



AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

(Approved by A.I.C.T.E., New Delhi & Permanently Affiliated to JNTU-GV, Vizianagaram)

NAAC "A+" Accredited Institute

Cherukupally (Village), Near Tagarapuvalasa Bridge, Vizianagaram (Dist), AP, Pin-531162.

www.aietta.ac.in, principal@aietta.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Institute Vision:

To develop highly skilled professionals with strong ethical and human values.

Institute Mission:

M1: To impart quality education through effective teaching–learning practices, industrial exposure, and professional training.

M2: To produce competent and highly knowledgeable graduates with a positive attitude and problem-solving ability.

M3: To instill self-confidence, leadership qualities, and social responsibility to face professional and societal challenges.

Department Vision:

To emerge as a center of excellence in Computer Science and Engineering education and research, producing competent, industry-ready professionals with innovation, ethical values, and social responsibility.

Department Mission:

M1: To impart strong foundational knowledge in computer science, programming, and emerging technologies through effective teaching–learning practices.

M2: To develop problem-solving, analytical, and research skills by encouraging innovation, projects, and industry interaction.

M3: To produce industry-ready professionals with proficiency in modern tools, teamwork, and professional communication.

M4: To inculcate ethical values, social responsibility, and lifelong learning skills to meet professional and societal challenges.

Program Educational Objectives (PEOs):

PEO1: Graduates will apply strong foundations in computer science, mathematics, and engineering principles to analyze and solve real-world problems.

PEO2: Graduates will design, develop, and deploy software systems and applications using modern Engineering tools, technologies, and industry-standard practices.

PEO3: Graduates will pursue successful careers in industry, entrepreneurship, higher education, or research through continuous learning and skill enhancement.

PEO4: Graduates will demonstrate professional ethics, effective communication, teamwork, and leadership while contributing responsibly to society.



AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

(Approved by A.I.C.T.E., New Delhi & Permanently Affiliated to J.N.T.U-GV, Vizianagaram)

NAAC "A+" Accredited Institute

Cherukupally (Village), Near Tagarapuvalasa Bridge, Vizianagaram (Dist) -531162.

www.aietta.ac.in, principal@aietta.ac.in

Program Specific Outcomes (PSOs):

PSO1: Apply core computing concepts to design efficient software and systems.

PSO2: Use modern tools and emerging technologies to solve real-world problems ethically.

Program Outcomes (POs):

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 Team work: 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



AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

(Approved by A.I.C.T.E., New Delhi & Permanently Affiliated to J.N.T.U-GV, Vizianagaram)

NAAC “A+” Accredited Institute

Cherukupally (Village), Near Tagarapuvalasa Bridge, Vizianagaram (Dist) -531162.

www.aietta.ac.in, principal@aietta.ac.in

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

PO11: Life-Long Learning: Recognize the need for, and have the preparation and ability for i) independent and life-long learning ii) adaptability to new and emerging technologies and iii) critical thinking in the broadest context of technological change. (WK8).



AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

(Approved by A.I.C.T.E., New Delhi & Permanently Affiliated to JNTU-GV, Vizianagaram)

NAAC “A+” Accredited Institute

Cherukupally (Village), Near Tagarapuvalasa Bridge, Vizianagaram (Dist), AP, Pin -531162.

www.aietta.ac.in, principal@aietta.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Course Structure

Program– B. Tech Computer Science and Engineering

Regulation-R24

(Applicable from the academic year 2024-2025 to 2026-2027)

Program: B.Tech Computer Science and Engineering

Regulation: R24

II Year I Semester-Course Structure

S.No	Category	Course Code	Course Title	Hours per Week			
				Lecture	Tutorial	Practical	Credits
1	PC	R24CSPC03	Mathematical Foundations of Computer Science	3	0	0	3
2	ES	R24ES09	Design and Analysis of Algorithms	3	0	0	3
3	PC	R24CSPC04	Object Oriented Programming using JAVA	3	0	0	3
4	HS	R24HS03	Universal Human Values-Understanding Harmony & Human Ethical Conduct	2	0	0	2
5	PC	R24CSPC05	Software Engineering and Project Management	3	0	0	3
6	PC	R24CSPC06	Object Oriented Programming using JAVA Lab	0	0	3	1.5
7	ES	R24ES12	Design and Analysis of Algorithms Lab	0	0	3	1.5
8	SC	R24CSSC01	Python Programming	0	1	2	2
9	HS	R24HS04	Logical Reasoning and Corporate Skills	0	0	2	1
10	MC	R24MC03	Environmental Science	2	0	0	-
Total				16	01	10	20

Category	Courses	Credits
ES- Engineering Science Course	2	4.5
PC-Proficitional Core Course	4	10.5
HS- Humanity Science and Management Course	2	03
SC- Skill Oriented Course	1	02
MC-Mandatory Course	1	00
Total	10	20



AVANTHI INSTITUTE OF ENGINEERING AND TECHNOLOGY

(Autonomous)

(Approved by A.I.C.T.E., New Delhi & Permanently Affiliated to JNTU-GV, Vizianagaram)

NAAC “A+” Accredited Institute

Cherukupally (Village), Near Tagarapuvalasa Bridge, Vizianagaram (Dist), AP, Pin -531162.

www.aietta.ac.in, principal@aietta.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Program: B. Tech Computer Science and Engineering

Regulation: R24

II Year II Semester- Course Structure

S.No	Category	Course Code	Course Title	Hours per Week			
				Lecture	Tutorial	Practical	Credits
1	BS	R24BS11	Probability & Statistics	3	0	0	3
2	PC	R24CSPC08	Formal Languages and Automata Theory	3	0	0	3
3	PC	R24CSPC09	Database Management Systems	3	0	0	3
4	PC	R24CSPC10	Digital Logic and Computer Organization	3	0	0	3
5	PC	R24CSPC11	Operating Systems	3	0	0	3
6	PC	R24CSPC12	Database Management Systems Lab	0	0	3	1.5
7	PC	R24CSPC13	Operating Systems Lab	0	0	3	1.5
8	PC	R24CSPC14	Advanced Coding using C & JAVA Lab-I	0	0	3	1.5
9	SC	R24CSSC02	Full Stack Development	0	0	3	1.5
10	HS	R24HS05	Numerical Ability and Professional Communication Skills	0	0	2	1
11	MC	R24MC04	Intellectual Property Rights	2	0	0	-
Total				17	00	14	22
Mandatory Community Service Project Internship of 08 weeks duration during summer vacation							

Category	Courses	Credits
BS- Basic Sciences Course	1	03
PC-Proficitional Core Course	7	16.5
SC- Skill Oriented Course	1	1.5
HS- Humanity Science and Management Course	1	01
MC-Mandatory Course	1	--
Total	11	22

Chairperson
Board of Studies (CSE)

Course Objectives:

The course is designed with the objective to:

- Apply the knowledge in equivalence formulas, tautological implications in finding normal forms, and theory of inference in Statement Calculus and predicates, and explain Mathematical Induction principle and apply the same
- Develop the skill in equivalences and inference theory in Predicate Calculus
- Learn the properties of relations, Hasse diagram, and applying problem solving skills to find GCD and testing for prime numbers.
- Acquire the knowledge and skill to draw graphs and related structures and to apply the knowledge skill in solving problem
- Enhance skill and knowledge to draw of trees and solve minimum cost spanning tree problems and solve and formulate, generating functions and recurrence relations.

Course Outcomes

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs										
		PO1	PO2	PO3	PO4	PO5	PO6	PO10	PO11	PSO1	PSO2	DoK
R24CSPC03.1	Apply equivalence formulas, tautological implications in finding normal forms, and theory of Inference in Statement Calculus and predicates	3	3	2	2	3	2	3	2	3	2	L2
R24CSPC03.2	Apply skill in equivalences and inference theory in Predicate Calculus and GCD and testing for prime numbers	3	3	3	3	3	3	3	2	3	2	L3
R24CSPC03.3	Students gain the knowledge of the properties of relations, Posets, Lattices, Sets and apply the same in solving the problems.	3	3	3	3	3	2	3	2	3	3	L2, L3, L4
R24CSPC03.4	Identify the properties of graphs and related structures and solve the related problems	3	3	3	2	3	3	3	3	3	2	L3, L4

R24CSPC03.5	Identify the properties of Trees and solve minimum cost spanning tree problems and solve Generating functions and recurrence relations	3	3	3	2	3	3	-	-	-	-	L5 , L6
-------------	--	---	---	---	---	---	---	---	---	---	---	---------------

SYLLABUS

UNIT-I

14 hours

Mathematical logic: Fundamentals (statements and notations, connectives, Truth tables), Tautologies, Equivalence of formulas, Tautological implications, Normal forms, Theory of Inference for statement calculus.

CO's-CO1

Self-Learning Topics:

- Build and analyze truth tables manually.
- Practice rewriting expressions into CNF/DNF.
- Memorize common logical equivalences.

UNIT-II

14 hours

Predicate Calculus: Predicate logic, statement functions, variables and quantifiers, free and bound variables. Inference Theory of the Predicate Calculus: Logical implication involving quantifiers, Statements with more than one variable.

CO's-CO2

Self-Learning Topics:

- Memorize Inference Rules.
- Practice symbolic representation of logical statements with quantifiers

UNIT-III

15 hours

Relations: Relations, Properties of Relations, Equivalence relations, Operations on relations, Representation of relations, partial orders, Hasse Diagram, Lattices, properties of Lattices, Special types of Lattices (Proofs not required).

Number Theory: Properties of Integers, Division Theorem, The Greatest Common Divisor, Euclidean Algorithm, Least Common Multiple, Testing for Prime Numbers, The Fundamental Theorem of Arithmetic.

CO's-CO3

Self-Learning Topics:

- Practice solving problems on GCD
- Memorize the properties of relations
- Practice solving problems on LCM and testing for prime numbers

UNIT-IV

13 hours

Graph Theory: Basic Concepts of Graphs, Matrix representation of graphs: Adjacency Matrices, Incidence Matrices, Isomorphic Graphs, Paths and Circuits, Eulerian & Hamiltonian graphs, Planar

Graphs, Graph coloring.

CO's-CO4

Self-Learning Topics:

- Memorize all types of graphs and their properties
- Practice the problems on Isomorphic, Hamiltonian and Eulerian graphs

UNIT-V

14 hours

Trees: Introduction, Spanning Trees, BFS, DFS, Minimum cost spanning trees (Prim's & Kruskal's),

Recurrence relations: Generating Function of Sequences, calculating coefficient of Generating function, Partial Fractions. First order Linear Homogeneous and Non- Homogeneous recurrence relations, Method of generating functions, Method of characteristic root.CO5

Self-Learning Topics:

- Practice BFS, DFS, Prim's, Kruskal
- Solve recurrence problems using both methods

Board of Studies : Computer Science and Engineering

Approved in BoS No : 02, 9th May, 2025

Approved in ACM No: 02

Text Books:

1. Trembly J.P. and Manohar. P, "Discrete Mathematical Structures with applications to computer science,"Tata McGraw Hill, New Delhi, 2017
2. Kolman B, Busoy R.C, Ross S.C, Discrete Mathematical Structures, 5thEdition,PrenticeHall, 2004.
3. D.S. Chandra sekharaiyah, "Mathematical Foundation of Computer Science" Prism Publications 2009.

Reference Books:

1. J.L. Mott, A. Kandel, T. P Baker, "Discrete Mathematics for Computer Scientists and Mathematicians," Prentice Hall India, 2nd Edition2015
2. Kenneth H. Rosen, "Discrete Mathematics and its Applications," Tata McGraw Hill, New Delhi, 7th edition, 2017
3. V. Krishnamurthy, "Combinatory: Theory and Applications", East-West Press. Seymour Lipschutz, M.Lipson, "Discrete Mathematics" Tata McGraw Hill, 2005.

Reference Links:

1. <https://nptel.ac.in/courses/106106094>
2. <https://nptel.ac.in/courses/106103205>
3. <https://nptel.ac.in/courses/106106183>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35%	--
L2	40%	--
L3	25%	30%
L4	--	30%
L5	--	25%
L6	--	15%
Total (%)	100%	100%

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define Tautology
2. Define Relation
3. Define free and bound variables
4. Define Equivalence relation
5. Define simple graph, degree of a sequence, degree of a vertex?
6. Define Lattice
7. Define planar graphs

L2: Understand

1. Define quantifiers
2. Write the properties of relation
3. Write the rules of inference theory of predicate logic
4. Define Partial order relation
5. Write the types of lattices
6. Define Adjacency matrix and Incidence matrix representation
7. Define Spanning trees

L3: Apply

1. Show that $P \rightarrow (Q \rightarrow R) \leftrightarrow P \rightarrow (\neg Q \vee R)$
2. Obtain PDNF for $\neg P \vee Q$
3. Let $X = \{ 1,2,3,4\}$ and relation $R=\{ (x,y) / x > y\}$. Draw the graph of R and also give its

matrix

4. Let R, S be two relations on $A=\{1,2,3\}$ and $R= \{ (1,1), (1,2), (2,3), (3,1), (3,3) \}$, $S=\{(1,2), (1,3), (2,1), (3,3)\}$ then find the following
5. $R \cup S$
6. $R \circ R$
7. $R - S$
8. Is the degree sequence $(1, 2, 3, 3, 3, 5, 5)$ a graphic?
9. Solve the recurrence relation $a_n - a_{n-1} + 10a_{n-2} = 0$ for $n \geq 2$ using generating function method

L4: Analysing

1. Show that $(x) (P(x) \rightarrow R(x))$ follows logically from the premises $(x) (P(x) \rightarrow Q(x))$ and
2. $(x) (Q(x) \rightarrow R(x))$
3. Let $\{(2,3,4,9,12,18), | \}$ be a poset. Draw its Hasse Diagram
4. Write the rules of isomorphism of graphs
5. Define Hamiltonian graph and write the rules for constructing Hamiltonian paths and cycles
6. Show that $K_{3,3}$ is non planar
7. Determine whether the integer 133 is prime or not.

L5: Evaluating

1. Show that R is a valid inference from the premises $P \rightarrow Q$, $Q \rightarrow R$ and P .
2. Find GCD of 414 and 662 using Euclidean algorithm.
3. Find the GCD of 615 and 1080 and find integers u and v such that $\text{g.c.d}(615, 1080) = 615u + 1080v$

L6: Creating

1. Explain Kruskal algorithm with an example
2. Solve $a_n - 7a_{n-1} + 10a_{n-2} = 4^n$
3. Explain Prims algorithm with an example

**Chairperson
Board of Studies (CSE)**

R24ES09

Design and Analysis of Algorithms
(Common to CSE, CSE (DS), CSE (AI&ML))

3 0 0 3

Course Objectives:

1. To understand and apply the algorithm analysis techniques
2. To critically analyze the efficiency of alternative algorithmic solutions for the same problem
3. To understand different algorithm design techniques
4. To understand the limitations of Algorithmic power

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

Course Code	Course Outcomes	Mapping with POs and PSOs							Dok
		PO1	PO2	PO3	PO4	PS01	PS02	PS03	
R24ES09.1	Review the fundamentals of algorithmic problem solving and analyzing efficiency of algorithms	3	3	1	0	2	3	3	L1,L2,L4
R24ES09.2	Apply mathematical formulation, complexity analysis and methodologies	3	3	1	0	2	3	3	L2,L3
R24ES09.3	Compare the time complexities of various algorithms	2	3	0	1	3	2	3	L3,L4
R24ES09.4	Critically analyze the different algorithm design techniques for a given problem	2	3	0	2	2	3	3	L3,L4
R24ES09.5	Illustrate NP class problems and formulate solutions using standard approach	1	2	0	1	2	3	3	L4,L5

SYLLABUS

UNIT-I: Introduction of an Algorithm
15 Hours

Notion of an Algorithm – Fundamentals of Algorithmic Problem Solving –Important Problem Types – Fundamentals of the Analysis of Algorithmic Efficiency –Asymptotic Notations and their properties - Analysis Framework –Empirical analysis - Mathematical analysis for Recursive and Non-recursive algorithms –Visualization.

Self-Learning Topics: Amortized analysis

COs–CO1
UNIT–II: Brute Force and Divide-and-Conquer
13 Hours

Brute Force – Computing an – String Matching - Closest-Pair and Convex-Hull Problems - Exhaustive Search - Travelling Salesman Problem – Knapsack Problem - Assignment Problem.

Divide and Conquer Methodology – Binary Search – Merge sort – Quick sort –Heap Sort - Multiplication of Large Integers – Closest-Pair and Convex – Hull Problems.

Self-Learning Topics: Knuth-Morris-Pratt Algorithm

COs–CO2
UNIT-III: Dynamic Programming and Greedy Techniques
14 Hours

Dynamic programming – Principle of optimality - Coin Changing Problem -Computing a Binomial Coefficient – Floyd's Algorithm – Multi Stage Graph -Optimal Binary Search Trees – Knapsack Problem and Memory functions

Greedy Technique – Container Loading Problem - Prim's Algorithm and Kruskal's Algorithm – 0/1 Knapsack Problem - Optimal Merge pattern -Huffman Trees.

Self-Learning Topics: Travelling Sales Person Problem

COs–CO3

UNIT- IV Iterative Improvement

14 Hours

The Simplex Method - The Maximum-Flow Problem – Maximum Matching in Bipartite Graphs - Stable Marriage problem

Self-Learning Topics: Hill Climbing

COs–CO4

UNIT-V: Coping with the Limitations of Algorithm Power

14 Hours

Lower - Bound Arguments - P, NP, NP - Complete and NP-Hard Problems-Backtracking – n-Queen Problem - Hamiltonian Circuit Problem – Subset Sum Problem. Branch and Bound – LIFO Search and FIFO Search –Assignment Problem – Knapsack Problem – Travelling Salesman Problem.

Self-Learning Topics: Approximation Algorithms for NP-Hard Problems – Travelling Salesman Problem – Knapsack Problem

COs–CO5

Board of Studies : Computer Science and Engineering

Approved in BOS No: 02, 9th May, 2025

Approved in ACM No: 02

Expert Talk (To be delivered by SMEs from Industries)

	COs	POs / PSOs
1. Branch and Bound Algorithmic Approach	CO5	PO1,PO2,PO4,PSO1,PSO2,PSO3

Text Books:

1. Anany Levitin, “Introduction to the Design and Analysis of Algorithms”, Third Edition, Pearson Education, 2012.
2. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, “Computer Algorithms/ C++”, Second Edition, Universities Press, 2007

Reference Books:

1. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and Clifford Stein, “Introduction to Algorithms”, Third Edition, PHI Learning Private Limited, 2012.
2. Alfred V.Aho, John E. Hopcroft and Jeffrey D. Ullman, “Data Structures and Algorithms”, Pearson Education, Reprint 2006.
3. Harsh Bhasin, “Algorithms Design and Analysis”, Oxford university press,2016.
4. S. Sridhar, “Design and Analysis of Algorithms”, Oxford university press,2014

Web References:

1. <https://nptel.ac.in/courses/106101060>
2. https://www.cse.iitm.ac.in/course_details.php?arg=OTI
3. https://swayam.gov.in/nd1_noc19_cs47/previ

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
-----------------	---------------------------	---------------------------

L1	35	-
L2	30	-
L3	20	30
L4	15	40
L5	-	30
TOTAL (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. What is an algorithm, and why is it important in computer science?
2. Name two examples of important problem types in algorithm design.
3. What does asymptotic notation analyze, and what are three common types of asymptotic notations?
4. Define dynamic programming and provide a short description of its principle of optimality.
5. What is the difference between NP and NP-Hard problems?

L2: Understand

1. Explain the purpose of empirical analysis versus mathematical analysis when evaluating algorithm efficiency.
2. Describe how the divide-and-conquer methodology works and provide an example of an algorithm that uses this approach.
3. Discuss the advantages and disadvantages of using a brute force approach versus a more efficient algorithm like quick sort.
4. What role does visualization play in understanding and analyzing algorithms?
5. How does the Simplex Method solve linear programming problems?

L3: Apply

1. Given an algorithm to sort a list of numbers, analyze its time complexity using asymptotic notation to demonstrate its efficiency.
2. Implement a brute-force solution for the string matching problem, and analyze the time complexity of your approach.
3. Use dynamic programming to solve the Coin Changing Problem and explain your solution step-by-step.
4. Apply the concept of the maximum-flow problem to a specific network example and demonstrate how to calculate the maximum flow.
5. Create a greedy algorithm to tackle the 0/1 Knapsack Problem and explain the logic behind your selection at each step.

L4: Analyze

1. Compare the performance of the naive recursive solution versus a dynamic programming solution for the Fibonacci sequence in terms of time and space.
2. Analyze why the Travelling Salesman Problem is categorized as NP-Complete and discuss its implications for algorithm design.
3. Evaluate the efficiency of various algorithms (like Merge Sort and Quick Sort) for sorting a

large dataset under different conditions.

4. Investigate the effectiveness of the branch-and-bound method compared to backtracking for solving combinatorial optimization problems.
5. Examine the potential trade-offs in solving an optimization problem using iterative improvement methods versus exact algorithms.

L5: Evaluating

1. Critically assess how well amortized analysis can predict the performance of data structures over time.
2. Evaluate the practical applications of approximation algorithms for NP-Hard problems, citing specific examples.
3. Discuss the implications of classifying problems into different categories (P, NP, NP-Complete, and NP-Hard) in the context of computer science and real-world applications.
4. Assess the effectiveness of dynamic programming techniques in practical scenarios compared to greedy algorithms.
5. Evaluate the significance of understanding algorithmic limitations in practical computing, particularly regarding the P vs NP debate.

**Chairperson
Board of Studies (CSE)**

R24CSPC04
Object Oriented Programming using JAVA
3 0 0 3

(Common to CSE, CSE (DS) & CSE (AI&ML))

Course Objectives:

The main objectives of the course is to

- Understand the basic ideas of Object-Oriented Programming (OOP) and implement them using Java through classes, objects, inheritance, and polymorphism.
- Understand and implement basic Java programming using data types, operators, control statements, and arrays to solve simple problems.
- Implement programs using Java features like constructors, method overloading, exception handling, and multithreading for better program design.
- Understand and use Java's built-in libraries like collections, file handling, and interfaces to create useful and organized programs.
- Implement graphical programs using JavaFX and understand how to create simple network-based applications using Java networking.

