Bansilal Ramnath Agarwal Charitable Trust's

Vishwakarma Institute of Technology

(An Autonomous Institute affiliated to Savitribai Phule Pune University)

Structure & Syllabus

B.Tech. Computer Science & Engineering (Artificial Intelligence)

With Effect from Academic Year 2026-27

Prepared by: - Board of Studies in CSE (Artificial Intelligence)

Approved by: - Academic Board, Vishwakarma Institute of Technology, Pune

Chairman – BOS **Chairman – Academic Board**

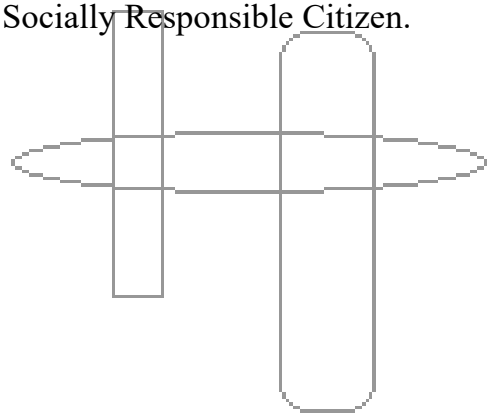
INDEX

Vision of the Institution

"To be globally acclaimed Institute in Technical Education and Research for holistic Socio-economic development".

Mission of the Institution

- To ensure that 100% students are employable and employed in Industry, Higher Studies, become Entrepreneurs, Civil / Defense Services / Govt. Jobs and other areas like Sports and Theatre.
- To strengthen Academic Practices in terms of Curriculum, Pedagogy, Assessment and Faculty Competence.
- Promote Research Culture among Students and Faculty through Projects and Consultancy.
- To make students Socially Responsible Citizen.

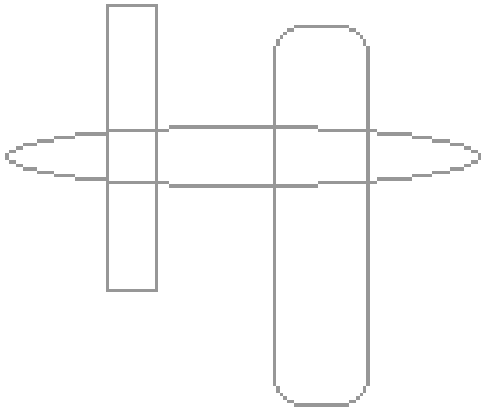


Vision of the Department

Excellence in Computer Science and Engineering, specialized in Artificial Intelligence, fostering skilled professionals and innovators for holistic societal Development

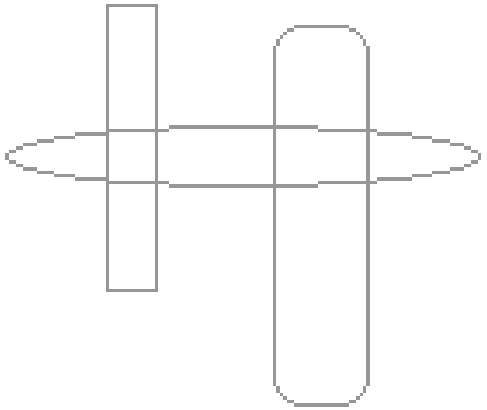
Mission of the Department

- To equip aspiring engineers for industry, academia and entrepreneurship by providing quality education in emerging Artificial Intelligence techniques.
- To impart value-added, research-oriented technical education by inculcating life skills, critical thinking and human values.
- To educate and empower learners with professional integrity through innovation, industry engagement and higher studies in Artificial Intelligence for sustainable solutions.
- To nurture ethical and socially responsible professionals by fostering lifelong learning skills for holistic societal development.



Program Educational Objectives (PEOs)

PEO	PEO Focus	PEO Statements- AI
PEO1	**Core competence**	Demonstrate core c**ompetence** in principles of computing and AI based technologies'.
PEO2	**Breadth**	Apply AI principles, methodologies, algorithms, and tools, to effectively **design**, develop, and implement AI-driven solutions.
PEO3	**Professionalism**	Excel as professionalism with the necessary **soft skills** to work collaboratively in interdisciplinary teams.
PEO4	**Learning Environment**	Aim for **continuing education** and **entrepreneurship** in emerging areas of computing and Artificial Intelligence.



Program Outcomes [PO]

PO	PO Statements
PO1	Engineering Knowledge: Apply knowledge of mathematics, natural science, computing, engineering fundamentals and an engineering specialization as specified inWK1 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& 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. (WK9)

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

PO10 Project Management and Finance: Apply knowledge and understanding of engineering management principles and economic decision-making and apply these to 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).

