOP: Static members, Function overloading and related concepts, Abstraction, Encapsulation, Inheritance, Polymorphism, Virtual functions, Abstract classes, Exception handling.

Self-study component:	Examples of Abstract Data Type

UNIT – V	DATA STRUCTURES II – Linear Data Structures and Tress	06 Hours

Linked Lists: Introduction to linked list, Inserting node in linked list, Deleting node from linked list, Midpoint of linked list, Merge two sorted linked lists, merge sort of a linked list, Reversing a linked list.

Stacks and Queues: Introduction to stacks, Stack using arrays, Dynamic Stack class, Stack using linked list, Inbuilt stack, Queue using arrays, Dynamic queue class, Queue using linked list, Inbuilt queue.

Generic Trees: Introduction to Trees, Making a tree node class, Taking a tree as input and printing, Tree traversals, Destructor for tree node class.

Binary Trees: Introduction to Binary Trees, Taking a binary tree as input and printing, Binary Tree traversals, Diameter of binary tree.

Binary Search Trees: Introduction to Binary Search Trees, Searching a node in BST, BST class, Inserting and Deleting nodes in BST, Types of balanced BSTs.

Self-study component:	Huffman tree, Expression Trees.

Course Outcomes: On completion of this course, students are able to:

COs	**Course Outcomes** with ***Action verbs*** for the Course topics	**Bloom's Taxonomy Level**	**Level Indicator**
CO1	Solve the problems based on simple and compound interests, averages, allegations & mixtures, ratios, proportions, variations and partnerships.	Applying	L3
CO2	Solve logical reasoning problems based on seating arrangements, data arrangement and verbal ability skills of sentence corrections and ordering of sentences.	Applying	L3
CO3	Analyze and represent various data structures and its operations.	Analyzing	L4
CO4	Develop programs with suitable data structure based on the requirements of the real-time applications	Applying	L3

Text Book(s):

1. Data Structures and Algorithms Made Easy by Narasimha Karumanchi
2. Data Structures through C in Depth by by S K Srivastava and Deepali Srivastava
3. Quantitative aptitude by Dr. R. S Agarwal, published by S. Chand private limited.
4. Verbal reasoning by Dr. R. S Agarwal, published by S. Chand private limited.

Reference Book(s):

1. Aaron M Tenenbaum, Yedidyah Langsam and Moshe J Augenstein, ñData Structures using Cò, 2014, low price edition ,Pearson education.
2. Seymour Lipschutz ,òData Structures with C (Schaum's Outline Series)ò , July 2017, McGraw Hill Education.
3. Quantitative Aptitude by Arun Sharma, McGraw Hill Education Pvt Ltd.

CO ↓ / PO →	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11
CO1: Solve problems on simple & compound interest, averages, alligations & mixtures, ratios, proportions, variations, partnerships.	3	3		2	1					1	2
CO2: Solve logical reasoning & verbal ability problems (arrangements, sentence ordering).	2	3	1	1					1	3	1
CO3: Analyze & represent various data structures and their operations.	3	3	3	3	3					1	1
CO4: Develop programs with suitable data structures for real-time applications.	3	3	3	2	3				1	1	2

Academic Year: **2025-26**	Semester: **IV**	Scheme: **P24**
Course Title: **National Service Scheme**		
Course Code: **P24NSS411**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P): **0:0:2**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy: **20-24 Hrs**	Exam Hours: -	
Credits: **00**		

Course Outcomes (COs):

Upon successful completion of this course, students will be able to:

CO1: Analyze and propose water conservation: Assess water resource issues and recommend conservation strategies considering stakeholder roles.

CO2: Develop rural business proposals: Create actionable business proposals for increasing village income, including market analysis and implementation plans.

CO3: Enhance educational outcomes and access: Design and implement initiatives to improve school performance and promote higher/technical/vocational education enrolment.

CO4: Apply engineering to community development: Integrate engineering knowledge to develop solutions for water conservation, business development, and educational initiatives.

CO5: Evaluate community development impacts: Assess the social, economic, and environmental impacts of community development projects.

Course Description: This course focuses on practical strategies for community development, covering water conservation techniques, business development in rural areas, and educational enhancement initiatives. It emphasizes stakeholder engagement, project planning, and implementation.

Course Content:

- Water conservation techniques, the role of different stakeholders (e.g., government, communities, NGOs), and implementation strategies.
- Developing actionable business proposals to increase village income and outlining implementation approaches.
- Supporting local schools to improve academic results and increase enrolment in higher/technical/vocational education.

Academic Year: **2025-26**	Semester: **IV**	Scheme: **P24**
Course Title: **Yoga**		
Course Code: **P24YOG411**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P): **0:0:0**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy: **20-24 Hrs**	Exam Hours: -	
Credits: **00**		

Course Outcomes (COs):

Upon successful completion of this course, students will be able to:

CO1: Understand Yoga's ethics and philosophy: Explain Patanjali's Ashtanga Yoga (Yamas and Niyamas) and their relevance to personal and professional life.