Course Outcomes

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

Course Code	Course Outcomes	Mapping with POs and PSOs							Dok
		PO 1	PO 2	PO 3	PO 5	PO 11	PS0 1	PS0 2	
R24CSPC04.01	Implement object orientated programming strategies and Contrast classes and objects.	3	2	2	3	2	3	3	L2, L3
R24CSPC04.02	Analyze Inheritance and Dynamic Method Dispatch	3	3	2	2	2	2	3	L2, L3, L4
R24CSPC04.03	Demonstrate various classes in different packages and can design own packages	3	2	3	3	2	2	3	L2, L3
R24CSPC04.04	Manage Exceptions and Apply Threads	3	3	3	3	2	2	3	L3, L4
R24CSPC04.05	Create GUI screens along with event handling and write network programs	3	2	3	3	3	2	3	L5, L6

SYLLABUS
UNIT-I:
15 Hours

INTRODUCTION TO OBJECTS & CLASSES: What is Object Oriented Programming? Object Orientation as a New Paradigm, An Overview of Java: Process Oriented Vs Object Oriented

Programming, OOP Principles, Java BuzzWords, The Byte Code, A First Simple Program. Class Fundamentals with Variables and Methods, Declaring objects for accessing variables and methods. Data Types and Variables, Operators and Expressions, Control Statements, Type Conversion and Casting, Lexical Issues in Java, command line arguments,

ARRAYS: Introduction, Declaration and Initialization of Arrays, Single Dimension, Operations on Array Elements, Sorting of Arrays, Search for Values in Arrays, Multi Dimension. **CO's-CO1**

Self-learning topics:

- Difference between JDK, JRE, and JVM
- History and evolution of Java
- Best practices for writing Java code (naming conventions, comments, code formatting)
- Enhanced for-loop and foreach in Java

UNIT-II:

13 Hours

CONSTRUCTORS: Default and Parameterized, this keyword and Garbage Collection, Final and Static Keywords, Overloading Methods, Overloading Constructors, Using objects as Parameters, Returning objects, Strings, String methods, String Buffer and String Builder.

INHERITANCE: Inheritance Basics, Types of Inheritance, Using Super keyword for constructors, Super to call variables and methods, Method Overriding, Dynamic Method Dispatch. **CO's-CO2**

Self-learning topics:

- Constructor chaining and private constructors
- Immutable classes and the final keyword usage in depth
- StringBuilder vs StringBuffer: performance comparison

UNIT-III:

15 Hours

PACKAGES AND INTERFACES: Defining a Package, importing a package, Package Example, Access Protection, An Access Example, Abstract classes, Interfaces: Defining and Implementing Interfaces

EXPLORING java.lang: Wrapper classes, Object, Math, Runtime

EXPLORING java.util: The collection framework: Array List, HashSet HashMap, StringTokenizer, Calendar, Random, and Scanner

EXPLORING java.io: File class, Byte Streams, Character Streams, File Input Stream, File Output Stream, File Reader and File Writer classes. **CO's-CO3**

Self-learning topics:

- Creating custom packages and JAR files
- Interface vs Abstract class: When to use what?
- Iterators and ListIterator in Java Collections

- Serialization and Deserialization using java.io

UNIT-IV:

14 Hours

EXCEPTION HANDLING: Exception Handling Fundamentals, Exception Types, throw, throws and finally, Creating your own exceptions, Chained Exceptions.

MULTITHREADED PROGRAMMING: Java Thread Model, The Main thread, Two ways of Creating a Thread, Creating Multiple Threads, is Alive(), join(), Thread Priority, Synchronization, Inter Thread Communication.

COs-CO4

Self-learning topics:

- Try-with-resources statement in exception handling
- Uncaught exceptions and the default handler
- Daemon threads and thread groups

UNIT-V:

12 Hours

JAVA FX GUI: Java FX Scene Builder, Java FX App Window Structure, displaying text and image, event handling, laying out nodes in scene graph, mouse events

EXPLORING JAVA.NET: Socket, Server Socket, Inet Address, Data Gram Socket, URL, Client Server Program using Sockets.

COs-CO5

Self-learning topics:

- Creating custom controls in JavaFX
- Building forms and handling validation in JavaFX
- Introduction to Java NIO (New I/O)
- Understanding protocols (TCP vs UDP) in Java networking

Board of Studies : Computer Science and Engineering

Approved in BoS No : 02, 9th May, 2025

Approved in ACM No: 02

Text Books:

1. Herbert Schildt, Java The complete reference, 11th Edition, McGrawHill, 2019
2. Timothy budd, An introduction to object-oriented programming, 3rd Edition, Pearson, 2009.
3. JAVA 9 for Programmers, Paul Deitel, Harvey Deitel, 4th Edition, Pearson.

Reference Books:

1. Cay S. Horstmann, Core Java Volume I–Fundamentals, 11th Edition, Pearson 2019
2. Y. Daniel Liang Introduction to Java Programming Comprehensive Version, 10th Edition, Pearson, 2015.

3. Bruce Eckel, Thinking in Java, 4th Edition, Prentice Hall, 2006

Reference Links:

1. http://en.wikibooks.org/wiki/Java_Programming - Java Learning Wiki Book
2. <http://www.javabeginner.com> - Java Beginner Tutorial

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35%	--
L2	40%	--
L3	25%	35%
L4	--	35%
L5	--	20%
L6	--	10%
Total (%)	100%	100%

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define Object-Oriented Programming.
2. List different types of inheritance supported by Java.
3. What is an exception in Java?
4. What is the Java thread model?
5. What is JavaFX?
6. Compare and contrast Process-Oriented Programming with Object-Oriented Programming. Provide examples.
7. What is bytecode in Java? Describe how Java achieves platform independence using bytecode and JVM.
8. What are multi-dimensional arrays? How are they declared and used in Java? Provide an example of a 2D array.
9. What is a constructor in Java? Explain constructor overloading with an example.
10. What is garbage collection in Java? How does Java handle memory management and object destruction?

11. Explain the different types of inheritance in Java. Why is multiple inheritance not supported through classes?
12. What are wrapper classes in Java? Why are they needed? Explain auto boxing and unboxing with examples.
13. Discuss the ArrayList class. How does it differ from arrays? Provide examples of adding, removing, and accessing elements.
14. What are File Reader and File Writer classes? How do they work with character data? Write a program to demonstrate file reading/writing.
15. Write a Java program to demonstrate client-server communication using Socket and Server Socket. Explain how data is sent and received.

L2: Understand

1. Explain the concept of encapsulation with an example.
2. What is a constructor? How is it different from a method?
3. What is a package? How is it created in Java?
4. What is an exception? How is it different from an error?
5. What is JavaFX? How does it differ from AWT and Swing?
6. Explain the key principles of Object-Oriented Programming (OOP). Compare process-oriented programming with object-oriented programming using suitable examples.
7. Discuss the various data types available in Java. Also explain type conversion and casting with examples.
8. Define constructors in Java. Differentiate between default and parameterized constructors. Illustrate with example
9. What is method overloading and constructor overloading in Java? How are they useful? Give examples for both.
10. What are packages in Java? How do you define and import packages? Explain access protection in Java with an example.
11. Define interfaces. How are they implemented in Java? Compare abstract classes and interfaces with code examples
12. What is exception handling in Java? Explain the types of exceptions and the role of try, catch, throw, throws, and finally blocks with examples.
13. Explain the Java thread model. How can threads be created in Java? Illustrate both approaches with code examples.
14. What is JavaFX? Describe the structure of a JavaFX application window. How are text and images displayed in JavaFX?
15. Explain JavaFX event handling mechanism. How can mouse events be handled in JavaFX?

Give a suitable code example.

L3: Apply

1. Write a Java program to demonstrate type casting from double to int
2. Write a Java method that accepts an object as a parameter and prints one of its attributes.
3. Write a Java program that uses the Math class to find the square root of 25.
4. Write a Java program to handle divide-by-zero exception using try-catch.
5. Write JavaFX code to handle a mouse click event on a rectangle.
6. Develop a Java program that takes user input via command line arguments, performs basic arithmetic operations using operators, and displays the results.
7. Write a Java program to demonstrate use of various control statements (if, switch, for, while) in calculating factorial of a number, checking for even/odd, and printing a pattern.
8. Write a Java program that demonstrates the use of default and parameterized constructors in a Book class. Display book details using methods.
9. Write a Java program that overrides a method from a superclass and uses dynamic method dispatch to invoke overridden methods during runtime.
10. Create a custom package shapes with classes Circle and Rectangle. Use them in another Java program by importing the package and calculating area of shapes.
11. Define an interface Vehicle with methods start() and stop(). Create classes Car and Bike that implement this interface. Demonstrate polymorphism using the interface reference.
12. Write a Java program that demonstrates exception handling using multiple catch blocks. Include finally block and handle exceptions like division by zero and array index out of bounds.
13. Write a multithreaded Java program where multiple threads update a shared counter variable. Use synchronization to avoid data inconsistency.
14. Write a JavaFX application using Scene Builder or code that includes text input, a dropdown, and a button. Display selected values in a label upon button click.
15. Design a JavaFX application with a label, two text fields, and a button. When the user enters two numbers and clicks the button, display their sum using an event handler.

L4: Analysing

1. Differentiate between type conversion and type casting with an example.
2. Identify differences between method overloading and constructor overloading
3. Analyze the role of access modifiers in package-level protection.
4. Compare checked and unchecked exceptions with examples?
5. Compare JavaFX with Swing in terms of GUI design and performance.
6. Explain with examples how Java's data types, type casting, and operators interact during

expression evaluation.

7. Compare single, multilevel, and hierarchical inheritance with examples. What are the implications of using inheritance in software design?
8. Analyze the memory handling in Java with garbage collection. How does object creation and destruction differ from other languages like C++?
9. Compare and analyze abstract classes and interfaces in Java. In what situations would you use one over the other? Give code examples.
10. Analyze the role of Java packages in access protection and modularity. Provide a detailed example of defining and importing custom packages.
11. Create a custom exception class. Analyze the impact of creating user-defined exceptions in large-scale applications.
12. Analyze the architecture of a JavaFX application. Explain how the Scene Graph and layout panes are used to build a responsive GUI.
13. Compare event handling in traditional Java AWT/Swing with JavaFX. How are mouse events handled in JavaFX with code examples?

L5: Evaluating

1. Evaluate the significance of bytecode in Java's platform independence.
2. Evaluate the use of the final keyword for security and immutability in Java.
3. Evaluate the use of access modifiers in packages for building secure APIs.
4. Evaluate the role of event handling in improving GUI interactivity.
5. Evaluate the role of constructors (default and parameterized) in initializing objects. How does constructor overloading enhance object creation flexibility?
6. Evaluate the use of final and static keywords. How do they influence the design and behavior of Java programs?
7. Evaluate the importance of the Java Collection Framework. How do ArrayList, HashSet, and HashMap differ in usage and performance?
8. Evaluate Java's approach to exception handling. How does it improve code reliability compared to traditional error handling mechanisms?
9. Compare TCP (using Socket) and UDP (using DatagramSocket) for client-server communication. Which protocol is preferable in real-time applications and why?
10. Evaluate the role of Java networking classes (Socket, ServerSocket, InetAddress, URL) in building distributed applications.

L6: Create

1. Construct a Java collection-based application (using ArrayList, HashMap, and HashSet) for managing student records with operations like insert, search, and delete.

2. Design and implement a multithreaded ticket booking system that demonstrates thread synchronization to prevent overbooking.
3. Construct a banking application where multiple threads represent clients depositing and withdrawing money. Use synchronization to ensure account balance consistency.
4. Create a JavaFX GUI-based calculator using scene graph layout. Implement features like number input, arithmetic operations, and keyboard/mouse interaction.
5. Design a Client-Server chat application in Java using Socket and Server Socket. Include features like timestamp, sender name, and simple UI for message display.

**Chairperson
Board of Studies (CSE)**

R24HS03 Universal Human Values-Understanding Harmony & Human Ethical Conduct
(Common to CSE, CSE (DS), CSE (AI&ML), ECE, EEE, ME) **2 0 0 2**

Course Objectives:

The main objectives of the course are to:

- To enable students to recognize the vital connection between **values and skills**, emphasizing how their integration leads to lasting happiness and prosperity core aspirations shared by all human beings.
- To guide students in developing a **holistic outlook on life and profession** grounded in a deep understanding of human nature and existence. This perspective supports the cultivation of **Universal Human Values** and encourages a natural transition toward **value-based living**.
- To illustrate the practical outcomes of a holistic understanding, particularly in fostering **ethical behavior, trust-based and fulfilling relationships**, and **harmonious interaction with nature**.

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

Course Code	Course Outcomes	Mapping with POs & PSOs											DoK
		PO01	PO02	PO03	PO04	PO05	PO06	PO07	PO08	PO09	PO10	PO11	
R24HS03.1	Develop an understanding of human values.	2	2	-	-	2	3	3	2	2	-	3	L1, L2
R24HS03.2	Achieve harmony within self and body.	2	-	-	-	-	3	3	2	-	-	3	L2, L3
R24HS03.3	Build trustful and respectful relationships.	2	-	-	2	3	2	3	3	3	-	3	L3, L4
R24HS03.4	Understand harmony in nature.	2	2	-	-	-	3	3	2	2	-	3	L4, L5
R24HS03.5	Practice ethical and value-based living.	2	2	-	-	2	2	3	2	2	-	3	L5, L6

SYLLABUS

Unit I: Introduction to Value Education (6 lectures and three tutorials for practice session)

Lecture 1: Introduction to Human Development – Physical, Emotional, and Intellectual Dimensions

Lecture 2: Core Concepts of Value Education in Contemporary Society

Tutorial 1: Practice Session PS1 – Reflective Sharing on Life Values and Personal Experiences

Lecture 3: Self-awareness and Self-exploration: Tools for Inner Clarity

Lecture 4: Aspirations of the Human Being – Joy, Fulfillment, and Meaning

Tutorial 2: Practice Session PS2 – Exploring Self-awareness and Emotional Intelligence

Lecture 5: Challenges to Human Fulfillment in Modern Life

Lecture 6: Mindfulness and Ethical Decision-Making as a Path to Human Aspiration Fulfillment

Tutorial 3: Practice Session PS3 – Identifying Personal Ethical Dilemmas and Responses **CO's CO1**

Self-Learning Concepts: Explore how education influences the development of core human values, the significance of emotional intelligence in personal growth, the role of mindfulness in enhancing self-awareness, the impact of consumerism and modern lifestyle on ethical choices, and reflect on one's own aspirations and value system – all of which build a foundation for inner development.

Unit II: Harmony in the Human Being (6 lectures and three tutorials for practice session)

Lecture 7: Human Being: Alignment of Thoughts, Emotions, and Actions

Lecture 8: Needs and Desires: Distinguishing Between Essentials and Superfluous

Tutorial 4: Practice Session PS4 – Mind Mapping Personal Needs and Desires

Lecture 9: The Role of Body, Mind, and Consciousness in Well-being

Lecture 10: Cultivating Harmony within the Self – Practices and Techniques

Tutorial 5: Practice Session PS5 – Journaling: Tracking Inner Conflicts and Clarity

Lecture 11: Body Awareness and Physical Well-being Practices (Yoga/Meditation)

Lecture 12: Sustainable Self-Regulation and Lifestyle Choices for Health

Tutorial 6: Practice Session PS6 – Wellness and Lifestyle Assessment Activity **CO's-CO2**

Self-Learning Concepts: Understand the benefits of journaling for emotional and mental clarity, examine the relationship between diet, exercise, and mental well-being, explore how to identify and manage sources of stress, study the scientific basis of meditation and mindfulness, and create a personalized routine for achieving physical and emotional balance – fostering harmony within the self.

Unit III: Harmony in the Family and Society (6 lectures and three tutorials for practice session)

Lecture 13: Foundations of Healthy Relationships – Emotional and Ethical Dimensions

Lecture 14: Cultivating Trust and Empathy in Family and Social Contexts

Tutorial 7: Practice Session PS7 – Role-play on Building Trust in Relationships

Lecture 15: Respect and Empathy – Cornerstones of Human Interaction

Tutorial 8: Practice Session PS8 – Group Activity: Expressing Respect in Conversations

Lecture 16: Conflict Resolution and Emotional Safety in Social Interaction

Lecture 17: Principles of Ethical Leadership and Community Engagement

Lecture 18: Global Citizenship and Social Justice in the Modern World

Tutorial 9: Practice Session PS9 – Simulation: Creating a Model Ethical Society **CO's-CO3**

Self-Learning Concepts: Learn how to practice active listening and empathy, recognize and overcome personal biases in relationships, analyze inspiring examples of ethical leadership and community service, understand the concept of emotional safety in interpersonal dynamics, and explore the role of volunteerism in building a responsible and connected society – enhancing harmony in relationships and the social fabric.

Unit IV: Harmony in the Nature/Existence (4 lectures and two tutorials for practice session)

Lecture 19: Ecological Intelligence and Human-Nature Relationship

Lecture 20: Sustainability and Resource Responsibility – A Human Imperative

Tutorial 10: Practice Session PS10 – Case Study: Local Environmental Issue & Action Plan

Lecture 21: Systems Thinking – Viewing Nature as an Interconnected Whole

Lecture 22: Eco-Spirituality and the Harmony of Existence

Tutorial 11: Practice Session PS11 – Nature Walk + Reflection on Human-Nature Coexistence

CO's-CO4

Self-Learning Concepts: Study the principles of minimalism as a lifestyle for environmental consciousness, calculate and reduce your carbon footprint, compare environmental issues in urban and rural areas, explore sustainable agriculture and local food movements, and research the involvement of youth in global climate action – deepening your understanding of harmony with nature.

Unit V: Implications of the Holistic Understanding – a Look at Professional Ethics (6 lectures and tutorials for practice session)

Lecture 23: Personal Integrity and Professional Ethics – Building the Link

Lecture 24: Ethics in the Digital Age – Responsibility and Cyber Citizenship

Tutorial 12: Practice Session PS12 – Case Analysis: Ethical Dilemma in Technology Use

Lecture 25: Humanistic Leadership and Social Entrepreneurship

Lecture 26: Corporate Social Responsibility and Sustainable Business Models

Tutorial 13: Practice Session PS13 – Designing a Value-based Startup Idea

Lecture 27: Innovation with Purpose: Case Studies in Ethical Innovation

Lecture 28: Life Planning and Career Pathways Anchored in Values

Tutorial 14: Practice Session PS14 – Action Plan: Aligning Career with Core Human Values

CO's-CO5

Self-Learning Concepts: Explore ethical concerns in the digital age such as data privacy and responsible AI use, understand the concept of human-centered design in business, study successful examples of green and socially responsible enterprises, formulate your own personal code of ethics, and analyze real-life ethical dilemmas encountered in the workplace – promoting value-based professionalism and ethical living.

Practice Sessions for UNIT I – Introduction to Value Education

PS1 Sharing about Oneself

PS2 Exploring Human Consciousness

PS3 Exploring Natural Acceptance

Practice Sessions for UNIT II – Harmony in the Human Being

PS4 Exploring the difference between the Needs of the self and body

PS5 Exploring Sources of Imagination in the Self

PS6 Exploring Harmony of self with the body

Practice Sessions for UNIT III – Harmony in the Family and Society

PS7 Exploring the Feeling of Trust

PS8 Exploring the Feeling of Respect

PS9 Exploring Systems to Fulfill Human Goals

Practice Sessions for UNIT IV – Harmony in the Nature (Existence)

PS10 Exploring the Four Orders of Nature

PS11 Exploring Co-existence in Existence

Practice Sessions for UNIT V – Implications of the Holistic Understanding – a Look at Professional Ethics

PS12 Exploring Ethical Human Conduct

PS13 Exploring Humanistic Models in Education

PS14 Exploring Steps of Transition towards Universal Human Order

Board of Studies: Master of Business Administration

Approved in BoS No: 02, 13th May, 2025

Approved in ACM No: 02

Text Book and Teachers Manual

1. **R R Gaur, R Asthana, G P Bagaria**, A Foundation Course in Human Values and Professional
2. Ethics, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034-47-1 **R R Gaur, R Asthana, G P Bagaria**, Teachers' Manual for A Foundation Course in Human
3. Values and Professional Ethics, 2nd Revised Edition, Excel Books, New Delhi, 2019. ISBN 978-93-87034-53-2

Reference Books:

1. **Jeevan Vidya: Ek Parichaya**, A Nagaraj, Jeevan Vidya Prakashan, Amarkantak, 1999.
2. **Human Values**, A.N. Tripathi, New Age Intl. Publishers, New Delhi, 2004.
3. **The Story of Stuff** (Book).
4. **The Story of My Experiments with Truth** - by Mohandas Karamchand Gandhi
5. **Small is Beautiful** - E. F Schumacher.
6. **Slow is Beautiful** - Cecile Andrews
7. **Economy of Permanence** - J C Kumarappa
8. **Bharat Mein Angreji Raj** – Pandit Sunderlal
9. **Rediscovering India** - by Dharampal
10. **Hind Swaraj or Indian Home Rule** - by Mohandas K. Gandhi
11. **India Wins Freedom** - Maulana Abdul Kalam Azad
12. **Vivekananda** - Romain Rolland (English)
13. **Gandhi** - Romain Rolland (English)

Web References:

1. <https://www.uhv.org.in/uhive>

2. <https://fdp-si.aicte-india.org/UHVIL.php>
3. https://www.aicte-india.org/sites/default/files/Model_Curriculum/Minor%20Degree%20in%20UHV.pdf

Mode of Conduct:

1. Lecture hours are to be used for interactive discussion, where proposals about the topics are to be made and students are to be motivated to reflect, explore, and verify them.
2. Tutorial hours are to be used for practice sessions.
3. While analyzing and discussing the topic, the faculty mentor's role is to point to essential elements to help sort them out from the surface elements. In other words, it allows the students to explore the critical components.
4. In the discussions, particularly during practice sessions (tutorials), the mentor encourages the student to connect with oneself and do self-observation, self-reflection, and self-exploration.
5. Scenarios may be used to initiate discussion. The student is encouraged to take up "ordinary" rather than "extraordinary" situations. Such observations and their analyses are shared and discussed with other students and faculty mentors in a group setting.
6. Tutorials (experiments or practical) are essential for the course. The difference is that the laboratory is part of everyday life; it is helpful in how you behave and work in real life. Worksheets, home assignments, and/or activities are included depending on the nature of the topics.
7. The practice sessions (tutorials) would also support student in performing actions commensurate to their beliefs. It is intended that this would lead to the development of commitment, namely behaving and working based on fundamental human values.
8. It is recommended that this content be placed before the student as it is, in the form of an introductory foundation course, without including anything else or excluding any part of this content. Additional content maybe offered in separate, higher courses. This course is to be taught by faculty from every teaching department, not exclusively by any one department.