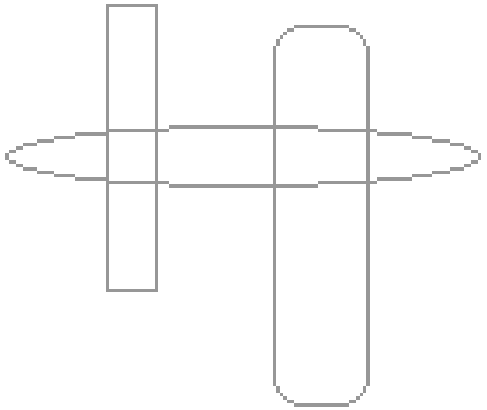
List of Program Outcomes [PO]

Program Specific Objectives (PSO)

PSO1: Demonstrate proficiency in essential concepts of computer science and programming solutions.

PSO2: Formulate robust software design, execution, and testing strategies employing a software paradigms and Artificial Intelligence knowledge to solve real word problems.

PSO3: Adapt and exhibit expertise in evolving areas of computer science, engineering and technology.



Course Name Nomenclature as per NEP (For FY and SY)

BSC: Basic Science Course

ESC: Engineering Science Course

PCC: Program Core Course

PEC: Program Elective Course

ELC: Experiential Learning Course

MD: Multi Disciplinary

MDOE: Multi-Disciplinary Open Elective

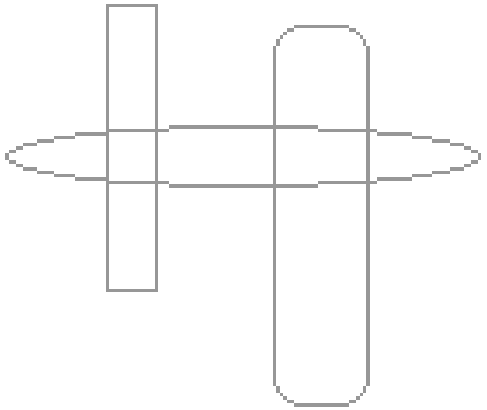
CC: Co-curricular Course

HSSM: Humanities Social Science and Management

IKS: Indian Knowledge System

FP: Field Project

INT: Internship



Nomenclature for Teaching and Examination Assessment Scheme AY 2026-27

Sr No.	Category	Head of Teaching/ Assessment	Abbreviation used
1	Teaching	Theory	Th
2	Teaching	Laboratory	Lab
3	Teaching	Tutorial	Tut
4	Teaching	Open Elective	OE
5	Teaching	Multi-Disciplinary	MD
6	Teaching	Computer Science and Engineering – Artificial Intelligence	CI
7	Assessment	Laboratory Continuous Assessment	CA
8	Assessment	Mid Semester Assessment	MSA
9	Assessment	End Semester Assessment	ESE
10	Assessment	Home Assignment	HA
11	Assessment	Course Project	CP
12	Assessment	Group Discussion	GD
13	Assessment	PowerPoint Presentation	PPT
14	Assessment	Class Test –1	CT1
15	Assessment	Class Test –2	CT2
16	Assessment	Mid Semester Examination	MSE
17	Assessment	End Semester Examination	ESE
18	Assessment	Written Examination	WRT
19	Assessment	Multiple Choice Questions	MCQ
20	Assessment	Laboratory	LAB

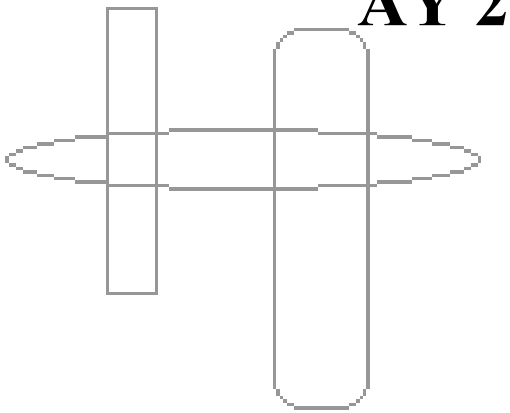
Title: Course Structure **FF No. 653**

Branch: Computer Science & Engineering (Artificial Intelligence) **Year:** S.Y. **A.Y.:** 2026-27 **Module:** III

Course Code	NEP Course Category	Course Name	Teaching Scheme (Hrs./week)			Credits															
			Theory	Tutorial	Practical	CVV	CP	LAB /TUT	GD/ PPT	HA	MSE(W)	MSE(R)	ESE(R)	ESE (W)	ESE (O)	PRACT+ CVV	ESE-Direct Evaluation	Theory	Tutorial	Practical	Total
CI2011	PCC	Fundamentals Of Data Structures	3		2		20							40		40		3		1	4
CI2012	PCC	Database Management Systems	2		2	20	30	10						40				2		1	3
CI2013	PCC	Object Oriented Programming	2		2	20	30	10						40				2		1	3
CI2014	PCC	Digital Systems and Microprocessor	2		2	20	30	10			40							2		1	3
MM0903	MDM	Discrete Mathematics	2	1		20	20			20	40							2		1	3
HS2002	HSSM-FCTC	From Campus To Corporate - 1	2									50	50					1			3
HS2001	AEC-RAD	Reasoning and Aptitude Development - 3	1														100		1		2
CI2015	VSEC-DT	Design Thinking - 1		1													100		1		1
CI2016	FP-EDI	Engineering Design And Innovation - 1										30	70								1
Total																					22