CO2: Perform Yoga practices safely: Execute Suryanamaskar, selected Asanas, Kapalabhati, and Pranayama techniques with correct technique, breathing, and safety awareness.

CO3: Analyze Yoga's effects: Describe the benefits and contraindications of practiced techniques, explaining their impact on body and mind.

CO4: Apply Yoga for well-being: Integrate Yoga for stress management, focus, mindfulness, and overall well-being.

CO5: Understand Yoga's interconnectedness: Articulate the relationship between physical practices,mental states, and ethical principles in Yoga.

Course Description: This course introduces students to the ancient practice of Yoga, focusing on its physical, mental, and ethical dimensions. It covers key components of Patanjali's Ashtanga Yoga, including Yamas and Niyamas, along with practical training in Asanas, Suryanamaskar, Pranayama, and Shatkarmas like Kapalabhati. The course aims to equip students with tools for stress management, improved focus, and overall well-being.

Course Content:

- **Patanjali's Ashtanga Yoga:** Yama (Ahimsa, Satya, Asteya, Brahmacharya, Aparigraha), Niyama (Shaucha, Santosha, Tapas, Svadhyaya, Ishvarapranidhana)
- **Suryanamaskar:** 12 counts, 4 rounds
- **Asanas:**
  - Sitting: Sukhasana, Paschimottanasana
  - Standing: Ardhakati Chakrasana, Parshva Chakrasana
  - Prone: Dhanurasana
  - Supine: Halasana, Karna Peedasana
- **Kapalabhati:** 40 strokes/min, 3 rounds
- **Pranayama:** Suryanuloma-Viloma, Chandranuloma-Viloma, Suryabhedana, Chandra Bhedana, Nadishodhana

Meaning, Need, importance of Pranayama. Di fferent types. Meaning by name, technique, precautionary measures and benefits of each Pranayama

Academic Year: **2025-26**	Semester: **IV**	Scheme: **P24**
Course Title: **Physical Education**		
Course Code: **P24PED411**	CIE Marks:**50**	CIE Weightage:**50%**
Teaching hours/week (L:T:P): **0:0:0**	SEE Marks:**50**	SEE Weightage:**50%**
Teaching hours of Pedagogy: **20-24 Hrs**	Exam Hours: -	
Credits: **00**		

Course Outcomes: At the end of the course, the student will be able to

1. Understand the ethics and moral values in sports and athletics
2. Perform in the selected sports or athletics of studentôs choice.
3. Understand the roles and responsibilities of organisation and administration of sports and games.

Module I: Ethics and Moral Values 4 Hours

1. Ethics in Sports
2. Moral Values in Sports and Games

Module II: Specific Games (Any one to be selected by the student) 16 Hours

1. Volleyball ï Attack, Block, Service, Upper Hand Pass and Lower hand Pass.
2. Athletics (Track Events) ï Any event as per availability of Ground

Module III: Role of Organization and administration 4 Hours

Academic Year: **2025-26**	Semester: **IV**	Scheme: **P24**
Course Title: **Basic Engineering Mathematics – II**		
Course Code: **P24MADIP401**	CIE Marks:**100**	CIE Weightage:**100%**
Teaching hours/week (L:T:P): **2:2:0**		
Teaching hours of Pedagogy: **40 Hours**		
Credits: **00**		

Course Objectives: To provide essential concepts of linear algebra, introductory concepts of second & higher order differential equations along with various techniques/methods to solve them, Laplace & inverse Laplace transforms and elementary probability theory.

UNIT-I	
Linear Algebra: Introduction - Rank of matrix by elementary row operations - Echelon form of a matrix. Consistency of system of linear equations - Gauss elimination method. Gauss-Jordan and LU decomposition methods. Eigen values and Eigen vectors of a square matrix. **Self-study Components**: Application of Cayley-Hamilton theorem (without proof) to compute the inverse of a matrix-Examples.	10 Hrs
UNIT-II	
Higher order ODE's: Linear differential equations of second and higher order equations with constant coefficients. Homogeneous /non-homogeneous equations. Inverse differential operators and variation of parameters. Solution of Cauchyôs homogeneous linear equation and Legendreôs linear differential equation. **Self-study Components:** Method of undetermined coefficients	14 Hrs
UNIT-III	
Multiple Integrals: Double and triple integrals-region of integration. Evaluation of double integrals by change of order of integration. **Vector Integration**: Vector Integration: Integration of vector functions. Concept of a line integrals, surface and volume integrals. Greenôs, Stokesôs and Gauss theorems (without proof) problems. **Self-study Components:** Orthogonal curvilinear coordinates.	10Hrs
UNIT-IV	