Internal Assessment Pattern

Cognitive Level	Internal Assessment # 1 (%)	Internal Assessment # 2 (%)
L1	15	-
L2	34	-
L3	34	15
L4	17	34
L5	-	34
L6	-	17
Total (%)	100	100

Sample Short and Long Answers to Questions of Various Cognitive Levels

L1: Remember

1. What is meant by Natural Acceptance?
2. Define Right Understanding.
3. List the four orders of nature.
4. What are the basic human aspirations?
5. State the meaning of Sanyam (self-regulation).
6. What is meant by prosperity?
7. Name the two parts of a human being.
8. What is the role of the body according to Value Education?
9. List the feelings essential in human-to-human relationships.
10. What is meant by co-existence?

L2: Understand

1. Explain the relationship between Right Understanding and Relationship.
2. Describe the harmony between the self and the body.
3. Interpret the concept of continuous happiness and prosperity.
4. Discuss the difference between the needs of the self and the body.
5. Explain the meaning of justice in relationships.
6. Describe the importance of self-exploration.
7. Clarify the concept of mutual fulfillment in nature.
8. Discuss how education helps in holistic development.
9. Explain the vision of a universal human order.
10. Describe what is meant by 'respect' as the right evaluation.

L3: Apply

1. Apply the concept of Natural Acceptance in making ethical decisions.
2. Illustrate harmony in the family with an example from your own life.
3. Demonstrate how you can practice self-regulation in your daily routine.
4. Use the idea of mutual fulfillment in a plan for an eco-friendly campus.
5. Apply the idea of Right Understanding to resolve a conflict.
6. Show how holistic development can be integrated into your lifestyle.
7. Implement practices that promote harmony in your social group.
8. Use the concept of the body as an instrument to improve physical health.
9. Plan a group activity that promotes trust and respect.
10. Apply the principles of co-existence in managing natural resources.

L4: Analysing

1. Differentiate between happiness and prosperity with examples.
2. Analyze the role of the self and body in human behavior.
3. Compare trust and respect in family relationships.
4. Examine how lack of Right Understanding leads to disharmony.
5. Distinguish between self-regulation and external regulation.
6. Investigate the root causes of conflict in society.
7. Analyze the systems in nature that demonstrate harmony.
8. Examine how justice arises from mutual fulfillment.

9. Break down the structure of a universal human order.
10. Explore the interconnection between individual values and societal peace.

L5: Evaluating

1. Evaluate the relevance of Value Education in modern society.
2. Assess the impact of materialism on human happiness.
3. Judge the effectiveness of education in achieving holistic development.
4. Critique current societal systems in light of natural co-existence.
5. Appraise the role of mutual trust in building strong relationships.
6. Defend the need for harmony at all levels of human existence.
7. Evaluate the importance of distinguishing between self and body.
8. Justify the inclusion of Value Education in professional courses.
9. Compare the traditional vs. modern views on personal success.
10. Argue whether self-exploration is sufficient for understanding human values.

L6: Create and Develop

1. Design a value-based weekly routine for students.
2. Construct a model society based on mutual trust and justice.
3. Formulate strategies to integrate value education into school curricula.
4. Develop a plan for promoting harmony in a multicultural community.
5. Compose an essay on the role of harmony in personal and professional life.
6. Create a campaign to raise awareness on sustainable coexistence with nature.
7. Design a self-assessment tool for personal value alignment.
8. Develop a short workshop on self-exploration for college students.
9. Write a story that illustrates the principles of Right Understanding.
10. Create a visual mind map connecting harmony at the individual, family, and societal levels.

**Chairperson
Board of Studies (MBA)**

R24CSPC05**Software Engineering and Project Management****3 0 0 3**

(Computer Science and Engineering)

Course Objectives:

1. To comprehend software development life cycle.
2. To gain knowledge of requirement engineering and SRS documents.
3. To understand software architecture styles.
4. To learn various software testing techniques and their applicability.
5. To apply and analyze project management life cycle.

Course Outcomes

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs								DoK
		PO1	PO2	PO3	PO4	PO5	PO11	PSO1	PSO2	
R24CSPC05-01	Describe the principles of software engineering, life cycle models	3	2	2	-	1	1	2	-	L1,L2
R24CSPC05-02	Analyze the computing requirements to solve a given problem	2	3	2	2	2	1	3	-	L1,L2,L3
R24CSPC05-03	Demonstrate the importance of software modeling and modeling languages	2	2	3	2	3	1	2	-	L2,L3,L4
R24CSPC05-04	Illustrate the necessity of software testing and design test cases for a software	2	3	2	3	3	1	3	-	L4,L5
R24CSPC05-05	Interpret Software maintenance and state the concepts of project Management.	2	2	2	1	2	2	-	2	L4,L5,L6

SYLLABUS**UNIT-I:****10 Hours**

Introduction: Evolution, Software development projects, Exploratory style of software developments, Emergence of software engineering, Software Product and Process Characteristics, Software Process customization and improvement, Computer system engineering.

Software Life Cycle Models: Basic concepts, Waterfall model and its extensions, Rapid application development, agile development model, Spiral model.

CO's-CO1**Self-Learning Topics:** Advanced Software Development Paradigms**UNIT II:****15 Hours**

Requirements Analysis and Specification: Requirements gathering and analysis, Software

Requirements Specification (SRS), Formal system specification, Axiomatic specification, Algebraic specification, Executable specification and 4GL, Petri Nets- Data Dictionary. **CO's–CO2**

Self-Learning Topics: Modern Requirements Engineering Approaches

UNIT – III: **15 Hours**

Design process and Design Concepts, Design Model– Design Heuristic, Architectural Design - Architectural styles, Architectural Design, Architectural Mapping using Data Flow- User Interface, Design: Interface analysis, Interface Design, Component level Design: Designing Class based components, traditional Components) **CO's–CO3**

Self-Learning Topics: Design Patterns (Gang of Four - GoF)

UNIT- IV: **10 Hours**

Coding And Testing: Coding, Code review, Software documentation, Software testing fundamentals, Internal and external views of Testing, Black-box testing, White-Box testing, , Debugging, Software **Implementation Techniques:** Coding practices- Refactoring, Maintenance and Reengineering-BPR model , Reengineering process model-Reverse and Forward Engineering, Program analysis tools, Unit Testing – Integration Testing – Validation Testing Q, Testing object-oriented programs, Smoke testing, and Some general issues associated with testing. . **CO's–CO4**

Self-Learning Topics: Test Driven Development (TDD) and Behavior Driven Development (BDD)

UNIT-V: **10 Hours**

Software Maintenance & Software Project Measurement: Software Configuration Management (SCM), Software Change Management, Version Control, Change control and Reporting, Re-engineering, Reverse Engineering, Project Management Concepts, Project Scheduling and Tracking, Software Quality Assurance (SQA)

Software Reuse: reuse- definition, introduction, reason behind no reuse so far, Basic issues in any reuse program, a reuse approach, and Reuse at organization level. **CO's–CO5**

Self-Learning Topics: Adaptive and Preventive Maintenance Strategies

Board of Studies : Computer Science and Engineering

Approved in BoS No : 02, 9th May, 2025

Approved in ACM No: 02

Expert Talk (To be Delivered by SMEs from Industries) **CO's/ PO's / PSO's**

1. Agile Software Development in the Real World: Challenges and Best Practices" Industry adoption of Agile, Scrum, and Dev Ops with practical case studies. **CO1–CO5, PO1,PO2,PO3,PO12,PSO1,PSO2**
2. Software Architecture and Design Patterns for Scalable Applications" Focus on modern design principles used in scalable and maintainable software **CO1–CO5, PO1, PO2,PO3,PO12,PSO1,PSO2**

Text Books:

1. Fundamentals of Software Engineering, Rajib Mall, 5th Edition, PHI.
2. Software Engineering a Practitioner's Approach, Roger S. Pressman, 9th Edition, Mc Graw Hill International Edition.

Reference Books:

1. Software Engineering, Ian Sommerville, 10th Edition, Pearson.
2. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.
3. Software Project Management” by Bob Hughes, Mike Cotterell, and Rajib Mall.
4. Managing Software Projects” by Gopalaswamy Ramesh.

Web References:

1. <https://www.computer.org/education/bodies-of-knowledge/software-engineering>
2. <https://www.geeksforgeeks.org/software-engineering/>
3. <https://www.pmi.org/pmbok-guide-standards>
4. https://www.mindtools.com/pages/main/newMN_PPM.htm

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35%	--
L2	40%	--
L3	25%	25%
L4	--	35%
L5	--	25%
L6	--	15%
Total (%)	100%	100%

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. What is the purpose of software engineering?
2. Define the term "software life cycle."
3. List the characteristics of a software product.
4. Name any four software life cycle models.
5. What is the waterfall model?
6. What is the purpose of a Software Requirements Specification (SRS)?
7. Define formal system specification.
8. What is a data dictionary?
9. List the types of formal specification techniques.
10. What are Petri Nets used for in system modeling?

L2: Understand

1. Explain the difference between software product and software process.
2. Describe how exploratory software development differs from planned development.

3. Compare traditional software development with agile development.
4. Explain why customization of the software process is important.
5. Describe the evolution of software engineering.
6. Explain the role of requirements gathering in software development.
7. Describe the characteristics of a good SRS document.
8. Differentiate between axiomatic and algebraic specifications.
9. Explain how Petri Nets represent system behavior.
10. Describe the use of 4GLs in executable specifications.
11. Explain the importance of architectural design in software engineering.
12. Describe the purpose of design heuristics in the design process.
13. Discuss the role of data flow diagrams in architectural mapping.
14. Differentiate between class-based and traditional components.
15. Explain the concept of interface analysis in UI design.

L3: Apply

1. Write a simple example of an algebraic specification for a stack ADT.
2. Apply Petri Nets to model a simple producer-consumer system.
3. Develop a basic data dictionary for a student registration system.
4. Create an SRS for a library management system.
5. Use axiomatic specification to define a simple banking operation.
6. Apply design heuristics to improve an existing system design.
7. Use a data flow diagram to map the architecture of a student management system.
8. Develop a simple interface design for an online bookstore.
9. Design class-based components for a ride-sharing application.
10. Implement a basic architectural model using a layered style.

L4: Analysing

1. Analyze the relationship between software architecture and system performance.
2. Compare and contrast different architectural styles (e.g., layered vs. client-server).
3. Identify the strengths and weaknesses of traditional components in modern software systems.
4. Break down the steps involved in interface design for a mobile application.
5. Examine the impact of poor design heuristics on software maintainability.
6. Analyze the impact of poor documentation on software maintenance.
7. Compare white-box testing and black-box testing with respect to coverage.
8. Differentiate between reverse engineering and reengineering.
9. Examine how object-oriented features affect testing strategies.

10. Identify and explain common issues associated with testing large-scale systems.
11. Analyze the relationship between software maintenance and configuration management.
12. Compare re-engineering and reverse engineering processes.
13. Differentiate between corrective, adaptive, and perfective maintenance.
14. Identify the barriers to software reuse in large organizations.
15. Examine the role of SQA in reducing post-deployment defects.

L5: Evaluating

1. Evaluate the effectiveness of smoke testing in agile environments.
2. Justify the use of program analysis tools in improving code quality.
3. Assess the benefits of incorporating refactoring in continuous integration.
4. Critically evaluate various testing strategies for object-oriented software.
5. Review and recommend practices for effective debugging in real-time applications.
6. Evaluate the effectiveness of SCM tools in large-scale software projects.
7. Justify the need for strict change control in mission-critical applications.
8. Assess the impact of poor version control on project success.
9. Critique the current industry practices on software reuse.
10. Evaluate a project tracking system in terms of its ability to meet deadlines and budget.

L6: Creating

1. Design a comprehensive test plan for an e-commerce application.
2. Create a code review checklist tailored for a distributed development team.
3. Develop a test suite for validating a banking application using black-box techniques.
4. Construct a model that integrates reverse engineering and forward engineering in legacy system migration.
5. Propose a set of guidelines for testing object-oriented programs effectively.
6. Design a version control and change management strategy for a distributed team.
7. Develop an SQA plan for a medical software application.
8. Create a reuse strategy for an educational software company.
9. Construct a project management schedule using available software tools for a mid-sized web application.
10. Propose an organization-wide reuse program, addressing its key challenges and success factors.

**Chairperson
Board of Studies (CSE)**

R24CSPC06	Object Oriented Programming using JAVA Lab	0	0	3	1.5
(Common to CSE, CSE (DS) & CSE (AI&ML))					

Course Objectives:

1. To develop skills to design and analyze the applications with respect to java programming.
2. To strengthen the ability to identify and apply the suitable object oriented concept for the given real world problem.

Course Outcomes

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

Course Code	Course Outcomes	Mapping with POs and PSOs						Dok
		PO1	PO2	PO3	PO5	PS01	PS02	
R24CSPC06.1	Write basic Java programs using arrays, control structures, and demonstrate the use of classes and objects.	3	2	2	1	3	2	L1, L3
R24CSPC06.2	Apply object-oriented concepts such as inheritance and packages, and implement applications using built-in and user-defined packages.	2	2	3	3	2	3	L3, L4
R24CSPC06. 3	Develop Java applications with exception handling, multithreading, graphical user interfaces, and basic networking.	2	2	3	3	1	3	L3, L4

Board of Studies: Computer Science and Engineering

Approved in BOS No: 02, 9th May, 2025

Approved in ACM No: 02

Developing the following programs:

1. a) Implement the following programs using command line arguments and Scanner class
 - i) Accept two strings from the user and print it on console with concatenation of “and” in the middle of the strings. **CO’s-CO1**
 - ii) To find the perimeter and area of a circle given a value of radius. **CO’s-CO1**
- b) Write a program using classes and objects in java? **CO’s-CO1**
2. a) Write a program to call default constructor first and then any other constructor in the class? **CO’s-CO1**
- b) Write a program that accepts an array of integers and print those which are both odd and prime. If no such element in that array print “Not found”. **CO’s-CO1**

- c) Write a program to accept contents into an Integer Array and print the frequency of each number in the order of their number of occurrences. **CO's-CO1**
- d) Write a program that accepts an 'm x n' double dimension array, where 'm' represents financial years and 'n' represents Ids of the items sold. Each element in the array represents number of items sold in a particular year. Identify the year and id of the item which has more demand. **CO's-CO2**
3. a) Create a class Box that uses a parameterized constructor to initialize the dimensions of a box. The dimensions of the Box are width, height, depth. The class should have a method that can return the volume of the box. Create an object of the Box class and test the functionalities. **CO's-CO2**
- b) Create a new class called Calculator with the following methods:
A static method called power Int(int num1,int num2) This method should return num1 to the power num2.
A static method called power Double (double num1,double num2). This method should return num1 to the power num2.
Invoke both the methods and test the functionality. Also count the number of objects created. **CO's-CO2**
4. a) Accept a String and a number 'n' from user. Divide the given string into substrings each of size 'n' and sort them lexicographically. **CO's-CO1**
- b) Accept an array of strings and display the number of vowels and consonants occurred in each string. **CO's-CO1**
- c) Accept two strings from the user and determine if the strings are anagrams or not. **CO's-CO1**
5. a) Create a multilevel inheritance for classes vehicle, brand and cost. The vehicle class determines the type of vehicle which is inherited by the class brand which determines the brand of the vehicle. Brand class is inherited by cost class, which tells about the cost of the vehicle. Create another class which calls the constructor of cost class and method that displays the total vehicle information from the attributes available in the super classes. **CO's-CO2**
- b) Create an inheritance hierarchy of Figure_3D, Cylinder, Cone, Sphere etc. In the base class provides methods that are common to all Figure_3Ds and override these in the derived classes to perform different behaviors, depending on the specific type of Figure_3D. Create an array of Figure_3D, fill it with different specific types of Figure_3Ds and call your base class methods. **CO's-CO2**
6. a) Design a package to contain the class Student that contains data members such as name,

roll number and another package contains the interface Sports which contains some sports information. Import these two packages in a package called Report which process both Student and Sport and give the report. **CO's-CO2**

b) Write a program that accepts values of different data types and convert them to corresponding wrapper classes and display using the vector. **CO's-CO1**

7. a) Write a program to generate a set of random numbers between two numbers x1 and x2, and x1>0. **CO's-CO1**

b) Write a program to implement a new Array List class. It should contain add(), get(),remove(), size() methods. Use dynamic array logic. **CO's-CO2**

c) Create an employee class containing at least 3 details along with Id, setters, and getters. Insert the employee objects dynamically key as employee id and value as its corresponding object into a HashMap. Perform Id based search operation on the HashMap. **CO's-CO2**

8. a) Write a program that reads file name from the user then displays information about that file, also read the contents from the file in byte stream to count the number of alphabets, numeric values, and special symbols. Write these statistics into another file using byte streams **CO's-CO3**

b) Write a program that reads a CSV file containing a super market data containing product ID, Name, Cost and Quantity of sales and calculate the total revenue of the supermarket also sort the products in the order of their demand. **CO's-CO3**

c) Write a program that reads a text file containing some technical content and identify the technical terms and sort them alphabetically. **CO's-CO3**

Note: use a file containing stop words (general English and Grammar terms as many as possible)

9. a) Write a program that reads two numbers from the user to perform integer division into Num1 and Num2 variables. The division of Num1 and Num2 is displayed if they are integers. If Num1 or Num2 were not an integer, the program would throw a Number Format Exception. If Num2 were Zero, the program would throw an Arithmetic Exception. **CO's-CO3**

b) Create a user defined exception. **CO's-CO3**

10. a) Write a program that creates 3 threads by extending the Thread class. First thread displays “Good Morning” every 1 sec, the second thread displays “Hello” every 2seconds and the third displays Welcome” every 3 seconds. (Repeat the same by implementing Runnable). **CO's-CO3**

b) Write a program to illustrate Thread synchronization. **CO's-CO3**

11. a) Create a JApplet that displays a message which is scrolling from left to right. **CO's-CO3**

- b) Write a program that displays a sample registration page using Swing controls use appropriate layout managers. **CO's-CO3**
- c) Write a program for handling mouse events with adapter classes. **CO's-CO3**
12. a) Create an interface containing 3 radio buttons named line, rectangle, and oval. Based on the radio button selected, allow user to draw lines, rectangles, or ovals as per the locations selected by the user. **CO's-CO3**
- b) Write a program to create a Table inside a JFrame. **CO's-CO3**
- c) Create an interface that illustrates JFile Chooser class and read CSV file containing employee data of various departments and display the records department wise on the interface. **CO's-CO3**
13. a) Check all the fields filled or not, display success dialogue if all fields are filled with the help of Action Listener for program **CO's-CO3**
- b) Display respective error dialogue if a field is empty. **CO's-CO3**
14. Write a program to create three JSliders where each represents colors RED, GREEN and BLUE. Each slider has a value from 0 to 255. The background color of the applet is set based on the values retrieved from each slider to form a color using the color class constructor. On sliding any slider, the background color of the applet changes. **CO's-CO3**
15. Complete the code to develop an ADVANCED CALCULATOR that emulates all the functions of the GUI Calculator as shown in the image. **CO's-CO3**
16. Write a program that implements a simple client/server application. The client sends data to a server. The server receives the data, uses it to produce a result, and then sends the result back to the client. The client displays the result on the console.
- For ex: The data sent from the client is the radius of a circle, and the result produced by the server is the area of the circle. **CO's-CO3**

Case Studies:

1. Grading a Multiple-choice Test for students **CO's-CO1**
2. Create a Person class containing basic details like Name, Gender, Mobile and Email. Based on that create and manage the objects that are related to student and employee Classes **CO's-CO2**
3. Create a package called Banking containing classes and interfaces related to various banking operations such as withdrawal, deposits, loans and insurance etc. Create two classes related to any two specific banks that uses this package. **CO's-CO2**
4. Write a program that implements simple chat application using GUI. **CO's-CO3**

Textbooks:

1. Herbert Schildt, Java The complete reference, 11th Edition, McGrawHill, 2019
2. Timothy budd, An introduction to object-oriented programming, 3rd Edition, Pearson Education, 2009.