S. Y. B. Tech.
Computer Science & Engineering
(Artificial Intelligence)

AY 2026-27



FF No.: 654

CI2001: Fundamentals of Data Structures

Teaching Scheme	Examination Scheme
Credits :4	CP: Marks — 30
Lectures: 3 Hrs/Week	CVV: MARKS-20
Practical: 2 Hrs/Week	LAB: Marks — 10
Tutorial: — Hrs/Week	ESE (TH(W)): Marks — 40

Prerequisites:	Basic programming skills (C/ Python)

Course Objectives:	1. Develop a strong foundation in algorithms, abstract data types, and data structures, and understand their role in designing efficient software solutions.
	2. Analyze the efficiency of algorithms using time and space complexity, asymptotic notations, and algorithm design strategies such as Divide-and-Conquer and Greedy methods.
	3. Apply sequential data structures, including arrays and multidimensional arrays, to solve computational problems involving polynomial operations, sparse matrices, and efficient memory organization.
	4. Design and implement searching and sorting algorithms, and evaluate their correctness, efficiency, stability, and suitability for different application domains.
	5. Develop dynamic data structure solutions using linked lists, stacks, and queues, and apply them to solve problems involving memory management, expression processing, recursion, and backtracking.
	6. Select and implement appropriate data structures and algorithms for real-world applications by considering computational efficiency, memory utilization, scalability, and software design principles.

Course Outcomes:	*After completion of the course, student will be able to*
	1. Explain the concepts of algorithms, abstract data types, data structures, and algorithmic complexity, and apply suitable algorithm design strategies to solve computational problems. *(Understand – L2)*
	2. Implement sequential data structures such as arrays and multidimensional arrays for polynomial manipulation, sparse matrix representation, and efficient memory organization. *(Apply – L3)*
	3. Analyze and implement appropriate searching and sorting algorithms by comparing their computational complexity, stability, and performance for different applications. *(Analyze – L4)*
	4. Design and implement dynamic data structures using singly, doubly, circular, and generalized linked lists to perform fundamental data manipulation operations. *(Apply/Create – L3/L6)*
	5. Develop applications using stack and queue data structures for expression evaluation, recursion, backtracking, and efficient data processing. *(Apply – L3)*
	6. Select, design, and implement appropriate data structures and algorithms to solve real-world computational problems by considering efficiency, scalability, and optimal resource utilization. *(Create – L6)*

Course Relevance:

This is a basic course for Computer Engineering and allied branches. It has high relevance across all domains of computer engineering — industry, research, and beyond — as a foundational prerequisite. Data structures are central to computer algorithms, enabling efficient data management; a well-chosen data structure improves the performance of a program or algorithm significantly.

Section 1:	**Topics/Contents**
Section 1:	**Unit I – Introduction to Algorithms and Data Structures (7 Hours)** From Problem to Program (Problem, Solution, Algorithm, Data Structure and Program). Data, Information, Knowledge and Data Structure; Abstract Data Types (ADT); classification (Linear/Non-linear, Static/Dynamic, Persistent/Ephemeral). Algorithms: problem solving, characteristics, design tools (pseudo-code, flowchart). Complexity: space and time complexity, asymptotic notation (Big-O, Theta, Omega), step-count method, analysis of Linear, Quadratic, Cubic and Logarithmic constructs. Algorithmic strategies: Divide and Conquer, Greedy.