Laplace transforms: Laplace transforms of elementary functions. Transforms of derivatives and integrals, transforms of periodic function and unit step function-Problems only. Inverse **Laplace transforms**: Definition of inverse Laplace transforms. Evaluation of Inverse transforms by standard methods. **Self-study Components:** Application to solutions of linear differential equations and simultaneous differential equations.	12Hrs
UNIT-V	
Probability: Introduction. Sample space and events. Axioms of probability. Addition and multiplication theorems. Conditional probability ï illustrative examples. **Self-study Components:** State and prove Bayesôs theorem.	06 Hrs

Course Outcomes: After completing the course, the students will be able to	
CO1	Apply matrix theory for solving systems of linear equations in the different areas of linear algebra.
CO2	Solve second and higher order differential equations occurring in electrical circuits, damped/un-damped vibrations.
CO3	Identify-the technique of integration evaluates double and triples integrals by change of variables, and vector integration technique to compute line integral.
CO4	Explore the basic concepts of elementary probability theory and apply the same to the problems of decision theory.

Text Book:

1. B. S. Grewal: Higher Engineering Mathematics, Khanna Publishers, New Delhi,43rd Ed.,2015.

Reference books:

1. E. Kreyszig: Advanced Engineering Mathematics, John Wiley & Sons, 10th Ed., 2015.
2. N. P. Bali and Manish Goyal: Engineering Mathematics, Laxmi Publishers, 7th Ed., 2007.

Academic Year: **2025-26**	Semester: **IV**	Scheme: **P24**
Course Title: **Additional Communicative English – II**		
Course Code: **P24HDIP408**	CIE Marks: **100**	CIE Weightage:**100%**
Teaching hours/week (L:T:P): **0:2:0**	SEE Marks: **100**	SEE Weightage: **-**
Teaching hours of Pedagogy: **30 Hours**	Exam Hours: **3 Hrs**	
Credits: 00		

Module-1	
Listening Skills II	**2 Hours**
Levels of listening, Active listening, Techniques of listening. Activity: Listening for main ideas and Listening for specific information	
Speaking Skills II	**6 Hours**
Language of discussion ï Giving opinion, agreeing / disagreeing, asking questions, making suggestions. Sentence stress ï content and structure words, Speaking situations, Intonations and Summarizing skills	
Module-2	
Reading Skills II	**2 Hours**
Guessing meaning from the context, Understanding graphical information, Summarizing. Activity: Book review	
Writing Skills II	**4 Hours**
Linkers and connectives, Sentence and paragraph transformation, Mind mapping techniques, Letter writing, Essay writing	
Module-3	
Email Etiquette	**4 Hours**
Parts of an email, Writing an effective subject line, email language and tone. Activity: Email writing practice - Scenario based emails	
Group Presentations	**2 Hours**
Group presentations by the students	
Module-4	
Goal Setting	**2 Hours**
Defining goals, types of goals, Establishing SMART goals, Steps in setting goals, Goal setting activity	
Individual Presentations	**4 Hours**
Individual presentation by the students	
Module-5	
Teamwork	**4 Hours**
Defining teams, Team vs. Group, Benefits and challenges of working in teams, Stages of team building, Building effective teams, Case studies on teamwork	

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

CO 1: Understand the role of communication in personal and professional success

CO 2: Comprehend the types of technical literature to develop the competency of students to apprehend the nature of formal communication requirements.

CO 3: Construct grammatically correct sentences to strengthen essential skills in speaking & writing and to develop critical thinking by emphasizing cohesion and coherence

CO 4: Demonstrate effective individual and teamwork to accomplish communication goals.

Textbooks and Reference Books:

1. Communication Skills by Sanjay Kumar and Pushpa Lata, Oxford University Press - 2015.
2. Everyday Dialogues in English by Robert J. Dixson, Prentice-Hall of India Ltd., 2006.
3. Developing Communication Skills by Krishna Mohan& Meera Banerjee (Macmillan)
4. The Oxford Guide to Writing and Speaking, John Seely, Oxford.
5. English Language Communication Skills - Lab Manual cum Workbook by Rajesh Kumar Singh, Cengage learning India Pvt Limited ï 2018
6. The 7 Habits of Highly Effective People by Stephen R Covey, Simon & Schuster ï 2020
7. You Are the Team: 6 Simple Ways Teammates Can Go from Good to Great by Michael G. Rogers

CO – PO – PSO Matrix

CO	PO												PSO		
	PO1	PO 2	PO 3	PO4	PO5	PO6	PO7	PO8	PO9	PO 10	PO 11	PO 12	PS O1	PS O2	P S O 3
CO1												2			
CO2										2					
CO3										2					
CO4									2						
CO									**2**	**2**		**2**			