Reference Books:

1. Cay S. Horstmann, Core Java Volume I–Fundamentals, 11th Edition, Pearson 2019
2. Y. Daniel Liang Introduction to Java Programming Comprehensive Version, 10th Edition, Pearson, 2015.
3. Bruce Eckel, Thinking in Java, 4th Edition, Prentice Hall, 2006

Web References:

1. https://onlinecourses.nptel.ac.in/noc19_cs84/preview

**Chairperson
Board of Studies (CSE)**

R24ES12**Design and Analysis of Algorithms Lab****0 0 3 1.5**

(Common to CSE, CSE (DS) & CSE (AI&ML))

Course Objectives:

The course aims to give students hands – on experience and train them to learn Design and Analysis Algorithm

1. To learn Prolog Program.
2. To Implement in prolog and its working environment
3. To Implement N-Queen problem and puzzle problem using Prolog
4. To Analyze the problem using BFS and DFS algorithm

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

Course Code	Course Outcomes	Mapping with POs and PSOs							Dok
		PO 1	PO 2	PO 3	PO 5	PS O1	PS O2	PS O3	
R24ES12.1	Understand of algorithm design techniques applicable across various tasks and ensuring foundational knowledge in problem-solving.	3	3	3	3	3	3	3	L3
R24ES12.2	Implement of the algorithms in both sequential and parallel paradigms, relevant for weeks involving sorting and graph traversals.	3	3	3	1	3	3	3	L4
R24ES12.3	Handle real-world computational problems.	3	3	3	1	3	3	3	L5

Board of Studies: Computer Science and Engineering

Approved in BOS No: 02, 9th May, 2025

Approved in ACM No: 02

Developing the following programs**Week 1:**

1. Sort a given set of elements using the Quick sort method and analyze time. **CO's-CO1**

Week 2:

2. Implement a parallelized Merge Sort algorithm using Open MPI **CO's-CO1**

Week 3:

3. Implement binary tree traversal techniques using recursion and non-recursion. **CO's-CO1**

Week 4:

4. Print reachable nodes in a digraph using BFS; check connectivity using DFS. **CO's-CO2**

Week 5:

5. Determine articulation points and biconnected components in a graph. **CO's-CO2**

Week 6:

6. Implement algorithms for minimum cost spanning trees (Prim's and Kruskal's). **CO's-CO2**

Week 7:

7. Find shortest paths in a graph using Dijkstra's algorithm; study of Prolog. **CO's-CO2**

Week 8:

8. Write simple fact for the statements using prolog **CO's-CO2**

Week 9:

9. Write predicates one convert's centigrade temperature to Fahrenheit, other checks if a temperature is below freezing. **CO's-CO2**

Week 10:

10. Write a program to solve 4-Queen problem. **CO's-CO3**

Week 11:

11. Write a program to solve 8-puzzle problem. **CO's-CO3**

Week 12:

12. Write a program to solve any problem using Breadth First Search. **CO's-CO3**

Week 13:

13. Write a program to solve any problem Depth First Search **CO's-CO3**

Week 14:

14. Write a program to solve Travelling salesman Problem **CO's-CO3**

Week 15 and Week 16:

15. Develop a tool to analyze social networks using graph algorithms (e.g., finding the shortest path, clustering, and connectivity). Input social network data and visualize the connections. **CO's-CO3**

Text Books:

1. Anany Levitin, "Introduction to the Design and Analysis of Algorithms", Third Edition, Pearson Education, 2012.
2. Ellis Horowitz, Sartaj Sahni and Sanguthevar Rajasekaran, "Computer Algorithms/ C++", Second Edition, Universities Press, 2007

Reference Books:

1. Thomas H.Cormen, Charles E.Leiserson, Ronald L. Rivest and CliffordStein, "Introduction to Algorithms", Third Edition, PHI Learning PrivateLimited, 2012.
2. S. Sridhar, "Design and Analysis of Algorithms", Oxford university press,2014

Web References:

1. <https://nptel.ac.in/courses/106101060>
2. https://www.cse.iitm.ac.in/course_details.php?arg=OTI
3. https://swayam.gov.in/nd1_noc19_cs47/previ

**Chairperson
Board of Studies (CSE)**

R24C SSC01

Python Programming

0 1 2 2

(Common to CSE, ECE)

Course Objectives:

- Introduce core-programming concepts of Python programming language.
- Demonstrate about Python data structures like Lists, Tuples, Sets and dictionaries
- Implement Functions, Modules and Regular Expressions in Python Programming and to create practical and contemporary applications using these

Course Outcomes

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

Course Code	Course Outcomes	Mapping with POs and PSOs			
		PO1	PO2	PO3	Dok
R24C SSC01.1	Introduce core-programming concepts of Python programming language.	3	3	3	L1
R24C SSC01.2	Develop, run and manipulate Python programs using Functions, Core data structures like Lists, Dictionaries, and use of Strings Handling methods	3	3	3	L2
R24C SSC01.3	Demonstrate about Python data structures like Tuples, Sets and dictionaries	3	3	3	L2
R24C SSC01.4	Develop, run and manipulate Python programs using File Operations and concepts of object-oriented programming	3	3	3	L3
R24C SSC01.5	Understand Data Science, Numpy, Pandas and working with XML, JSON and other file formats.	3	3	3	L3

Board of Studies : Computer Science and Engineering

Approved in BOS No: 02, 9th May, 2025

Approved in ACM No: 02

Developing the following programs:**UNIT-I:****CO's:CO1**

History of Python Programming Language, Thrust Areas of Python, Installing Anaconda Python Distribution, Installing and Using Jupyter Notebook, Features, Limitations, advantages, and applications of python.

Parts of Python Programming Language: Identifiers, Keywords, Statements and Expressions, Variables, Operators, Precedence and associativity, Data Types, Indentation, Comments, Reading Input, Print Output, Type Conversions, the type () Function and Is Operator, Dynamic and Strongly Typed Language.

Control Flow Statements: if statement, if-else statement, if...elif...else, Nested if statement, while Loop, for Loop, continue and break Statements, Catching Exceptions Using try and except Statement.

Sample Experiments:

1. Write a program to find the largest element among three Numbers.
2. Write a Program to display all prime numbers within an interval
3. Write a program to swap two numbers without using a temporary variable.
4. Demonstrate the following Operators in Python with suitable examples.
 - i. Arithmetic Operators
 - ii. Relational Operators
 - iii. Assignment Operators
 - iv. Logical Operators
 - v. Bit wise Operators
 - vi. Ternary Operator
 - vii. Membership Operators
 - viii. Identity Operators
5. Write a program to add and multiply complex numbers
6. Write a program to print multiplication table of a given number.
7. Write how to Handle specific exceptions – like division of a number by zero

UNIT-II:

CO's:CO2

Functions: Built-In Functions, Commonly Used Modules, Function Definition and Calling the function, return Statement and void Function, Scope and Lifetime of Variables, Default Parameters, Keyword Arguments, *args and **kwargs, Command Line Arguments, Lambda Functions: Syntax and usage of lambda functions..

Strings: Creating and Storing Strings, Basic String Operations, Accessing Characters in String by Index Number, String Slicing and Joining, String Methods, Formatting Strings, Regular Expressions (Regex)

Lists: Creating Lists, Basic List Operations, Indexing and Slicing in Lists, Built-In Functions Used on Lists, List Methods, del Statement.

Sample Experiments:

1. Write a Python program to square each element in a list using a lambda function.
2. Write a program to define a function with multiple return values.
3. Write a program to define a function using default arguments.
4. Write a program to find the length of the string without using any library functions.
5. Write a program to check if the substring is present in a given string or not.

6. Write a program to perform the given operations on a list:
 - i. Addition
 - ii. Insertion
 - iii. Slicing
7. Write a program to perform any 5 built-in functions by taking any list.

UNIT-III:**CO's:CO3**

Dictionaries: Creating Dictionary, Accessing and Modifying key:value Pairs in Dictionaries, Built-In Functions Used on Dictionaries, Dictionary Methods, del Statement. Tuples and Sets: Creating Tuples, Basic Tuple Operations, tuple() Function, Indexing and Slicing in Tuples, Built-In Functions Used on Tuples, Relation between Tuples and Lists, Relation between Tuples and Dictionaries, Using zip() Function, Sets, Set Methods, Frozenset, Defaultdict and OrderedDict (from collections).

Sample Experiments:

1. Write a program to create tuples (name, age, address, college) for at least two members and concatenate the tuples and print the concatenated tuples.
2. Write a program to count the number of vowels in a string (No control flow allowed).
3. Write a program to check if a given key exists in a dictionary or not.
4. Write a program to add a new key-value pair to an existing dictionary.
5. Write a program to sum all the items in a given dictionary.
6. Write a program using defaultdict to count the frequency of each fruit in the given list: ['apple', 'banana', 'apple', 'orange', 'banana', 'apple']

UNIT-IV:**CO's:CO4**

Files: Types of Files, Creating and Reading Text Data, File Methods to Read and Write Data, Reading and Writing Binary Files, Pickle Module, Reading and Writing CSV Files, Python os and os.path Modules.

Object-Oriented Programming: Classes and Objects, Creating Classes in Python, Creating Objects in Python, Constructor Method, Classes with Multiple Objects, Class Attributes Vs Data Attributes, Encapsulation, Inheritance, Polymorphism.

Sample Experiments:

1. Write a program to sort words in a file and put them in another file. The output file should have only lower-case words, so any upper-case words from source must be lowered.
2. Python program to print each line of a file in reverse order.
3. Python program to compute the number of characters, words and lines in a file.
4. Write a program to create, display, append, insert and reverse the order of the items in the array.

5. Write a program to add, transpose and multiply two matrices.
6. Write a Python program to create a class that represents a shape. Include methods to calculate its area and perimeter. Implement subclasses for different shapes like circle, triangle, and square.

UNIT-V:

CO's:CO5

Introduction to Data Science: Functional Programming, JSON and XML in Python, NumPy with Python, Pandas.

Board of Studies: Computer Science and Engineering

Approved in BoS No: 02,--April, 2025

Approved in ACM No: 02

Sample Experiments:

1. Python program to check whether a JSON string contains complex object or not.
2. Python Program to demonstrate NumPy arrays creation using array () function.
3. Python program to demonstrate use of ndim, shape, size, dtype.
4. Python program to demonstrate basic slicing, integer and Boolean indexing.
5. Python program to find min, max, sum, cumulative sum of array
6. Create a dictionary with at least five keys and each key represent value as a list where this list contains at least ten values and convert this dictionary as a pandas data frame and explore the data through the data frame as follows:
 - a) Apply head () function to the pandas data frame
 - b) Perform various data selection operations on Data Frame
7. Select any two columns from the above data frame, and observe the change in one attribute with respect to other attribute with scatter and plot operations in matplotlib

Reference Books:

1. Gowri shankar S, Veena A., Introduction to Python Programming, CRC Press.
2. Python Programming, S Sridhar, J Indumathi, V M Hariharan, 2nd Edition, Pearson, 2024
3. Introduction to Programming Using Python, Y. Daniel Liang, Pearson

Online Learning Resources Links:

1. <https://www.coursera.org/learn/python-for-applied-data-science-ai>
2. <https://www.coursera.org/learn/python?specialization=python#syllabus>

Web References:

1. <https://www.geeksforgeeks.org/python-programming-language-tutorial/>
2. <https://www.python.org/about/gettingstarted/>
3. <https://www.w3schools.com/python/>

Sample Short and Long Answers questions of Various Cognitive Levels**L1: Remember**

1. Define identifiers and give two examples.
2. List any three features of Python.
3. What is the use of the type() function.
4. Define a function. What is a lambda function?
5. What are default and keyword arguments?
6. List any four string methods in Python.
7. Define a dictionary. How is it different from a list?
8. What is a frozenset?
9. What is the syntax of zip()?
10. What is a constructor in Python?
11. List any three file handling functions.
12. Define encapsulation.
13. What is a NumPy array?
14. Define a DataFrame in pandas.
15. Mention any two data visualization functions from matplotlib.

L2: Understand

1. Explain the significance of indentation in Python.
2. Differentiate between is and == with examples.
3. Explain the flow of execution for the if...elif...else structure.
4. Differentiate between *args and **kwargs.
5. Explain string slicing and joining with examples.
6. Explain the scope and lifetime of a variable in Python
7. Explain the relation between tuples and dictionaries.
8. Describe the purpose of defaultdict and OrderedDict.
9. Differentiate between class and object.
10. Explain the difference between class attributes and instance attributes.
11. Explain the difference between .ndim, .shape, and .size in NumPy.
12. Describe the process of reading JSON data in Python.

L3: Apply

1. Write a Python program to swap two variables without a temporary variable.
2. Write a program to print the largest of three numbers using conditional statements.
3. Write a Python program using a lambda function to square each element of a list.
4. Write a function that returns both sum and product of two numbers.

5. Write a program to check if a substring exists in a string.
6. Write a program to count vowels in a string using only string and set operations.
7. Write a program to check if a given key exists in a dictionary.
8. Write a program to concatenate two tuples.
9. Write a Python program to read from a file and print its contents in reverse line order.
10. Write a class representing a rectangle with methods to calculate area and perimeter.
11. Write a Python program to create a NumPy array and perform slicing.
12. Create a dictionary with lists and convert it into a pandas DataFrame.

L4: Analysing

1. Compare the use of for and while loops with scenarios.
2. Analyze how operator precedence affects the outcome of expressions in Python.
3. Analyze how positional arguments differ from keyword arguments in function calls.
4. Compare string formatting methods: %, .format(), and f-strings.
5. Compare the use of sets and dictionaries in Python.
6. Analyze the output when applying slicing on a tuple with an example.
7. Compare inheritance and polymorphism with suitable class examples.
8. Analyze the structure and behavior of a file object during read and write operations.
9. Analyze how data is structured differently in NumPy and pandas.
10. Compare JSON and XML data formats with Python parsing tools.

L5: Evaluating

1. Evaluate the use of dynamic typing in Python. What are its pros and cons?
2. Assess the effectiveness of Anaconda and Jupyter Notebook for beginner programmers
3. Evaluate the usefulness of using regular expressions for string validation.
4. Justify the use of built-in functions over writing manual logic
5. Evaluate the use of immutable data types (tuples, frozensets) in large applications.
6. Assess the performance difference between regular dict and defaultdict when counting frequencies.
7. Evaluate the usefulness of OOP in Python compared to procedural programming.
8. Judge whether using binary files or text files is better for storing structured data.
9. Evaluate the benefits of using pandas over lists/dictionaries for data manipulation.
10. Assess the effectiveness of functional programming in handling large-scale data pipelines.

**Chairperson
Board of Studies (CSE)**

R24HS04

Logical Reasoning and Corporate Skills
(Common to all UG programmes)

0 0 2 1

Course Objectives:

1. **Develop Logical thinking skills:** Students will be able to analyze information, identify patterns, and make informed decisions.
2. **Enhance analytical skills:** Students will be able to break down complex information into its component parts, analyze each part, and draw conclusions.
3. **Understand corporate culture and etiquette:** Students will be able to understand and demonstrate corporate culture and etiquette, including professional behaviour and networking.
4. **Develop effective communication skills:** Students will be able to communicate effectively in a corporate setting, including verbal and written communication.
5. **Develop negotiation and conflict resolution skills:** Students will be able to negotiate effectively and resolve conflicts in a corporate setting.
6. **Improve teamwork and collaboration skills:** Students will be able to work effectively in teams, build strong relationships, and manage conflicts.

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

Course Code	Course Outcome	PO1	PO7	PO9	P011	Dok
R24HS04.1	Use their logical thinking and analytical abilities to solve reasoning questions from number analogy and series and letter and coding and decoding based aptitude questions company specific and other competitive tests.	2	---	---	---	L4,L5
R24HS04.2	Solve questions related to Blood Relations clock and calendar, etc. From company specific and other competitive tests.	1	---	---	---	L4,L5
R24HS04.3	Enforce corporate etiquette, and precise usage of English grammar to enhance their professional communication.	---	2	3	1	L1,L3
R24HS04.4	Master negotiation skills and telephone etiquette with emotional intelligence for corporate interactions.	---	2	3	1	L3
R24HS04.5	Enhance email writing skills by incorporating vocabulary acquired from storytelling, situational dialogues and reading activities by using various digital tools.	2	---	3	1	L2,L3

SYLLABUS**UNIT I****15 Hours**

Number Series, Letter Series, Number analogy, letter analogy, word analogy. Coding Decoding-Letter to letter, letter to digit, letter to number and symbol, Word to word coding, odd man out. Directions-Finding distance, Direction and Shadow based problem, Blood Relations-Mixed Blood Relations, Puzzle-Based Blood Relation, Single-Person Blood Relation, Symbol based Blood Relations.

Self Learning Topic: Number Puzzles

UNIT II**15 Hours**

Clocks-finding Angle, Time, Mirror image, Faulty clock, Calendars – Finding day of the week, Number of odd days, Repetition of same calendar, Seating Arrangement-Circular arrangement, linear arrangement, Order Sequence and Ranking

Self Learning Topic: Arrangement puzzles

UNIT III**10 Hours**

Corporate Etiquette – Work Place Etiquette and Conflict Resolution – Grammar Revision, Leadership and Management skills. Verbal Ability: Prepositions, Articles, tenses and conjunction

Self Learning Topic: Successful Team Leadership

UNIT IV**10 Hours**

EQ – Negotiation Skills – Telephone Etiquette – MNCs Paper Model Introduction, Situational Dialogue Practice – Team Activities Related to Spoken

English Verbal Ability: Fill in the blanks (Based on the given appropriate words)

Self Learning Topic: Group Discussion Skills

UNIT V**10 Hours**

E Mail Writing-Vocabulary, Story Telling Activity, Functional English-IELTS Vocabulary News Paper Reading.

Verbal Ability: Sentence arrangements

Self Learning Topic: Sentence Construction

Board of Studies: MBA

Approved in BOS No: 02, 13th, April, 2025

Approved in ACM No: 02

Text Books:

1. A Modern Approach to Logical Reasoning R.S Aggarwal – S.chand publications.
2. Personality Development and Soft Skills by Barun K.mitra

Reference Books:

1. Shortcuts in Reasoning (Verbal, Non-Verbal, Analytical & Critical) for Competitive Exams Disha's publications
2. Communication by C.S.G Krishnamacharyulu & Lalitha Ramakrishnan-Himalaya publishing house

Web References:

1. <https://faceprep.in/logical> reasoning
2. www.sawaal.com
3. <http://nptel.ac.in/courses>
4. www.prepinsta.com

Model Questions of Logical Reasoning & Corporate Skills

L1: Remember

1. What is the proper way to greet a client or colleague in a professional setting?
2. The book is _____ the table.
3. I'm going _____ the store to buy some milk.
4. If I _____ (know) the answer, I would tell you.
5. By next year, I _____ (study) English for five years.
6. If it _____ (rain), we would have stayed home.
7. I _____ (finish) my homework before I went to bed.

L2: Understand

1. What is your favorite childhood story? Why do you enjoy it?
2. Tell a story about a memorable experience from your life.
3. If you could tell a story to inspire others, what would it be about?
4. Rearrange the following sentences to form a coherent paragraph:
 - A. The sun was setting over the ocean.
 - B. The sound of the waves was soothing.
 - C. We walked along the beach, enjoying the peaceful atmosphere.
 - D. The smell of saltwater filled the air.
5. Put the following sentences in the correct order to tell a story:
 - A. She packed her bags and said goodbye to her family.
 - B. After a few months, she returned home with new experiences and stories.
 - C. Emily had always wanted to travel abroad.
 - D. She boarded a plane and took off for a new adventure.
6. Rearrange the following sentences to form a logical argument:
 - A. Therefore, we should prioritize renewable energy sources.
 - B. Fossil fuels are a finite resource and contribute to climate change.
 - C. In conclusion, our reliance on fossil fuels is unsustainable.
 - D. Renewable energy sources, on the other hand, are sustainable and environmentally friendly.

L3: Apply

1. What qualities do you think are essential for a good leader?
2. How would you motivate a team member who is struggling with their work?
3. Describe a time when you had to make a difficult decision as a leader.
4. The new policy aims to _____ the gap between the rich and the poor.
5. The company's _____ in the market has increased significantly over the past year.
6. The teacher asked the students to _____ their essays before submitting them.
7. The _____ of the new smart phone is its advanced camera system.
8. The company will _____ a new product line next quarter.

9. By the time I arrived, they _____ (eat) dinner.

L4: Analyzing

1. Complete the series: 2, 4, 8, and 16?
2. Find the next number: 1, 2, 4, 7, 11,?
3. Identify the pattern: 3, 6, 9, 12, ?
4. Complete the sequence: 5, 10, 15, 20, ?
5. Determine the next number: 2, 6, 12, 20, ?
6. Identify the pattern: B, D, F, H, ?
7. Complete the sequence: M, O, Q, S, ?
8. Determine the next letter: T, S, R, Q, ?
9. Complete the series: Monday, Tuesday, Wednesday, ?
10. Find the next word: January, February, March, ?
11. Determine the next word: Book, Magazine, Newspaper, ?
12. Find the odd one out: Apple, Banana, Carrot, Mango
13. Identify the odd one out: Football, Basketball, Tennis, Piano
14. Determine the odd one out: Car, Bus, Train, Book
15. Find the odd one out: Red, Blue, Green, Chair
16. Identify the odd one out: Dog, Cat, Elephant, House
17. What is the angle between the hour and minute hands of a clock at 3:15?
18. 1. What is the mirror image of the time 3:45?
19. 2. A person sees their mirror image and notices that their watch shows 9:30. What is the actual time?

L5: Evaluating

1. Five friends - A, B, C, D, and E - are sitting in a row. If A is at one end, B is next to A, and C is in the middle, who is sitting at the other end?
2. Seven people - A, B, C, D, E, F, and G - are sitting in a circle. If A is between B and C, and D is opposite A, who is sitting next to D?
3. If today is Monday, what day of the week will it be 30 days from now?
4. A meeting is scheduled for the 15th of March, which falls on a Wednesday. What day of the week will the 22nd of March be?
5. If you are facing north and turn 90 degrees to your right, which direction are you facing?
6. A person walks 5 km east and then 3 km west. How far is he from his starting point?
7. If you are facing south and walk 2 km, then turn left and walk 1 km, which direction are you facing?
8. A is the brother of B. C is the sister of A. How is B related to C?
9. A woman introduces a man as the son of her brother's father. How is the man related to the woman?
10. A man points to a woman and says, "She is the daughter of my mother's only child." How is the woman related to the man?

Course Objectives:

1. To make the students to get awareness on environment.
2. To understand the importance of protecting natural resources, ecosystems for future generations and pollution causes due to the day to day activities of human life
3. To save earth from the inventions by the engineers.

SYLLABUS**UNIT I****6 Hours**

Multidisciplinary Nature of Environmental Studies: – Definition, Scope and Importance – Need for Public Awareness.

Natural Resources : Renewable and non-renewable resources – Natural resources and associated problems – Forest resources – Use and over – exploitation, deforestation, case studies – Timber extraction – Mining, dams and other effects on forest and tribal people – Water resources – Use and over utilization of surface and ground water – Floods, drought, conflicts over water, dams – benefits and problems , Energy resources:

Self-Learning Topics: Food resources, World Food Problems.

UNIT II**7 Hours**

Ecosystems: Concept of an ecosystem. – Structure and function of an ecosystem – Producers, consumers and decomposers – Energy flow in the ecosystem – Ecological succession – Food chains, food webs and ecological pyramids – Introduction, types, characteristic features, structure and function of the following ecosystem: a. Forest ecosystem. b. Aquatic ecosystem

Biodiversity and its Conservation : Introduction ,Definition, Values of biodiversity– India as a mega-diversity nation – Hot-spots of biodiversity – Threats to biodiversity, Endangered and endemic species of India – Conservation of biodiversity: In-situ and Ex-situ conservation of biodiversity.

Self-Learning Topics: Desert Ecosystem, Bio-Geographical Classification of India.

UNIT III**6 Hours**

Environmental Pollution: Definition, Cause, effects and control measures of:

- a. Air Pollution.
- b. Water pollution
- c. Soil pollution

Solid Waste Management: Causes, effects and control measures of urban and industrial wastes – Role of an individual in prevention of pollution – Pollution case studies – Disaster management: floods, earthquake, cyclone and landslides.

Self-Learning Topics : Cause, effects and control measures of :Marine Pollution,Thermal Pollution.

UNIT IV**6 Hours**

Social Issues and the Environment: From Unsustainable to Sustainable development Water conservation, rain water harvesting, – Resettlement and rehabilitation of people; its problems and concerns. Case studies – Environmental ethics: Issues and possible solutions – Climate Changes: global warming, acid rain, ozone layer depletion. Environment Protection Act. – Air (Prevention and Control of Pollution) Act. – Wildlife Protection Act – Forest Conservation Act .

Self-Learning Topics: Water Shed Management, Water (Prevention and control of Pollution) Act

UNIT V**7 Hours**

Human Population and the Environment: Population growth, variation among nations. Population explosion – Family Welfare Programmes. – Environment and human health – Human Rights – Value Education – HIV/AIDS – Women and Child Welfare – Role of information

Rights – Value Education – HIV/AIDS – Women and Child Welfare – Role of information Technology in Environment and human health – Case studies.

Field Work: Visit to a local area to document environmental assets River/forest grassland/hill/mountain – Visit to a local polluted site-Urban/Rural/Industrial/Agricultural Study of common plants, insects, and birds – river, hill slopes, etc.

Board of Studies : Basic Science and Humanities Chemistry

Approved in BOS No: 05th, August, 2024

Approved in ACM No: 01

Text Books:

1. Textbook of Environmental Studies for Undergraduate Courses Erach Bharucha for University Grants Commission, Universities Press.
2. Palaniswamy, “Environmental Studies”, Pearson education
3. S.Azeem Unnisa, “Environmental Studies” Academic Publishing Company
4. K.Raghavan Nambiar, “Text book of Environmental Studies for Undergraduate Courses as per UGC model syllabus”, Scitech Publications (India), Pvt. Ltd.

References:

1. Deeksha Dave and E.Sai Baba Reddy, “Textbook of Environmental Science”, Cengage Publications.
2. M.Anji Reddy, “Text book of Environmental Sciences and Technology”, BS Publication.
3. J.P.Sharma, Comprehensive Environmental studies, Laxmi publications.
4. J. Glynn Henry and Gary W. Heinke, “Environmental Sciences and Engineering”, Prentice Hall of India Private limited
5. G.R.Chatwal, “A Text Book of Environmental Studies” Himalaya Publishing House
6. Gilbert M. Masters and Wendell P. Ela, “Introduction to Environmental Engineering and Science, Prentice Hall of India Private limited.

Sample Questions

Unit –I

1. Environmental Science is Multidisciplinary in nature .Justify?
2. Explain the difference between renewable and non-renewable resources.
3. Why forests are considered a critical natural resource?
4. Discuss the positive & negative impacts of Big Dams.
5. Explain the Scope and importance of Environmental Science.
6. Apply the idea of conservation to suggest ways a community can reduce its reliance on fossil fuels.

Unit-II

1. Explain how energy flows through an ecosystem.
2. Describe the role of producers, consumers, and decomposers in an ecosystem
3. Apply the concept of ecological succession to explain how a forest might develop after a wildfire.
4. Explain the Values of Bio-Diversity
5. Evaluate the effectiveness of protected areas in conserving biodiversity.

Unit-III

1. Explain how air pollution can impact human health.
2. Examine how deforestation contributes to both air and water pollution.
3. Explain how agricultural runoff contributes to water pollution.

4. Compare the effects of chemical pollutants versus biological pollutants on freshwater ecosystems.
5. How can a municipality implement a successful recycling program to reduce the amount of waste sent to landfills?
6. What steps should a local government take to enhance resilience against floods in an urban area?

Unit-IV

1. Describe how sustainable development aims to balance economic growth, environmental protection, and social equity.
2. Explain how a rainwater harvesting system works from collection to storage.
3. How can a government design a resettlement program that minimizes disruption to affected communities?
4. What steps can individuals take to lower their personal contribution to global warming?
5. Discuss (i) Air (Prevention and Control of Pollution) Act (ii) Wildlife Protection Act

Unit-V

1. Explain how rapid population growth can impact natural resources.
2. Examine how population growth trends influence energy consumption and
3. Discuss possible solutions for sustainable energy use.
4. Evaluate the effectiveness of population control measures in addressing environmental issues in a specific country or region.
5. How can healthcare providers ensure that people at high risk for HIV are tested and receive appropriate prevention and care services?
6. Explain the concept of gender equality and its importance in women and child welfare.
- 7 Discuss the Role of information Technology in Environment and human health

**Chairperson
Board of Studies (Chemistry)**

R24BS11**PROBABILITY & STATISTICS**
(Common to CSE, CSE (AI&ML), CSE (DS))**3 0 0 3****Course Objectives:**

1. Understand the concepts of descriptive statistics and application of statistical measures.
2. Identify discrete and continuous random variables, apply probability distributions.
3. Understand the concepts of sampling distribution, estimation and construction of confidence interval.
4. Understand how to apply various statistical tests.
5. Understand how to find nature as well as the amount of relationship between the given variables.

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

Course Code	Course Outcomes	Mapping with POs			Dok
		PO1	PO2	PO12	
R24BS11.1	Acquire knowledge in finding the analysis of the data quantitatively or categorically and various statistical elementary tools	2	2	1	L1,L2
R24BS11.2	Develop skills in designing mathematical models involving probability, random variables and the critical thinking in the theory of probability and its applications in real life problems.	2	2	1	L2,L3
R24BS11.3	Apply the theoretical probability distributions like binomial, Poisson, and Normal in the relevant application areas.	2	2	1	L1,L2,L3
R24BS11.4	Analyze to test various hypotheses included in theory and types of errors for large samples.	2	2	1	L4
R24BS11.5	Apply the different testing tools like t-test, F-test, chi-square test to analyze the relevant real-life problems.	2	2	1	L4,L5

SYLLABUS**UNIT I: Descriptive statistics****10 hours**

Statistics Introduction, Population vs Sample, Collection of data, primary and secondary data, Measures of Central tendency, Measures of Variability (spread or variance) Skewness, Kurtosis, correlation, correlation coefficient, rank correlation, regression coefficients, method of least squares, regression lines.

COs-CO1

Self-Learning Topic: Positive correlations are the relationship between the heights and weights of the individual persons.

UNIT II: Probability Concepts**10 hours**

Probability, probability axioms, addition law and multiplicative law of probability, conditional Probability, Baye's theorem, random variables (discrete and continuous), probability density functions, properties, mathematical expectation.

COs-CO2

Self- Learning Topic: Analyzing patterns in health data, calculate the probability of certain health outcomes for individuals.

UNIT III: Probability distributions**10 hours**

hypothesis, alternative hypothesis, the critical and acceptance regions, level of significance, two types of errors and power of the test. Large Sample Tests: Test for single proportion, difference of proportions, test for single mean and difference of means. Confidence interval for parameters in one sample and two sample problems of fit. **COs-CO4**

UNIT V: Small sample tests:

8 hours

Student t-distribution (test for single mean, two means and paired t-test), testing of equality of variances (F- test), chi-square test for goodness of fit, chi-square test for independence of attributes.

COs-CO5

Textbooks

1. Miller and Friends, Probability and Statistics for Engineers, 7/e, Pearson, 2008.
2. S.C. Gupta and V.K. Kapoor, Fundamentals of Mathematical Statistics, 11/e, Sultan Chand & Sons Publications, 2012.

Reference Books

1. B.V. Ramana, Higher Engineering Mathematics, Mc Graw Hill Education. Scientists, 8th Edition, Pearson 2007.
2. S. Ross, A First Course in Probability, Pearson Education India, 2002.
3. W. Feller, An Introduction to Probability Theory and its Applications, 1/e, Wiley, 1968.

Online Learning Resources

1. https://onlinecourses.nptel.ac.in/noc21_ma74/preview
2. https://onlinecourses.nptel.ac.in/noc22_mg31/preview

Board of Studies : Basic Science and Humanities -Mathematics

Approved in BOS No: 2nd August, 2024

Approved in ACM No: 01

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	30	10
L2	30	10
L3	40	30
L4	-	25
L5	-	25
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

UNIT 1

- 1) Describe primary and secondary data with an example each. (L1)
- 2) Distinguish between positive and negative Skewness.. (L4)
- 3) What are the measures of variability and explain them. (L2)
- 4) Fit a second degree polynomial to the following data by the method of least squares. (L3)

X	0	1	2	3	4
Y	1	1.8	1.3	2.5	6.3

UNIT 2

- 1) List the properties of discrete and continuous random variables. (L1)
- 2) State and prove Baye's theorem. (L2)
- 3) A sample of 3 items is selected at random from a box containing 10 items of which 4 are defective. Find the expected number of defective items. (L4)

UNIT 3

- 1) Define: (i) Binomial (ii) Poisson (iii) Uniform and (iv) Normal distributions. (L1)
- 2) State and prove Chebyshev's Inequality. (L2)
- 3) Fit the Poisson distribution for the following data (L3)

X	0	1	2	3	4	5
Y	147	147	74	25	6	1

UNIT 4

- 1) Explain point and interval estimations. (L2)
- 2) Explain the types of errors in sampling. (L2)
- 3) A random sample of 125 teachers in a large metropolitan area revealed a mean weekly salary of Rs. 527 with a standard deviation Rs. 45. With what degree of confidence can we assert that the average weekly salary of all teachers in the metropolitan area is between 495 to 532? (L3)

UNIT -5

1. Define: t, F and chi square test. (L1)
2. Write the test statistic for
 - (i) The test of significance for single mean for the test of significance for single proportion and
 - (ii) difference of means (L3)
- 4) Find the maximum difference that we can expect with probability 0.95 between the means of sample sizes 10 and 12 from normal population if their standard deviations are found to be 2 and 3 respectively. (L 5)

**Chairperson
Board of Studies (Mathematics)**

R24CSPC08**Formal Languages and Automata Theory****3 0 0 3**

(Common to CSE, CSE (DS), CSE (AI&ML))

Course Objectives:

The course is designed with the objective to:

- Introduce languages, grammars, and computational models
- Understand the foundational principles of Automata Theory & Analyze and design Finite Automata (FA)
- Formulate Regular Expressions and Grammars and demonstrate their equivalence to Finite Automata
- Discussing regular expressions and regular languages
- Construct Context Free Grammars (CFG) to describe context-free languages, and simplify and normalize these grammars into CNF and GNF.
- Design and analyze Pushdown Automata (PDA) to recognize CFL'S and understand their relationship with CFG
- Explain the capabilities and limitations of Turing Machines & Demonstrate decidability and un decidability for NP Hard problems

Course out Comes

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

Course Code	Course Outcomes	Mapping with POs and PSOs						Dok
		PO 1	PO 2	PO 3	PO 4	PS0 1	PS0 2	
R24CSPC08.1	Comprehensible to understand the concept of abstract machine and their ability to recognize languages based on their computational power and the types of languages they can process	3	3	2	--	3	2	L1, L2
R24CSPC08.2	Understand regular expressions and finite automata	3	3	3	--	3	2	L1, L2
R24CSPC08.3	Develop context free grammars for formal languages.	3	3	3	--	2	3	L1, L3
R24CSPC08.4	Design pushdown automata for context free grammars	3	3	3	--	3	2	L4,L5
R24CSPC08.5	Design Turing machine and Formulate decidability and un-decidability problems and P&NP problems	3	3	3	2	3	3	L5,L6

SYLLABUS**UNIT-I: Finite Automata:****14 hours**

Need of Automata theory, Central Concepts of Automata Theory, Automation, Finite Automata, Transition Systems, Acceptance of a String, DFA, Design of DFAs, NFA, Design of NFA, Equivalence of DFA and NFA, Conversion of NFA into DFA, Finite Automata with ϵ -Transitions, Minimization of Finite Automata, Finite Automata with output-Mealy and Moore Machines, Applications and Limitation of Finite Automata.

Self Learning Concepts: Research how DFAs/NFAs are used in lexical analyzer, Network protocol Design & Control system.

CO's–CO1**UNIT-II: Regular Expressions****12 hours**

Regular Expressions, Regular Sets, Identity Rules, Equivalence of two RE, Inter Conversion, Equivalence between FA and RE, Pumping Lemma of Regular Sets, Grammars, Classification of Grammars, Chomsky Hierarchy Theorem, Right and Left Linear Regular Grammars, Equivalence between RG and FA, Inter Conversion.

CO's–CO2

Self Learning Concepts: Closure properties of Regular set

UNIT-III: Context Free Grammar**10 hours**

Formal Languages, Context Free Grammar, Leftmost and Rightmost Derivations, Parse Trees, Ambiguous Grammars, Simplification of Context Free Grammars, Normal Forms-CNF and GNF, Pumping Lemma CFG, Applications of Context Free Grammars.

COs–CO3

Self Learning Concepts: Elimination of left recursion & left factoring

UNIT-IV: Pushdown Automata**12 hours**

Pushdown Automata, Definition, Model, Graphical Notation, Instantaneous Description, Language Acceptance of Pushdown Automata, Design of Pushdown Automata, Deterministic and Non – Deterministic Pushdown Automata, Equivalence of Pushdown Automata and Context Free Grammars, Conversion, Two Stack Pushdown Automata, Application of Pushdown Automata.

CO's–CO4

Self Learning Concepts: Investigate practical applications of PDAs in parsing such as in compiler & syntax analysis.

UNIT-V: Turning Machine**14 hours**

Turning Machine: Definition, Model, Representation of TMs-Instantaneous Descriptions, Transition Tables and Transition Diagrams, Language Acceptance of a TM, Design of TMs, Types of TMs, Universal TM, Decidable and Un-decidable Problems, Halting Problem of TMs, Post's Correspondence Problem, Modified PCP, Classes of P and NP, NP-Hard and NP-Complete Problems.

CO's–CO5

Self Learning Concepts: Church's Thesis for turning machine.

Board of Studies: Computer Science and Engineering

Approved in BoS No: 02, 9th May, 2025

Approved in ACM No: 02

Text Books:

- 1) Introduction to Automata Theory, Languages and Computation, J. E. Hopcroft, R. Motwani and J. D. Ullman, 3rd Edition, Pearson, 2008.
- 2) Theory of Computer Science-Automata, Languages and Computation, K. L. P. Mishra and N.Chandrasekharan, 3rd Edition, PHI, 2007.

Reference Books:

- 1) Elements of Theory of Computation, Lewis H.P. & Papadimitriou C.H., Pearson /PHI
- 2) Theory of Computation, V. Kulkarni, Oxford University Press, 2013.
- 3) Theory of Automata, Languages and Computation, Rajendra Kumar, McGraw Hill, 2014

Web References:

- 1) <https://nptel.ac.in/courses/106103070>
- 2) <https://nptel.ac.in/courses/106106049>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35%	--
L2	40%	--
L3	25%	15%
L4	--	35%
L5	--	35%
L6	--	15%
Total (%)	100%	100%

Sample Short and Long Answers questions of Various Cognitive Levels**L1: Remember**

- 1) What is Automata Theory and why is it needed?
- 2) What are Mealy and Moore Machines?
- 3) Define Regular Expressions and Regular Sets.
- 4) List the Identity Rules for Regular Expressions.
- 5) Define Chomsky Normal Form (CNF) and Greibach Normal Form (GNF).
- 6) State the Pumping Lemma for CFG.
- 7) What is an Instantaneous Description of a PDA?
- 8) Differentiate between Deterministic and Non-deterministic PDA.
- 9) Define a Turing Machine..

10) Define Decidable and Undecidable Problems

L2: Understand

- 1) Describe the differences between a DFA and an NFA.
- 2) Explain the significance of minimizing Finite Automata.
- 3) Explain the equivalence of two Regular Expressions.
- 4) Describe the inter-conversion between Finite Automata and Regular Expressions.
- 5) Explain how Parse Trees represent the derivation of a string.
- 6) Describe the steps involved in simplifying Context Free Grammars.
- 7) Describe the relationship between PDA and Context Free Grammars.
- 8) Explain the process of converting a CFG to a PDA and vice versa.
- 9) Describe the different types of Turing Machines.
- 10) Describe the classes of P and NP.

L3: Apply

- 1) Design a DFA to accept strings with a specific pattern (e.g., strings containing "ab").
- 2) Construct a Mealy Machine to perform a specific sequence transformation.
- 3) Convert a given Finite Automaton to a Regular Expression.
- 4) Use the Pumping Lemma to prove whether a given set is regular or not.
- 5) Convert a given CFG to CNF. And Convert a given CFG to GNF.
- 6) Use the Pumping Lemma to prove whether a language is Context Free or no
- 7) Design a PDA to accept a given Context Free Language.
- 8) Convert a given CFG to an equivalent PDA.
- 9) Design a Turing Machine to perform a specific computation (e.g., addition, subtraction).
- 10) Represent a Turing Machine using Instantaneous Descriptions.

L4: Analysing

- 1) Analyze the time and space complexity of DFA vs. NFA.
- 2) Differentiate between the capabilities and limitations of Finite Automata
- 3) Analyze the equivalence of different Regular Expressions.
- 4) Analyze the given grammar and classify it according to Chomsky Hierarchy.
- 5) Analyze a given grammar and determine if it is ambiguous.
- 6) Compare the advantages and disadvantages of CNF and GNF
- 7) Analyze the limitations of Pushdown Automata.
- 8) Compare the complexity of Deterministic and Non-deterministic PDA.
- 9) Design a Turing Machine to perform a specific computation (e.g., addition, subtraction).
- 10) Represent a Turing Machine using Instantaneous Descriptions.
- 11) Show that a given problem is decidable or undecidable.

L5: Evaluating

- 1) Evaluate whether a given language can be recognized by a Finite Automaton.
- 2) Compare different designs of Finite Automata for the same language in terms of efficiency.
- 3) Justify the need for NFA when DFA can recognize the same language.
- 4) Evaluate the limitations of Regular Expressions in expressing certain languages.
- 5) Determine whether a given language can be described by a Regular Expression.
- 6) Evaluate whether a given language can be generated by a Context Free Grammar.
- 7) Assess the impact of ambiguity in a grammar.
- 8) Evaluate whether a given language can be generated by a Context Free Grammar.
- 9) Assess the impact of ambiguity in a grammar.
- 10) Evaluate the impact of the Halting Problem on computation.
- 11) Assess the theoretical limits of computation.
- 12) Justify the importance of P vs. NP problem.

L6: Creating

- 1) Design a Context Free Grammar for a given language.
- 2) Develop a parser for a simple Context Free Grammar.
- 3) Design a Pushdown Automaton with additional features.
- 4) Develop an application using PDA for tasks like expression evaluation or syntax checking

**Chairperson
Board of Studies (CSE)**

R24CSPC09**DATABASE MANAGEMENT SYSTEMS****3 0 0 3**

(Common to CSE, CSE (DS), CSE (AI&ML))

Course Objectives:

- Train in the fundamental concepts of database management systems, database modelling and design, SQL, PL/SQL, and System implementation techniques.
- Enable students to model ER diagram for any customized applications.
- Provide knowledge on concurrency techniques.
- Understand normalization theory and apply such knowledge to the normalization of a database.
- To learn the principles of systematically designing and using large scale Database Management Systems for various applications.

Course Outcomes

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

Course Code	Course Outcomes (COs)	Mapping with POs and PSOs								DoK
		PO1	PO2	PO3	PO4	PO5	PO11	PSO1	PSO2	
R24CSPC09.1	Understand basic concepts of database systems and data models.	3	2	-	1	2	2	3	-	L 1, L2
R24CSPC09.2	Design ER diagrams and apply conceptual modeling for real-world problems like Railway & Hospital systems.	3	3	3	1	2	2	3	1	L1, L2, L3
R24CSPC09.3	Apply relational model concepts and SQL operations to implement simple databases.	3	2	2	1	3	2	3	1	L 2, L3
R24CSPC09.4	Perform schema refinement using normalization techniques.	3	2	2	2	3	2	3	-	L3, L4
R24CSPC09.5	Explain transaction management concepts and implement concurrency control and recovery methods. Understand indexing techniques like B+ Trees and Hash-based indexing for efficient data retrieval.	3	3	2	3	3	2	3	-	L4, L5, L6

SYLLABUS**UNIT I:****15 Hours**

Introduction: Database system, Characteristics (Database Vs File System), Database Users, Advantages of Database systems, Database applications. Brief introduction of different Data Models; Concepts of Schema, Instance and data independence; three tier schema architecture for

data independence; Database system structure, environment, Centralized and Client Server architecture for the database.

Entity Relationship Model: Introduction, Representation of entities, attributes, entity set, relationship, relationship set, constraints, sub classes, super class, inheritance, specialization, generalization using ER Diagrams **CO's–CO1**

Self Learning Topics: Database Applications (Real-world applications: Banking, E-commerce, Healthcare, Education, and Social Media)

UNIT II: 15 Hours

Relational Model: Introduction to relational model, concepts of domain, attribute, tuple, relation, importance of null values, constraints (Domain, Key constraints, integrity constraints) and their importance, Relational Algebra, Relational Calculus.

BASIC SQL: Simple Database schema, data types, table definitions (create, alter), different DML operations (insert, delete, update). **CO's–CO2**

Self Learning Topics: Structure of a Relational Database

UNIT III: 15 Hours

SQL: Basic SQL querying (select and project) using where clause, arithmetic & logical operations, SQL functions (Date and Time, Numeric, String conversion). Creating tables with relationship, implementation of key and integrity constraints, nested queries, sub queries, grouping, aggregation, ordering, implementation of different types of joins, view (updatable and non-updatable), relational set operations. **CO's–CO3**

Self-Learning Topics: Advanced Grouping and Aggregation (GROUP BY with ROLLUP, CUBE, and GROUPING SETS, Filtering groups using HAVING)

UNIT IV: 10 Hours

Schema Refinement (Normalization): Purpose of Normalization or schema refinement, concept of functional dependency, normal forms based on functional dependency Lossless join and dependency preserving decomposition, (1NF, 2NF and 3 NF), concept of surrogate key, Boyce Codd-normal form (BCNF), MVD, Fourth normal form(4NF), Fifth Normal Form (5NF). **CO's–CO4**

Self-Learning Topics: Decomposition Deep Dive (Comparing Lossless vs. Lossy Decomposition, Dependency-Preserving Decomposition: When to compromise)

UNIT V: 10 Hours

Transaction Concept: Transaction State, ACID properties, Concurrent Executions, Serializability, Recoverability, Implementation of Isolation, Testing for Serializability, lock based, time stamp based, optimistic, concurrency protocols, Deadlocks, Failure Classification, Storage, Recovery and Atomicity, Recovery algorithm.

Introduction to Indexing Techniques: B+ Trees, operations on B+ Trees, Hash Based Indexing **CO's–CO5**

Self-Learning Topics: Locking & Deadlock Handling – Strategies for handling concurrent transactions.

Board of Studies: Computer Science and Engineering

Approved in BOS No: 02, 9th May, 2025

Approved in ACM No: 02

Text Books:

1. Introduction to Database Systems, CJ Date, Pearson.
2. Data base Management Systems, Raghurama Krishnan, Johannes Gehrke, TATA Mc Graw Hill 3rd Edition.

Reference Books:

1. Data base Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, Elmasri Navrate Pearson Education.
3. Database Systems - The Complete Book, H G Molina, J D Ullman, J Widom Pearson.
4. Data base System Concepts, 5/e, Silberschatz, Korth, TMH.

Web References:

1. <http://www.c4learn.com/>
2. <http://www.geeksforgeeks.org/c/>
3. <http://nptel.ac.in/courses/122104019/>
4. <http://www.learn-c.org/>
5. <https://www.tutorialspoint.com/c-programming/>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35%	--
L2	40%	--
L3	25%	25%
L4	--	35%
L5	--	25%
L6		15%
Total (%)	100%	100%

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. What is a database system?
2. Define a schema and an instance.
3. List the types of database users.
4. What are the differences between a file system and a database system?

5. State the purpose of data independence.
6. Name any three data models used in databases.
7. What is an entity set in an ER model?
8. Explain how a database system differs from a file-based system in terms of data handling.
9. Describe the role of a Database Administrator (DBA).
10. Illustrate the components of the three-tier schema architecture with an example.
11. Explain the importance of constraints in ER modeling.
12. Describe the relationship between entities and attributes in an ER diagram.

L2: Understand

1. What is a relation in the relational model?
2. Define the terms: domain, tuple, and attribute.
3. List different types of integrity constraints in relational databases.
4. What are the basic DML operations in SQL?
5. Name any four data types used in SQL.
6. Explain the importance of null values in relational databases.
7. Describe how key constraints help maintain data integrity.
8. Interpret a relational algebra expression involving selection and projection.
9. Differentiate between `CREATE` and `ALTER` table commands in SQL.
10. Explain the role of relational calculus in database querying.
11. Write an SQL query to insert data into a student table.
12. Create a table named `EMPLOYEE` with appropriate data types and constraints.
13. Apply the concept of domain constraints by defining a column with restricted values.
14. Use relational algebra to express: "Find names of all employees in department 10."
15. Use `DELETE` and `UPDATE` commands to modify data in a sample table.

L3: Apply

1. Explain the importance of null values in relational databases.
2. Describe how key constraints help maintain data integrity.
3. Interpret a relational algebra expression involving selection and projection.
4. Differentiate between `CREATE` and `ALTER` table commands in SQL.
5. Explain the role of relational calculus in database querying.
6. Write an SQL query to insert data into a student table.
7. Create a table named `EMPLOYEE` with appropriate data types and constraints.
8. Apply the concept of domain constraints by defining a column with restricted values.
9. Use relational algebra to express: "Find names of all employees in department 10."
10. Use `DELETE` and `UPDATE` commands to modify data in a sample table.

L4: Analysing

1. Convert the following unnormalized table to 1NF, 2NF, and 3NF step-by-step.
2. Apply the concept of functional dependency to identify keys in a relational schema.
3. Use a sample relation to demonstrate the process of decomposing a schema into BCNF.
4. Apply 4NF on a table containing multivalued dependencies.
5. Normalize a relation to 5NF using a suitable example
6. Compare and contrast BCNF and 3NF. In what scenario is BCNF preferred?
7. Analyze the importance of dependency preservation in database normalization.
8. Examine a given schema and identify all possible functional and multivalued dependencies.
9. Identify and explain the anomalies that can occur in an unnormalized database.
10. Distinguish between lossless decomposition and lossy decomposition with illustrations.

L5: Evaluating

1. Analyze a transaction schedule and determine if it is conflict-serializable
2. Identify and resolve a deadlock using wait-die or wound-wait protocol.
3. Compare lock-based and timestamp-based concurrency control protocols.
4. Examine failure classification and explain how each type affects recovery.
5. Analyze how optimistic concurrency control differs from pessimistic control.
6. Evaluate the trade-offs between optimistic and pessimistic concurrency control.
7. Justify the need for serializability in concurrent transaction execution.
8. Assess the efficiency of B+ Trees vs. Hash indexing for read-heavy applications.
9. Evaluate a given schedule for recoverability and cascading rollbacks.
10. Critique the limitations of timestamp ordering in high-concurrency environments.

L6: Creating

1. Design a concurrency control strategy for a banking application handling concurrent withdrawals.
2. Develop a custom recovery algorithm for a distributed database system.
3. Construct a complete B+ Tree from a given dataset and demonstrate insert/delete operations.
4. Design a system using optimistic concurrency control and simulate conflict resolution.
5. Create a detailed fault-recovery workflow that handles both system crash and transaction failure using logs and checkpoints.

**Chairperson
Board of Studies (CSE)**

R24CSPC10**Digital Logic & Computer Organization**
(Computer Science and Engineering)**3 0 0 3****Course Objectives:**

The main objectives of the course is to

- Provide students with a comprehensive understanding of digital logic design principles and computer organization fundamentals
- Apply Boolean algebras and simplify logic expressions and can design sequential circuits.
- Design combinational circuits and perform arithmetic operations using binary address, substructures, multiplies and code converter.
- Explain input/output (I/O) systems and their interaction with the CPU, memory, and peripheral devices
- Explore secondary storage mechanism and their role in memory hierarchy.

Course Outcomes

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

Course Code	Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	Dok
R24CSPC 10.01	Understand Significance of Number Systems, Conversions, and Binary Codes.	3	2	2	1	2	-	-	1	1	1	-	3	2	L2, L3
R24CSPC 10.02	Apply Different Simplification Methods For Minimizing Boolean Functions, The Operation Of Flip-Flops, Registers, And Counters	3	3	2	2	2	-	-	1	1	1	-	3	2	L1, L4
R24CSPC 10.03	Illustrate Knowledge On Design Of Various Combinational Circuits.	3	3	3	2	2	-	-	1	1	2	1	3	2	L1, L3
R24CSPC 10.04	Understand the Basic Structure of Computer and Memory Organization.	3	3	2	2	3	1	1	1	2	2	2	2	1	L2, L4
R24CSPC 10.05	Discuss The Basic Structure And Organization Of Computers.	3	3	2	2	3	1	2	1	1	2	2	2	1	L5, L6

SYLLABUS**UNIT I:****15 Hours**

Number Systems and Codes: Decimal, Binary, Octal, and Hexa-decimal number systems and their conversions, ASCII code, Excess-3 code, Gray code, Complement representation of negative numbers: Signed Magnitude, One's complement method, Two's complement method, Binary

Arithmetic.

Digital Logic Circuits-I: Basic Logic Functions, Logic gates, universal logic gates, Minimization of Logic expressions. K-Map Simplification, Combinational Circuits, Decoders, Multiplexers

Self Learning Concepts:

CO's-CO1

Basic Gates: AND, OR, NOT, XOR Learn 2, 3, 4-variable K-Maps, Multiplexers (MUX): Select one input among many

UNIT II:

15 Hours

Boolean algebra: Boolean operations, Boolean functions, algebraic manipulations, min-terms and max-terms, sum-of products and product-of-sum representations, two-input logic gates, NAND/NOR implementations, Minimization of Boolean functions using Karnaugh map, don't-care conditions, prime implicants, Tabular Method

CO's-CO2

Digital Logic Circuits-II: Sequential Circuits, Flip-Flops, Binary counters, Registers, Shift, Registers, Ripple counters

Self Learning Concepts: Shift Registers: Serial-In Serial-Out (SISO), Serial-In Parallel-Out (SIPO), etc

UNIT III:

15 Hours

Combinational Logic Design: Analysis of combinational circuits, Design Procedure – Binary Adder, Subtractor, BCD Adder, multiplier, comparator, decoders, encoders, multiplexers, demultiplexers, Code Converters

Computer Arithmetic : Addition and Subtraction of Signed Numbers, Design of Fast Adders, Multiplication of Positive Numbers, Signed-operand Multiplication, Fast Multiplication, Integer Division, Floating-Point Numbers and Operations

CO's-CO3

Self Learning Concepts: Use Half Adder (1-bit) and Full Adder (with carry), simulate arithmetic circuits and MUX/DEMUX.

UNIT IV:

15 Hours

Basic Structure of Computers: Computer Types, Functional units, Basic operational concepts, Bus structures, Software, Performance, multiprocessors and multi computers, Computer Generations, Von- Neumann Architecture.

Computer Organization

Organization and Architecture, Structure and Function, Computer Components, Bus Interconnection, Processor Organization, Register Organization. Instruction codes, Computer instructions, Memory reference instructions, Instruction Cycle. Stack organization, instruction formats, addressing modes, program control.

CO's-CO4

Self Learning Concepts: Memorize types of registers: Program Counter (PC), Instruction Register (IR), Accumulator, MAR, MDR

UNIT V:

10 Hours

The Memory Organization: Basic Concepts, Semiconductor RAM Memories, Read-Only Memories, Speed, Size and Cost, Cache Memories, Performance Considerations, Virtual Memories, Memory Management Requirements, Secondary Storage.

CO'S-CO5

Self Learning Concepts: Learn the difference between volatile (e.g., RAM) **and** non-volatile (e.g., ROM, SSD) memory,

Board of Studies : Computer Science and Engineering

Approved in BOS No: 02, 9th May, 2025

Approved in ACM No: 02

Text Books:

1. Computer Organization, Carl Hamacher, Zvonko Vranesic, Safwat Zaky, 6th edition, McGraw Hill
2. M. Moris Mano, Computer Systems Architecture, 3rd Edition, Pearson Education, 2007.
3. William Stallings, Computer Organization and Architecture: Designing for Performance, 10th Edition, Pearson Education, 2016

Reference Books:

1. Computer Systems Architecture, M. Moris Mano, 3rd Edition, Pearson
2. Computer Organization and Design, David A. Paterson, John L. Hennessy, Elsevier
3. Fundamentals of Logic Design, Roth, 5th Edition, Thomson M. Moris Mano, Computer Systems Architecture, 3rd Edition, Pearson Education, 2007.
4. John P. Hayes, Computer Architecture and Organization, 3rd Edition, WCB/McGraw-Hill.
5. M. Morris Mano and Michael D. Ciletti, Digital Design, 4th Edition, Pearson Education, 2013.
6. Vincent P. Heuring and Harry F. Jordan, Computer System Design and Architecture, 2nd Edition, Pearson Education, 2004.
7. Carl Hamacher, Computer Organization and Embedded Systems, 6th Edition, McGraw Hill Higher Education, 2002.

Online Learning Resources:

1. <https://archive.nptel.ac.in/courses/106/105/106105163/>
2. https://onlinecourses.nptel.ac.in/noc21_ee39/preview

Web References:

1. <http://www.c4learn.com/>
2. <http://www.geeksforgeeks.org/c/>
3. <http://nptel.ac.in/courses/106105163/>
4. <http://www.learn-c.org/>
5. <https://www.tutorialspoint.com/c-programming/>

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	20
L4	--	40
L5	--	30
L6	--	10
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. Define binary, octal, and hexadecimal number systems.
2. What is ASCII code?
3. List the basic logic gates and their symbols.
4. Define min-term and max-term.
5. What is a Karnaugh map?
6. Name different types of flip-flops.
7. Define BCD adder and comparator.
8. What is a multiplexer?
9. List types of code converters
10. Define Von Neumann Architecture.
11. What are the main types of computer memory?
12. List the generations of computers.
13. Define addressing modes.
14. What is a stack organization?
15. List the components of a processor

L2: Understand

1. Convert the decimal number 156 to binary, octal, and hexadecimal.
2. Explain the difference between signed magnitude and two's complement representation.
3. Draw truth tables for NAND and NOR gates.
4. Explain the concept of "don't care" conditions in K-map.
5. Describe the difference between sequential and combinational circuits.
6. Explain how a D flip-flop works with a logic diagram
7. Explain the design steps of a full adder.
8. Describe floating-point number representation.
9. Describe the role of demultiplexer in digital systems.

10. Explain the basic operational concepts of a computer.
11. Describe cache memory and its advantages.
12. Describe bus structures with examples

L3: Apply

1. Perform binary subtraction using one's complement: $(10101)_2 - (01010)_2$
2. Implement the function $F = AB + A'C$ using basic gates.
3. Simplify the Boolean function $F = AB + A'B$ using a K-map.
4. Implement $F = A'B + AB$ using NAND gates only.
5. Construct a 3-variable K-map and simplify $F(A, B, C) = \Sigma(0,2,3,5)$.
6. Design a 4-bit ripple counter
7. Analyze the difference between ripple and carry-look ahead adders.
8. Compare binary and BCD arithmetic operations.
9. Analyze the circuit of a 4-bit binary multiplier.
10. Compare multiprocessors and multicomputer.
11. Analyze the role of virtual memory in system performance.
12. Examine the trade-off between memory speed and cost
13. Apply addressing modes to access memory operands.
14. Implement a simple stack operation using instructions.
15. Demonstrate the working of a basic instruction decoder

L4: Analysing

1. Analyze the operation of a 4:1 multiplexer and write its truth table.
2. Compare Excess-3 and Gray code with examples.
3. Design a 2-to-4 decoder using logic gates.
4. Analyze the behaviour of JK flip-flop in toggle mode.
5. Compare different methods of Boolean function minimization.
6. Analyze the working of a 4-bit shift register
7. Analyze the difference between ripple and carry-look ahead adders.
8. Compare binary and BCD arithmetic operations.
9. Analyze the circuit of a 4-bit binary multiplier.
10. Compare multiprocessors and multicomputer.
11. Analyze the role of virtual memory in system performance.
12. Examine the trade-off between memory speed and cost
13. Analyze how register organization affects instruction execution speed.
14. Compare different addressing modes with examples.
15. Break down the execution flow of memory-reference instructions.

L5: Evaluating

1. Justify the use of universal gates in digital circuit design.
2. Evaluate different binary codes for their suitability in digital communication
3. Evaluate the efficiency of tabular method vs. K-map method.
4. Assess which flip-flop is best suited for counter design and why
5. Evaluate the performance of fast adders vs. normal adders in ALUs.
6. Judge the benefits of signed vs. unsigned number representation in hardware
7. Evaluate the Von Neumann architecture in the context of modern processors.
8. Assess how cache size affects overall system performance.
9. Evaluate the efficiency of stack-based organization vs. register-based architecture.
10. Justify the need for multiple instruction formats in modern processors

L6: Create and Develop

1. Create a comparative analysis report between SRAM, DRAM, and ROM technologies focusing on their suitability in smart phones, gaming consoles, and embedded systems.
2. Propose an optimized cache memory design for a real-time system. Include cache mapping techniques (direct, associative, or set-associative) and evaluate their trade-offs.
3. Evaluate the impact of virtual memory on system performance for a multitasking operating system. Suggest improvements or alternatives to reduce page faults.
4. Develop a theoretical memory access pattern and analyze how cache performance (hit/miss rates) would vary under different cache replacement policies (FIFO, LRU, LFU).
5. Compare and contrast static and dynamic RAM in terms of cost, speed, power consumption, and application use cases. Recommend one for a wearable health monitoring device and justify your choice.
6. Critically evaluate the role of memory management techniques (paging, segmentation, and virtual memory) in modern cloud computing environments. How would you improve efficiency?

**Chairperson
Board of Studies (CSE)**

R24CSPC11**OPERATING SYSTEMS****3 0 0 3**

(Computer Science and Engineering)

Course Objectives:

1. To understand operating system overview, functions and services.
2. To analyze process concept, inter process communication and synchronization.
3. Illustrate different conditions for deadlock and their possible solutions.
4. To understand secondary storage management.
5. To understand protection and security.

Course Outcomes

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

Mapping with POs and PSOs															Dok
Course Code	Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	
R24CSPC 11.1	Describe operating system overview, functions, and services	3	-	-	-	2	-	-	-	-	-	2	3	2	L1, L2
R24CSPC 11.2	Analyze process concept, inter-process communication, and synchronization	3	3	-	2	3	-	-	2	2	-	2	3	2	L2, L3
R24CSPC 11.3	Analyze process synchronization mechanisms and dead lock handling techniques.	3	3	-	2	3	-	-	-	-	-	2	3	3	L3, L4
R24CSPC 11.4	Describe methods for mapping secondary storage management.	3	-	-	2	3	-	-	-	-	-	2	3	2	L4, L5
R24CSPC 11.5	Evaluate protection mechanism and security	3	2	3	2	3	2	3	-	-	2	3	-	-	L5, L6

SYLLABUS**UNIT I:****15 Hours**

Operating System Overview: Introduction, Operating system functions, Operating systems operations, Computing environments, Free and Open-Source.

System Structures: Operating System Services, system calls, Types of System Calls, system programs, Operating system structure, Operating system Design and Implementation, Building and Booting an Operating System, Operating system debugging.

CO's-CO1

Self-Learning Topics: Evaluation of operating systems.

UNIT II:**15 Hours**

Processes: Process Concepts, Process states, Process scheduling, Operations on processes, Inter-process communication,

Threads and Concurrency: Multithreading models, Threads, Thread libraries, threading issues.

CPU Scheduling: Basic concepts, Scheduling-Criteria, CPU Scheduling Algorithms, Algorithms Evaluation, Thread Scheduling, Multiple processor scheduling. **CO's–CO2**

Self-Learning Topics: Multicore and Multithreading.

UNIT III:

15 Hours

Process Synchronization: Critical-Section Problem, Peterson's Solution, Synchronization Hardware, Mutex Locks, Semaphores, Monitors, Classic Problems of Synchronization, Synchronization examples

Deadlock: System Model, Deadlock Characterization, Deadlock Prevention, Detection and Avoidance, Recovery from Deadlock. **CO's–CO3**

Self-Learning Topics: Mutual Exclusion: Hardware Support

UNIT IV:

15 Hours

Memory-Management Strategies: Introduction, Contiguous Memory Allocation, Paging, Page-Table Structure, Swapping, Segmentation.

Virtual Memory Management: Virtual Memory, Demand Paging, Copy-on-write, Page-Replacement Algorithms, Frames Allocation, Thrashing.

File System Interface: File Concepts, Access Methods, Directory Structure, File system Implementation: File-system structure, File-system Operations, Directory implementation, File System Mounting, File Sharing and Protection, Allocation Methods, Free-Space Management.

COs–CO4

Self-Learning Topics: Memory Partitioning and Hardware Control Structures.

UNIT V:

10 Hours

Mass Storage: Overview Mass-Storage Structure, Disk Structure, Disk Attachment, Disk Scheduling, RAID Structure

Protection: Goals of Protection, Principles of Protection, Domain of Protection Access Matrix, Implementation of Access Matrix.

Security: Security Problem, Program Threats, System and Network Threats

CO's–CO5

Self-Learning Topics: Disk Cache and I/O Buffering

Board of Studies : Computer Science and Engineering

Approved in BoS No: 02, 9th May, 2025

Approved in ACM No: 02

Expert Talk (To be Delivered by SMEs from Industries)

CO's/ PO's / PSO's

1. Operating Systems in Cloud Computing (Virtualization Techniques in Industry Cloud Platforms (AWS, Azure, GCP), Containerization (Docker, Kubernetes) vs Virtual Machines, Operating System Abstractions in Cloud Architecture) CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2
2. Real-Time Operating Systems (RTOS) in Industry Applications (RTOS in Automotive Systems (ADAS, AUTOSAR), RTOS for Industrial Automation and Robotics vs General Purpose OS in Embedded Systems) CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2

3. Operating Systems in Cybersecurity (OS Hardening Techniques in Enterprises, Secure Boot and Trusted Execution Environments, Role of OS in Endpoint Security Solutions) CO1–CO5, PO1, PO2, PO3, PO11, PSO1, PSO2

Text Books:

1. Operating System Concepts, Abraham Silberchatz, Peter B. Galvin, Greg Gagne, 10th Edition, John Wiley
2. Operating Systems- A Concept based Approach, D. M. Dhamdhere, 2nd Edition, Tata McGraw-Hill
3. Operating System Concepts, Abraham Silberchatz, Peter B. Galvin, Greg Gagne, 7th Edition, John Wiley

Reference Books:

1. Operating Systems-Internal and Design Principles, Stallings, 5th Edition, Pearson Education/PHI, 2005
2. Operating Systems -Internals and Design Principles, Stallings W, 9th edition, Pearson, 2018
3. UNIX: Concepts and Applications, Sumit Abha Das, 4th Edition, Tata McGraw-Hill
4. Operating System: A Design-Oriented Approach, Charles Patrick Crowley

Web References:

1. <https://archive.nptel.ac.in/courses/106/102/106102132/>
2. https://edurev.in/courses/23057_Operating-System
3. https://www.tutorialspoint.com/operating_system

Internal Assessment Pattern

Cognitive Level	Internal Assessment #1(%)	Internal Assessment #2(%)
L1	35	--
L2	40	--
L3	25	25
L4	--	35
L5	--	25
L6	--	15
Total (%)	100	100

Sample Short and Long Answers questions of Various Cognitive Levels

L1: Remember

1. What is an operating system (OS)?
2. What is meant by an operating system's "kernel"?
3. Define "System Programs" in the context of operating systems.
4. What are system calls in an operating system?
5. Define multithreading.
6. What are the basic criteria for CPU scheduling?
7. What is the critical-section problem in process synchronization?
8. What are mutex locks, and how do they function in process synchronization?
9. Define deadlock in a system.
10. What is a page table, and what role does it play in memory management?
11. What is segmentation in memory management?
12. What are the key components of a mass-storage structure?
13. Define the disk structure and its main components.

L2: Understand

1. Explain the role of the operating system in managing hardware resources.
2. Compare the difference between free and open-source operating systems and proprietary operating systems.
3. Describe the process of booting an operating system from start to finish.
4. Explain the different states that a process can be in and how processes transition between these states.
5. How does inter-process communication (IPC) help processes communicate and synchronize?
6. Explain the difference between a process and a thread.
7. Describe the benefits and challenges of multithreading in modern operating systems.
8. Describe how Peterson's solution works to ensure mutual exclusion.
9. How does deadlock occur in a system? Explain the deadlock characterization in terms of the four necessary conditions.
10. Describe the structure of a page table and its role in translating virtual addresses to physical addresses.
11. Explain how the file system structure is organized to store data efficiently.
12. Describe how the directory structure in a file system helps in managing files.
13. Describe how disk scheduling algorithms manage read/write operations and improve efficiency.

L3: Apply

1. In what way does an operating system structure affect its performance?
2. Demonstrate how system services manage the hardware components of a computer system.
3. How can an operating system be debugged to ensure it works correctly?
4. Given a scenario with multiple processes, how would you apply a CPU scheduling algorithm (e.g., FCFS, SJF) to allocate CPU time?
5. In a system with multiple threads, demonstrate how semaphores can be used to solve the Producer-Consumer problem.
6. Given a scenario of resource allocation, explain how you would detect deadlock using a resource allocation graph.
7. Given a scenario with file system fragmentation, how would you apply free-space management techniques to handle the issue?
8. Apply demand paging in a system with limited physical memory. How would you ensure that the page-replacement algorithm minimizes page faults?
9. Given a system with limited physical memory, demonstrate how paging can be used to manage the memory.
10. Apply a disk scheduling algorithm (e.g., FCFS, SSTF) to a series of disk I/O requests and explain the resulting order of operations.
11. Given a security policy, how would you implement the access matrix model to control access to resources?
12. Given a security threat (e.g., buffer overflow), apply appropriate measures to secure the system against this type of attack.

L4: Analysing

1. Analyze how the structure of an operating system (monolithic vs. microkernel) impacts system performance and security.
2. Analyze how computing environments (such as cloud-based and embedded systems) influence the design and functionality of an operating system.
3. Examine how the operating system handles process synchronization using inter-process communication (IPC) mechanisms like message passing and shared memory.
4. Compare and contrast the different CPU scheduling algorithms (e.g., Round Robin, Shortest Job First, and Priority Scheduling). What are their strengths and weaknesses?
5. Analyze a system where deadlock has occurred. Based on the resource allocation graph, identify the processes involved in the deadlock.
6. Compare and contrast the effectiveness of Peterson's solution and hardware-based synchronization mechanisms in solving the critical-section problem.
7. Analyze how segmentation can affect the allocation of memory for processes with different

memory requirements.

8. Compare the different file access methods (sequential, direct, and indexed) and their suitability for various types of applications.
9. Compare the efficiency of different disk scheduling algorithms (e.g., FCFS, SSTF, SCAN, C-SCAN) in terms of average seek time and throughput.
10. Analyze the advantages and disadvantages of using different RAID levels (e.g., RAID 0, RAID 1, RAID 5) in terms of performance, redundancy, and cost.
11. Break down how the principles of protection are applied in a multi-user system to ensure fairness and security.
12. Analyze the security vulnerabilities that can arise from program threats versus network threats and recommend strategies to mitigate them.

L5: Evaluating

1. Evaluate the impact of free and open-source software on the development of modern operating systems.
2. Based on a particular computing environment (e.g., mobile vs. desktop OS), assess the suitability of specific operating systems for that environment.
3. Evaluate the efficiency of CPU scheduling algorithms in a real-time operating system.
4. Critically evaluate the fairness of the Round Robin scheduling algorithm in a system with diverse process priorities & evaluate the efficiency of the Shortest Job First (SJF) scheduling algorithm in a system with mixed process types (short and long processes).
5. Evaluate the advantages and disadvantages of using semaphores over mutex locks in process synchronization.
6. Assess the effectiveness of deadlock avoidance techniques in real-time systems. Which strategy would you recommend for a highly dynamic environment?
7. Evaluate the use of different page-replacement algorithms (e.g., FIFO, LRU, Optimal) in terms of efficiency and system performance.
8. Evaluate the effectiveness of demand paging versus preloading all necessary pages into memory at the beginning of a program's execution.
9. Evaluate how file protection mechanisms (such as access control lists) can be integrated into a file system to prevent unauthorized access.
10. Evaluate a system's vulnerability to program, system, and network threats and propose a security enhancement plan.
11. Evaluate the security strengths and weaknesses of the access matrix model in terms of scalability and flexibility.

L6: Create

1. Design a new microkernel-based operating system for a smart IoT device. Describe the architecture, essential system services, and justify your design choices with respect to security and performance.
2. Create a CPU scheduling simulator that allows the user to switch between different algorithms (FCFS, SJF, RR, Priority). Include multithreading support and compare the performance for different workloads.
3. Develop a solution to a new synchronization problem in a multi-robot warehouse environment using semaphores or monitors. Justify how your solution avoids deadlocks and ensures resource fairness.
4. Design a hybrid memory management scheme for a custom OS that combines paging and segmentation to optimize memory usage in embedded systems. Provide diagrams and justify your approach.
5. Develop a security framework for a cloud-based file storage system that integrates protection principles, access control, and encryption. Illustrate how your framework defends against both system and network threats.

**Chairperson
Board of Studies (CSE)**

R24CSPC12**DATABASE MANAGEMENT SYSTEMS LAB****0 0 3 1.5**

(Common to CSE, CSE (DS), CSE (AI &ML))

Course Objectives:

1. Populate and query a database using SQL DDL/DML Commands
2. Declare and enforce integrity constraints on a database
3. Writing Queries using advanced concepts of SQL
4. Programming PL/SQL including procedures, functions, cursors and triggers

Course Outcomes

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

Course Code	Course Outcomes	Mapping with POs and PSOs													
		PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	DoK
R24CSPC 12.1	Utilizing Data Definition Language (DDL), Data Manipulation Language (DML), and Data Control Language (DCL) commands effectively within a database environment	3	2	2	1	3	1	1	1	2	1	2	3	1	L1, L2
R24CSPC 12.2	Constructing and execute queries to manipulate and retrieve data from databases	3	3	2	2	3	1	1	1	2	1	2	3	1	L2, L3
R24CSPC 12.3	Develop application programs using PL/SQL. Establish database connectivity through JDBC (Java Database Connectivity)	2	2	3	2	3	1	1	2	2	2	2	2	3	L3

Board of Studies: Computer Science and Engineering

Approved in BOS No: 02, 9th May, 2025

Approved in ACM No: 02

Developing the following programs:**Week 1:****CO's:CO1**

1. Creation, altering and dropping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.

Week 2:**CO's:CO1**

2. Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION,

INTERSECT, Constraints. Example:- Select the roll number and name of the student who secured fourth rank in the class.

Week 3: **CO's:CO1**

3. Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.

Week 4: **CO's:CO1**

4. Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char, to_date)

Week 5: **CO's:CO1**

5.
 - i. Create a simple PL/SQL program which includes declaration section, executable section and exception –Handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)
 - ii. Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.

Week 6: **CO's:CO2**

6. Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.

Week 7: **CO's:CO2**

7. Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exception RAISEAPPLICATION ERROR.

Week 8: **CO's:CO2**

8. Programs development using creation of procedures, passing parameters IN and OUT of PROCEDURES.

Week 9: **CO's:CO2**

9. Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions.

Week 10: **CO's:CO2**

10. Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.

Week 11: **CO's:CO3**

11. Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and

INSTEAD OF Triggers

Week 12: **CO's:CO3**

12. Create a table and perform the search operation on table using indexing and no indexing techniques.

Week 13: **CO's:CO3**

13. Write a Java program that connects to a database using JDBC

Week 14: **CO's:CO3**

14. Write a Java program to connect to a database using JDBC and insert values into it

Week 15: **CO's:CO3**

15. Write a Java program to connect to a database using JDBC and delete values from it

Reference Books:

1. Oracle: The Complete Reference by Oracle Press
2. Nilesh Shah, "Database Systems Using Oracle", PHI, 2007
3. Rick F Vander Lans, "Introduction to SQL", Fourth Edition, Pearson Education, 2007
4. RamezElmasri, Shamkant, B. Navathe, "Database Systems", Pearson Education, 6th Edition, 2013.
6. Database Principles Fundamentals of Design Implementation and Management, 10th edition, Carlos Coronel, Steven Morris, Peter Robb, Cengage Learning, 2022

Online Learning Resources:

1. <http://www.scoopworld.in>
2. <http://vlabs.iitb.ac.in/vlabs-dev/labs/dblab/index.php>

**Chairperson
Board of Studies (CSE)**

R24CSPC13**Operating Systems Lab**
(Computer Science and Engineering)**0 0 3 1.5****Course Objectives:**

1. To understand the basic system calls related to process
2. To implement various CPU scheduling algorithms
3. To implement memory management and virtual memory techniques
4. To implement file management and disk scheduling algorithms

Course Outcomes

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

Mapping with POs and PSOs					DoK
Course Code	Course Outcomes (COs)	PO1	PO2	PO3	
R24CSPC13.1	Know the basic system calls related to process	3	2	1	L1
R24CSPC13.2	Simulate CPU scheduling algorithms, disk scheduling algorithms	3	3	2	L2
R24CSPC13.3	Understand memory management and virtual memory techniques, file management techniques	3	3	3	L3

Board of Studies: Computer Science and Engineering

Approved in BOS No: 02, 9th May, 2025

Approved in ACM No: 02

Developing the following programs:**Week 1:**

1. Practicing of Basic UNIX Commands.

CO's:CO1**Week 2:**

2. Write programs using the following UNIX operating system calls fork, exec, getpid, exit, wait, close, stat, opendir and readdir

CO's:CO1**Week 3:**

3. Simulate the following CPU scheduling algorithms a) FCFS b) SJF c) Priority d) Round Robin

CO's: CO2**Week 4:**

4. Control the number of ports opened by the operating system with a) Semaphore b) Monitors.

CO's: CO2**Week 5:**

5. Write a program to solve producer-consumer problem using Semaphores.

CO's: CO2

Week 6:

6. Implement the following memory allocation methods for fixed partition a) First fit b) Worst fit c) Best fit **CO's:CO3**

Week 7:

7. Simulate the following page replacement algorithms a) FIFO b) LRU c) LFU **CO's:CO3**

Week 8:

8. Simulate Paging Technique of memory management. **CO's:CO3**

Week 9:

9. Implement Bankers Algorithm for Dead Lock avoidance and prevention **CO's: CO2**

Week 10:

10. Simulate the following file allocation strategies a) Sequential b) Indexed c) Linked **CO's:CO5**

Week 11:

11. Simulate the following Disk scheduling algorithms. a) FCFS b) SSTF c) SCAN **CO's:CO4**

Week 12:

12. Simulate the following Disk scheduling algorithms. a) C-SCAN b) LOOK c) C-LOOK **CO's:CO4**

Textbooks:

1. Operating System Concepts, Abraham Silberchatz, Peter B. Galvin, Greg Gagne, 7th Edition, John Wiley
2. Operating Systems- A Concept based Approach, D. M. Dhamdhare, 2nd Edition, Tata McGraw-Hill

Reference Books:

1. Operating Systems-Internal and Design Principles, Stallings, 5th Edition, Pearson Education/PHI, 2005
2. UNIX: Concepts and Applications, Sumitabha Das, 4th Edition, Tata McGraw-Hill
3. Operating System: A Design-Oriented Approach, Charles Patrick Crowley
4. Operating Systems: A Modern Perspective, Gary J. Nutt
5. Design of the Unix Operating Systems, Maurice J. Bach

**Chairperson
Board of Studies (CSE)**

R24CSPC14**Advanced Coding using C & JAVA Lab****0 0 3 1.5**

(Computer Science and Engineering)

Course Objectives:

1. The objective of this course is to provide students with hands-on experience and advanced knowledge in programming through the use of multiple languages, including C (Data Structures (DS)), and JAVA.
2. The course aims to develop problem-solving skills, enhance algorithmic thinking, and train students in the practical implementation of core programming concepts across different paradigms and environments.

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

Course Code	Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	Dok
R24CSPC 14.1	Apply logical techniques such as pattern recognition, sequencing, and classification to solve problems.	3	3	2	2	2	-	-	-	1	1	-	3	2	L3,L4
R24CSPC 14.2	Implement and manage linear and nonlinear data structures (like stacks, queues, linked lists, trees, and graphs).	3	3	3	2	3	-	-	-	2	1	1	3	2	L4,L6
R24CSPC 14.3	Solve complex real-world problems using object-oriented programming concepts in Java.	3	3	3	3	3	2	1	1	3	3	3	2	3	L5,L6

Board of Studies: Computer Science and Engineering

Approved in BOS No: 02, 9th May, 2025

Approved in ACM No: 02

Developing the following programs:**C & DATA STRUCTURES****WEEK-1****CO's: CO1**

1. The given function has a string (str) and two characters, ch1 and ch2. Execute the function in such a way that str returns to its original string, and all the events in ch1 are replaced by ch2, and vice versa.

Replace character (Char str1, Char ch1, Int 1, Char ch2)

Assumption

This string has only alphabets that are in lower case.

Example

Input:

str: apples

ch1: a

ch2: p

Output:

paales

Explanation: All the 'a' in the string is replaced with 'p'. And all the 'p's are replaced with 'a'.

WEEK-2

COs: CO1

2. In an enchanted garden, flowers bloom in various colors, and each color has a specific point value. Your task is to calculate the total points based on the flowers picked by a gardener.

Input Format

A list of integers representing the points for each flower.

Output Format

An integer representing the total points earned from the flowers.

Example

Input: [5, 3, 8, 2]

Output: 18

WEEK-3

CO's: CO2

3. Count nodes in circular linked list

Algorithm for count nodes in circular linked list

A-Accept the head pointer in the function

B-Declare the count variable which maintains the count of nodes and initialize it to zero.

C- Traverse the link till next node is the head node

D- Define a Node class which represents a node in the list. It has two properties data and next which will point to the next node.

E-Define another class for creating the circular linked list and it has two nodes: head and tail. It has two methods: add () and display ().

WEEK-4

CO's: CO2

4. Implement a binary search tree (BST) and perform operations such as insertion, deletion, and searching.

WEEK-5

CO's: CO2

5. Implement a scenario-based program for a social network using graph data structures.

WEEK-6

CO's: CO1

6. Develop a scenario-based program to simulate a ticket booking system using queues.

WEEK-7

CO's: CO2

7. Implement graph data structure and perform depth-first search (DFS) and breadth-first search (BFS).

WEEK-8

CO's: CO1

8. The function accepts string str of size n as an argument. Implement the function which returns 1 if given string str is valid password else 0. str is a valid password if it satisfies the below conditions.

- – At least 4 characters
- – At least one numeric digit
- – At Least one Capital Letter
- – Must not have space or slash (/)
- – Starting character must not be a number

Assumption:

Input string will not be empty.

Example:

Input 1:

aA1_67

Input 2:

a987 abC012

Output 1:

1

Output 2:

0

WEEK-9

CO's: CO1

9. The function accepts an integer array 'arr', its length and two integer variables 'num' and 'diff'. Implement this function to find and return the number of elements of 'arr' having an absolute difference of less than or equal to 'diff' with 'num'. Note: In case there is no element in 'arr' whose absolute difference with 'num' is less than or equal to 'diff', return -1.

Example:

Input:

- arr: 12 3 14 56 77 13
- num: 13
- diff: 2

Output: 3

WEEK-10**CO's: CO1**

10. Write a function to validate if the provided two strings are anagrams or not. If the two strings are anagrams, then return 'yes'. Otherwise, return 'no'.

Input:

Input 1: 1st string

Input 2: 2nd string

Output:

(If they are anagrams, the function will return 'yes'. Otherwise, it will return 'no'.)

Example

Input 1: Listen

Input 2: Silent

Output:

Yes

Programming through JAVA**WEEK-11****CO's: CO3****11. Problem Statement:**

In a classroom some of the seats are already occupied by students and only a few seats are available in the classroom. The available seats are assumed as **r** and **n** number of students is looking for the seat. We need to find in how many different permutations **n** number of students can sit on **r** number of chairs.

Algorithm

- Input number of students in **n**
- Input number of seats in **r**
- Use permutation formula { $\text{factorial}(n) / \text{factorial}(n-r)$ }
- Print Output

WEEK-12**CO's: CO3**

12. Find all possible Palindromic Partitions of the given String in Java.

WEEK-13**CO's: CO3**

13. Write a Java program to remove a specific element from an array.

WEEK-14**CO's: CO3**

14. Develop a Java application that demonstrates comprehensive file handling operations. The application should perform the following tasks:

- Create a new file
- Read the contents of the file and display it on the console.
- Append additional text to the existing content of the file.

Delete the original file

WEEK-15

CO's: CO3

15. Can a number be expressed as a sum of two prime numbers using java

Here, we will discuss program to check whether a number be expressed as a sum of two prime numbers using java. We will ask the user to enter a positive integer and check whether that number can be expressed as the sum of two prime numbers. If that number can be expressed as sum of two prime numbers then print the numbers in the output and else print the statement that the number cannot be expressed as a sum of two prime numbers.

WEEK-16

CO's: CO3

16. Write a Java program to compare two files lexicographically.

According to Wikipedia:

In mathematics, the lexicographic or lexicographical order (also known as lexical order, dictionary order, alphabetical order or lexicographic (al) product) is a generalization of the way the alphabetical order of words is based on the alphabetical order of their component letters. This generalization consists primarily in defining a total order over the sequences (often called words in computer science) of elements of a finite totally ordered set, often called alphabet.

Textbooks:

1. Data structures Using C Second Edition, Reema Thareja.
2. Programming Problem Solving, Reema Thareja.
3. Herbert Schildt, Java The complete reference, 11th Edition, McGrawHill, 2019.
4. Timothy budd, An introduction to object-oriented programming, 3rd Edition, Pearson, 2009.

Reference Books:

1. Lets Us C, Yeswanth Kanethkar.
2. Data Structures, S.Chand And Company Limited, Dr.KV Sambasivarao.
3. Y. Daniel Liang Introduction to Java Programming Comprehensive Version, 10th Edition, Pearson, 2015.
4. Bruce Eckel, Thinking in Java, 4th Edition, Prentice Hall, 2006.

Chairperson
Board of Studies (CSE)

Course Objectives:

1. To implement the client side of the web application using java-script.
2. To understand Java script on the desktop using NodeJS.
3. To develop a web application using NodeJS and Express.
4. To implement a SPA using React.
5. To develop a full stack single page application using React,ode JS, and a Database (MongoDB or SQL).

Course Outcomes

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

Mapping with POs and PSOs																DoK
Course Code	Course Outcomes (COs)	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	
24SC02.1	Implement the client side of the web application.	2	1	2	1	3	1	0	2	3	2	1	-	2	1	L1, L2
24SC02.2	Deploy server side applications using NodeJS. Use express framework in web development	2	2	3	2	3	1	0	2	2	2	3	-	3	2	L2, L3
24SC02.3	Implement and architect database systems. Develop a full stack single page applications	3	2	3	2	3	2	1	3	3	3	3	-	3	2	L2, L3

Board of Studies: Computer Science and Engineering

Approved in BOS No: 02, 09th May, 2025

Approved in ACM No: 02

Experiments covering the Topics:

- Lists, Links and Images
- HTML Tables, Forms and Frames
- HTML 5 and Cascading Style Sheets, Types of CSS
- Selector forms
- CSS with Color, Background, Font, Text and CSS Box Model
- Applying JavaScript - internal and external, I/O, Type Conversion
- JavaScript Conditional Statements and Loops, Pre-defined and User-defined Objects

- JavaScript Functions and Events

Developing the following programs:

Week-1:

1. Write a HTML program, to explain the working of lists. **CO's:CO1**
2. Write a HTML program, to explain the working of hyperlinks using <a> tag and href, target attributes. **CO's:CO1**
3. Create a HTML document that has your image and your friend's image with a specific height and width. Also when clicked on the images it should navigate to their respective profiles. **CO's:CO1**
4. To create Form validation using JavaScript. **CO's:CO1**

Week-2:

1. Write a HTML program, to explain the working of tables. (use tags: <table>, <tr>, <th>, <td> and attributes: border, rowspan, colspan) **CO's:CO2**
2. Write a HTML program, to explain the working of forms by designing Registration form. **CO's:CO2**
3. Get data using Fetch API from an open-source endpoint and display the contents in the form of a card. **CO's:CO2**

Week-3:

1. Write a HTML program, that makes use of <article>, <aside>, <figure>, <figcaption>, <footer>, <header>, <main>, <nav>, <section>, <div>, tags. **CO's:CO2**
2. Write a HTML program, to embed audio and video into HTML web page. **CO's:CO2**
3. Write a program to apply different types (or levels of styles or style specification formats) inline, internal, external styles to HTML elements. (Identify selector, property and value). **CO's:CO2**

Week-4:

1. Write a program to apply different types of selector forms
 - i. Simple selector (element, id, class, group, universal)
 - ii. Combinator selector (descendant, child, adjacent sibling, general sibling)
 - iii. Pseudo-class selector
 - iv. Pseudo-element selector
 - v. Attribute selector **CO's:CO2**

Week-5:

1. Write a program to demonstrate the various ways you can reference a color in CSS. **CO's:CO2**
2. Write a CSS rule that places a background image halfway down the page, tilting it horizontally. The image should remain in place when the user scrolls up or down. **CO's:CO2**
3. Write a program using the following terms related to CSS font and text:
 - i. Font-size
 - ii. Font-weight
 - iii. font-style

iv. text-decoration v. text-transformation VI. Text-alignment CO's:CO2

4. Write a program, to explain the importance of CSS Box model using

i. Content ii. Border iii. Margin iv. Padding CO's:CO2

Week-6:

1. Write a program to embed internal and external JavaScript in a web page. CO's:CO3

2. Write a program to explain the different ways for taking input & displaying output..

CO's:CO3

3. Create a webpage which uses prompt dialogue box to ask a voter for his name and age.
Display the information in table format along with either the voter can vote or not

CO's:CO3

Week-7:

1. Write a program using document object properties and methods.

2. Write a program using window object properties and methods.

3. Write a program using array object properties and methods.

4. Write a program using math object properties and methods.

5. Write a program using string object properties and methods.

CO's:CO3

Week-8:

1. Write a program to display week days using switch case.

2. Write a program to print 1 to 10 numbers using for, while and do-while loops.

3. Write a program to print data in object using for-in, for-each and for-of loops Develop a program to determine whether a given number is an 'ARMSTRONG NUMBER' or not.

4. Write a program to display the denomination of the amount deposited in the bank in terms of 100's, 50's, 20's, 10's, 5's, 2's & 1's. (Eg: If deposited amount is Rs.163, the output should be 1-100's, 1-50's, 1- 10's, 1-2's & 1-1's)

CO's:CO3

Week-9:

1. Design a appropriate function should be called to display

I. Factorial of that number

II. Fibonacci series up to that number

III. Prime numbers up to that number

IV. Is it palindrome or not

CO's:CO3

Week-10:

1. Design a HTML having a text box and four buttons named Factorial, Fibonacci, Prime, and Palindrome. When a button is pressed an appropriate function should be called to display

- I. Factorial of that number
- II. Fibonacci series up to that number
- III. Prime numbers up to that number
- IV. Is it palindrome or not

CO's:CO3

Week-11:

1. Write a program to validate the following fields in a registration page
 - I. Name (start with alphabet and followed by alphanumeric and the length should not be less than 6 characters)
 - II. Mobile (only numbers and length 10 digits)
 - III. E-mail (should contain format like xxxxxxx@xxxxxx.xxx).

CO's:CO3

Week-12:

1. Create a form and validate the contents of the form using JavaScript. **CO's: CO3**
2. Get data using Fetch API from an open-source endpoint and display the contents in the form of a card. **CO's: CO3**
3. Create a Node JS server that serves static HTML and CSS files to the user without using express. **CO's: CO3**

Week-13:

1. Create a NodeJS server using Express that stores data from a form as a JSON file and displays it in another page. The redirect page should be prepared using Handlebars. **CO's:CO3**
2. Create a NodeJS server using Express that creates, reads, updates and deletes students' details and stores them in MongoDB database. The information about the user should be obtained from a HTML form. Conversion of Fahrenheit to Celsius and vice versa.

CO's:CO3

Week-14:

1. Create a NodeJS server that creates, reads, updates and deletes event details and stores them in a MySQL database. The information about the user should be obtained from a HTML form. **CO's:CO3**
2. Create a counter using ReactJS **CO's:CO3**
3. Create a Todo application using ReactJS. Store the data to a JSON file using a simple Node JS server and retrieve the information from the same during page reloads. **CO's:CO3**

Week-15:

1. Create a simple Sign up and Login mechanism and authenticate the user using cookies. The user information can be stored in either MongoDB or MySQL and the server should be built

using NodeJS and Express Framework.

CO's:CO3

2. Create and deploy a virtual machine using a virtual box that can be accessed from the host computer using SSH. Write a C program to print last n characters of a given file.

CO's:CO3

3. Create a docker container that will deploy a NodeJS ping server using the NodeJS image.

CO's:CO3

Textbooks:

1. John Dean, Web Programming with HTML5, CSS and JavaScript, Jones & Bartlett Learning, 2019.

Reference Books:

1. Programming the World Wide Web, 7th Edition, Robert W Sebesta, Pearson, 2013.
2. Pro MERN Stack: Full Stack Web App Development with Mongo, Express, React, and Node, Vasan Subramanian, 2nd edition, APress, O'Reilly.

Online Learning Resources:

1. <https://www.w3schools.com/html>
2. <https://www.w3schools.com/css>
3. <https://www.w3schools.com/js/>
4. <https://www.w3schools.com/nodejs>
5. <https://www.w3schools.com/typescript>

**Chairperson
Board of Studies (CSE)**

R24HS05	Numerical Ability & Professional Communication skills	0	0	2	1
(Common to all UG Programmes)					

Course Objectives:

1. **Develop problem-solving skills:** Students will be able to apply numerical concepts and techniques to solve problems in various contexts.
2. **Enhance data analysis skills:** Students will be able to collect, analyze, and interpret numerical data to inform decision-making.
3. **Develop effective communication skills:** Students will be able to communicate effectively in various professional settings, including verbal and non-verbal communication.
4. **Improve presentation skills:** Students will be able to deliver clear, concise, and engaging presentations to different audiences.
5. **Enhance writing skills:** Students will be able to write clear, concise, and well-structured professional documents, such as reports, emails, and memos.
6. **Develop interpersonal skills:** Students will be able to build and maintain effective relationships with colleagues, clients, and stakeholders.

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

Course Code	Course Outcomes	PO1	PO7	PO9	P011	Dok
R24HS05.1	Use their logical thinking and analytical abilities to solve Quantitative aptitude questions from company specific and other competitive tests..	2	---	---	---	L4
R24HS05.2	Solve questions related to time and distance and time and work etc. from company specific and other competitive tests.	1	---	---	---	L5
R24HS05.3	Comprehend the essentiality of LSRW skills in paper presentations, seminars, workshops, conferences etc. with teams and to solve different types of questions based on vocabulary, structure, grammar and verbal reasoning.	2	2	3	1	L1, L2
R24HS05.4	Attain the knowledge of soft skills in various conditions, Solve questions based on sentence completion and fill in the blanks	---	1	3	---	L2, L3
R24HS05.5	Explore diverse fields through English , To solve different types of questions based on vocabulary, structure, grammar and verbal reasoning	---	---	3	---	L4

SYLLABUS**UNIT-I: Numerical computation****15 Hours**

Application based on Numbers –Classification of Number System, Prime and Composite, Even and Odd Numbers, Divisibility Rule, BODMAS Rule, Unit digit, fractions, LCM &HCF. Percentages and its Applications, Profit Loss and Discount, Simple interest and Compound interest. Averages, Ratio Proportion, Application of Ratios, Partnerships, Shares and dividends.

Self Learning Topic: Problems on Ages

UNIT-II: Numerical Estimation**15 Hours**

Time and work, Application of Time-work (pipes and cisterns) , Time and Distance, circular Tracking, concept of Boats & steams. Mixtures and allegations, application of percentage and Ratios and Averages in Mixtures

Self Learning Topic: Problems on Trains

UNIT-III**5 Hours**

Abstract Preparation – Noticing Key Words –Literature Survey – Using Academic Verbs Verbal Ability: Sentence correction, Resume Writing and Cover letter

Self Learning Topic: Project Report

UNIT-IV**5 Hours**

Organizational Skills – Time Management – IELTS Test Papers Exercises Verbal Ability: sentence completion

Self Learning Topic: Jumbled Sentences

UNIT-V**5 Hours**

Analogy, origin of Words, Idioms and phrases, MNC's question papers Verbal Ability: Error Identification

Self Learning Topic: Speaking and writing diction (Choice of Words)

Board of Studies: MBA

Approved in BOS No: 02, 13th May, 2025

Approved in ACM No: 02

Text Books:

1. Quantitative Aptitude for competitive examinations by R.S.Agrawal - S.Chand publications.
2. Business Communication by C.S.G Krishnamacharyulu & Lalitha Ramakrishna – Himalaya publishing house

Reference Books:

1. Numerical Aptitude and Reasoning Ability tests in Competitive exams by Prof.A.Balasubramanian.
2. Numerical Ability and Mathematical Aptitude by Dr.A.B.Rao
3. Communication skills and soft skills by E.Suresh Kumar , p.sreehari & j.savithri

Web References:

1. <https://faceprep.in/logical> reasoning

2. www.sawaal.com
3. <http://nptel.ac.in/courses>
4. www.prepinsta.com

Model Questions from Numerical Ability

1. What is the unit digit of 2^{10} ?
2. Find the unit digit of 17^5 .
3. What is the unit digit of $3 \times 7 \times 9 \times 11$?
4. Determine the unit digit of 123^4 .
5. Find the unit digit of $2^{20} + 3^{15}$.
6. Is 432 divisible by 3?
7. Check if 756 is divisible by 9.
8. Determine if 1240 is divisible by 8.
9. Is 531 divisible by 11?
10. Check if 2700 is divisible by 15.
11. Simplify: $2 \times 3 + 12 \div 4 - 5$
12. Evaluate: $18 - 3 \times 2 + 12 \div 4$
13. Simplify: $9 + 8 \div 2 \times 3 - 1$
14. Evaluate: $15 - 2 \times 3 + 8 \div 2$
15. Simplify: $24 \div 4 \times 2 + 10 - 3$
16. Find the HCF of 12 and 18.
17. Determine the HCF of 24 and 30.
18. The HCF of two numbers is 6 and their LCM is 72. What are the two numbers?
19. The LCM of two numbers is 120 and their HCF is 10. If one of the numbers is 24, what is the other number?
20. A shirt is marked at \$80 and sold at a 20% discount. What is the selling price?
21. A student scored 75% marks in an exam. If the total marks were 200, how many marks did the student score?
22. A number is increased by 25% and then decreased by 10%. What is the net percentage change?
23. A book is bought for \$50 and sold for \$60. What is the profit percentage?
24. A shopkeeper sells an item at a 15% loss. If the cost price is \$100, what is the selling price?
25. A person buys a product for \$80 and sells it for \$100. What is the profit percentage?
26. Find the simple interest on \$1000 at 5% per annum for 2 years.
27. A sum of money amounts to \$1200 in 3 years at 8% per annum simple interest. What is the principal amount?
28. What is the rate of interest if a sum of \$500 amounts to \$600 in 4 years at simple interest?

Model Questions from professional communication skills

1. You're working with a team member who has a different work style. How would you adapt your communication approach?
2. A client is unhappy with the project's progress. How would you handle the situation?
3. You're part of a team, and there's a conflict between two members. How would you facilitate a resolution?
4. How would you use body language to convey confidence during a presentation?
5. You're in a virtual meeting, and your internet connection is unstable. How would you

handle the situation?

6. You're meeting a client for the first time. What nonverbal cues would you use to build rapport?
7. You're presenting a project update to a client. How would you structure your presentation to ensure effective communication?
8. A colleague is not meeting their deadlines, affecting the team's project timeline. How would you approach the conversation?
9. You're in a meeting, and a team member is dominating the discussion. How would you politely intervene?
10. You're tasked with writing a report on a project's progress. How would you structure the report?

Chairperson
Board of Studies (MBA)

R24MC04**INTELLECTUAL PROPERTY RIGHTS****3 0 0 3**

(Common to CSE, CSE (DS), CSE (AI&ML), ECE, EEE, ME)

Course Objectives:

The main objectives of the course are to:

1. To introduce the fundamental concepts, types, and importance of Intellectual Property Rights (IPR) in the modern knowledge economy.
2. To understand the legal framework and procedures for protecting and enforcing patents, copyrights, trademarks, and other IP forms.
3. To enable students to apply IPR knowledge in managing and safeguarding intellectual assets in business and research environments.
4. To analyze case studies involving IPR infringement, disputes, and resolution mechanisms in various industries.
5. To create awareness about emerging IPR challenges related to digital innovation, artificial intelligence, and global trade agreements.

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

Course Code	Course Outcomes	Mapping with POs										DoK
		PO1	PO2	PO3	PO4	PO5	PO6	PO8	PO9	PO10	PO11	
R24MC04.1	Understand the Fundamentals of IPR	3	2	-	-	-	2	3	-	2	-	L1, L2
R24MC04.2	Analyze Legal Aspects of Patents and Copyrights	3	3	2	-	2	-	3	-	2	-	L2, L3
R24MC04.3	Evaluate Trademarks and Industrial Design Protection	3	3	2	-	2	-	3	-	2	-	L2, L4
R24MC04.4	Examine the Significance of GIs and Trade Secrets	3	2	-	-	2	2	3	-	2	-	L4, L5
R24MC04.5	Apply IPR Laws and Address Emerging Challenges	3	3	3	3	3	2	3	2	3	-	L5, L6

SYLLABUS**UNIT I: Introduction to Intellectual Property Rights****9 Hours**

Concept and evolution of Intellectual Property Rights (IPR) – types and significance of IPR: patents, copyrights, trademarks, trade secrets, industrial designs, and geographical indications; nature and characteristics of IPR; historical development and global perspectives; importance of IPR in innovation and knowledge economy.

CO's-CO1

Self Learning Concepts: Explore the concept and evolution of Intellectual Property Rights (IPR), including patents, copyrights, trademarks, industrial designs, trade secrets, and geographical indications. Understand their significance in protecting innovation and intellectual assets. Study the

historical development of IPR and its role in the knowledge economy, fostering innovation and creativity in industries.

UNIT II: Patents and Copyrights

12 Hours

Patents-definition, features, types, patentable inventions, non-patentable inventions, procedure for filing, rights of patentees, and infringement issues; Copyrights-meaning, subject matter, rights of authors, duration and protection, exceptions and limitations, fair use, and remedies for infringement. **CO's-CO2**

Self Learning Concepts: Learn about patents, including the types of inventions that can be patented, the filing procedure, and the rights of patentees. Understand copyright protection, the rights of authors, and the duration of protection. Explore the concept of fair use and remedies for copyright infringement, alongside the limitations and exceptions within copyright law.

UNIT III: Trademarks and Industrial Designs

12 Hours

Trademarks – concept, types, functions, registration process, rights of the holder, infringement and remedies; Industrial designs – concept, features, registration procedure, rights conferred, term of protection, and case studies on design rights violations. **CO's-CO3**

Self Learning Concepts: Focus on trademarks and their role in brand protection, including registration, rights of holders, and infringement issues. Study the protection of industrial designs, registration procedures, and the term of protection. Review case studies on design rights violations and the process of trademark registration.

UNIT IV: Geographical Indications and Trade Secrets

14 Hours

Geographical Indications (GI) – meaning, significance, registration procedure, rights granted, famous Indian GIs; Trade secrets – definition, scope, protection mechanisms, trade secret misappropriation, and differences between patents and trade secrets; relevance of GIs and trade secrets in business and commerce. **CO's-CO4**

Self Learning Concepts:Examine Geographical Indications (GI) and their importance in protecting regional products. Understand the GI registration process and famous Indian GIs. Learn about trade secrets, their legal protection, and the differences between patents and trade secrets, especially in sectors like technology and business.

UNIT V: IPR Laws, Organizations and Emerging Issues

16 Hours

Overview of Indian IPR laws – key provisions of the Patents Act, Copyright Act, Trademarks Act, and GI Act; international IPR organizations – WIPO, WTO, TRIPS Agreement; IPR management in industries and academic institutions; contemporary issues – digital piracy, IPR in biotechnology, AI and IPR, and IPR challenges in the digital era. **CO's-CO5**

Self Learning Concepts: Study Indian IPR laws, including provisions in the Patents Act, Copyright Act, Trademarks Act, and GI Act. Learn about global IPR organizations like WIPO and

WTO, and the TRIPS Agreement. Explore emerging IPR issues such as digital piracy, IPR in biotechnology, and challenges related to AI and IPR in the digital age.

Board of Studies : Master of Business Administration

Approved in BoS No : 02, 13th May, 2025

Approved in ACM No: 02

Text Books:

- 1) **Neeraj Pandey & Khushdeep Dharni** – Intellectual Property Rights, PHI Learning.
- 2) **P. Narayanan** – Law of Intellectual Property, Eastern Law House.

Reference Books:

- 1) **Deborah E. Bouchoux** – Intellectual Property: The Law of Trademarks, Copyrights, Patents, and Trade Secrets, Cengage Learning.
- 2) **T. Ramakrishna** – Basic Principles and Acquisition of Intellectual Property Rights, Notion Press.
- 3) **N.S. Gopalakrishnan & T.G. Agitha** – Principles of Intellectual Property, Eastern Book Company.

Web References:

- 1) www.ipindia.gov.in – Official website of the Indian IPR Office.
- 2) www.wipo.int – World Intellectual Property Organization.
- 3) www.wto.org – World Trade Organization – TRIPS Agreement.
- 4) www.indianiprsurvey.in – IPR awareness and industry surveys.
- 5) www.mhrd.gov.in – IPR cell initiatives under the National IPR Policy.

Internal Assessment Pattern

Cognitive Level	Internal Assessment # 1 (%)	Internal Assessment # 2 (%)
L1	13	-
L2	38	10
L3	12	10
L4	25	30
L5	-	30
L6	12	10
Total (%)	100	100

Sample Short and Long Answers to Questions of Various Cognitive Levels

L1: Remember

1. What is the full form of IPR?
2. List any four types of Intellectual Property Rights.
3. What are patentable inventions?
4. Define a trademark.
5. Mention any two famous Indian Geographical Indications.
6. What is the term of protection for industrial designs?

7. State the main function of the World Intellectual Property Organization (WIPO).
8. Name the law that governs copyrights in India.
9. What is a trade secret?
10. List any two international agreements related to IPR.

L2: Understand

1. Explain the significance of IPR in the knowledge economy.
2. Differentiate between patentable and non-patentable inventions.
3. Describe the rights granted to a copyright holder.
4. Discuss the importance of geographical indications in protecting local heritage.
5. Explain how a trademark helps in brand recognition.
6. Clarify the concept of “fair use” under copyright law.
7. Interpret the difference between patents and trade secrets.
8. Describe the process for registering an industrial design.
9. Explain the function of the TRIPS Agreement.
10. Discuss how IPR supports innovation.

L3: Apply

1. Apply the knowledge of patents to identify protectable aspects of a new product.
2. Demonstrate how to check trademark availability for a business logo.
3. Use the concept of GI to suggest a registration for a local craft product.
4. Illustrate a situation where trade secret protection is more suitable than patenting.
5. Apply copyright law to determine if a user is violating fair use.
6. Identify which type of IPR would apply to a mobile app design.
7. Use an example to show how IPR can prevent product imitation.
8. Demonstrate the steps for filing a patent in India.
9. Apply IPR knowledge to analyze a case of music piracy.
10. Show how industries can manage IPR to protect their innovations.

L4: Analysing

1. Compare and contrast the protection mechanisms of patents and trade secrets.
2. Analyze how digital technologies have challenged copyright enforcement.
3. Examine the role of IPR in promoting economic growth.
4. Break down the key differences between trademarks and industrial designs.
5. Investigate the impact of geographical indications on rural economies.
6. Distinguish between moral rights and economic rights under copyright law.
7. Analyze a real case where trademark infringement occurred and its resolution.
8. Explore the challenges of enforcing IPR laws in the digital era.
9. Examine the role of WIPO and WTO in shaping global IPR policies.
10. Compare Indian IPR laws with those of another country in one IPR domain.

L5: Evaluating

1. Evaluate the effectiveness of India's patent laws in protecting inventions.
2. Assess whether trade secrets offer better protection than patents for tech startups.
3. Critique the fairness of copyright exceptions under "fair use."
4. Judge the impact of digital piracy on the creative industry.
5. Argue for or against compulsory licensing in pharmaceuticals.
6. Evaluate the relevance of geographical indications in the global market.
7. Appraise the challenges industries face in enforcing design rights.
8. Debate whether AI-generated content should be protected under IPR.
9. Justify stronger IPR laws to encourage academic research and innovation.
10. Review the pros and cons of current international IPR agreements for developing countries.

L6: Create and Develop

1. Design a proposal to raise awareness about GI protection in rural India.
2. Develop a guidebook for startups on managing their IPR portfolio.
3. Formulate a business strategy that incorporates trade secrets for competitive advantage.
4. Create a short case study on a successful use of IPR in product commercialization.
5. Construct a training module for students on patent filing.
6. Propose a digital solution for detecting copyright infringement.
7. Draft a sample copyright notice for an e-learning platform.
8. Build an IPR audit plan for a small software company.
9. Design a new trademark and justify its eligibility for registration.
10. Create a campaign to educate musicians on their copyright rights.

**Chairperson
Board of Studies (MBA)**