	Unit II – Linear Data Structures Using Sequential Organization (7 Hours) Concept of sequential organization; arrays as an ADT and array operations; merging of two arrays; storage representation and address calculation (row-major, column-major); multi-dimensional arrays. Ordered lists; single-variable polynomial representation using arrays, polynomial as array of structures, polynomial addition and multiplication. Sparse matrices: representation, addition, simple and fast transpose, time–space trade-off.
	Unit III – Searching and Sorting (7 Hours) Searching: sequential/linear search, sentinel search, binary search, Fibonacci search, indexed sequential search. Sorting: internal vs external sorting; sort order, stability, efficiency and number of passes; comparison-based methods (Bubble, Insertion, Selection, Quick, Shell); non-comparison-based methods (Radix, Counting, Bucket); comparison of all methods and their complexities.
Section 2:	**Topics/Contents**
Section 2:	**Unit IV – Linked List (7 Hours)** Static vs dynamic memory allocation; introduction to linked lists, realization using dynamic memory management, linked list as ADT. Types: singly linked (linear and circular), doubly linked, doubly circular linked lists. Primitive operations: create, traverse, search, insert, delete, sort, concatenate. Polynomial manipulation using linked lists; Generalized Linked List (GLL) concept and polynomial representation using GLL.
	Unit V – Stack (7 Hours) Basic concept, stack ADT, representation using sequential organization, stack operations, multiple stacks. Applications: expression evaluation and conversion, Polish notation, need for prefix/postfix expressions, postfix expression evaluation, linked stack and operations. Recursion: concept and variants (direct, indirect, tail, tree); backtracking strategy and use of stack in backtracking.
	Unit VI – Queue (7 Hours) Basic concept, queue as ADT, representation using sequential organization, queue operations, circular queue and its advantages, multi-queues, linked queue and operations. Deque: basic concept, input-restricted and output-restricted types. Priority queue: basic concept, ascending and descending types.

Text Books: (As per IEEE format)	
1	E. Horowitz, S. Sahni, and S. Anderson-Freed, *Fundamentals of Data Structures in C*, 2nd ed. Hyderabad, India: Universities Press, 2008.
2	A. M. Tenenbaum, Y. Langsam, and M. J. Augenstein, *Data Structures Using C*, New Delhi, India: Pearson Education, 1990.
3	R. Kruse, C. L. Tondo, and B. Leung, *Data Structures and Program Design in C*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall, 1997.

Reference Books: (As per IEEE format)	
1	M. A. Weiss, *Data Structures and Algorithm Analysis in C*, 2nd ed. Boston, MA, USA: Addison-Wesley, 1997.
2	R. Sedgewick, *Algorithms in C*, Parts 1–5, 3rd ed. Boston, MA, USA: Addison-Wesley, 1998.
3	S. Lipschutz, *Schaum's Outline of Data Structures with C*. New York, NY, USA: McGraw-Hill Education, 2014.

List of Practical's:

Practical 1: Algorithm Design and Complexity Analysis

Design algorithms using pseudocode and flowcharts for real-world problems. Analyze their time and space complexities using Big-O, Theta, and Omega notations.

Practical 2: Array Operations

Implement one-dimensional array operations including traversal, insertion, deletion, searching, updating, and merging. Compare the execution time of different operations.

Practical 3: Multidimensional Arrays and Sparse Matrix

Implement two-dimensional arrays with row-major and column-major address calculations. Represent sparse matrices and perform simple and fast transpose operations.

Practical 4: Polynomial Representation Using Arrays

Represent single-variable polynomials using arrays or structures. Implement polynomial addition and multiplication using array representation.

Practical 5: Searching Algorithms

Implement Linear Search, Sentinel Search, Binary Search, Fibonacci Search, and Indexed Sequential Search. Compare their performance for different input sizes.

Practical 6: Comparison-Based Sorting

Implement Bubble, Selection, Insertion, Shell, and Quick Sort algorithms. Analyze their best, average, and worst-case time complexities.

Practical 7: Non-Comparison-Based Sorting

Implement Counting Sort, Radix Sort, and Bucket Sort. Compare their efficiency with comparison-based sorting algorithms.

Practical 8: Singly Linked List

Implement a Singly Linked List using dynamic memory allocation. Perform insertion, deletion, traversal, searching, sorting, and concatenation operations.

Practical 9: Doubly and Circular Linked Lists

Implement Doubly Linked List, Circular Linked List, and Doubly Circular Linked List. Perform all primitive operations and compare their advantages.

Practical 10: Polynomial Manipulation Using Linked Lists

Represent polynomials using linked lists or generalized linked lists. Implement polynomial addition and compare it with array-based implementation.

Practical 11: Stack and Expression Evaluation

Implement Stack using arrays and linked lists. Perform infix-to-postfix/prefix conversion, postfix evaluation, and balanced parentheses checking.

Practical 12: Recursion and Backtracking

Implement recursive algorithms demonstrating direct, indirect, tail, and tree recursion. Solve a backtracking problem such as N-Queens or Rat in a Maze.

Practical 13: Queue Implementation

Implement Queue, Circular Queue, and Linked Queue using sequential and linked representations. Demonstrate enqueue, dequeue, and other queue operations.

Practical 14: Deque and Priority Queue

Implement Input-Restricted Deque, Output-Restricted Deque, and Ascending/Descending Priority Queue. Compare their applications and performance for different scenarios.

List of Project areas:

- Smart Student Record Management System
- AI-Based Library Recommendation System (Basic Data Structure Implementation)
- Hospital Queue Management System
- Intelligent Parking Management System
- Smart Inventory Management System

- Contact Book with Fast Search and Sorting
- Online Examination Management System
- Campus Placement Management System
- Smart Banking Queue Simulator
- Mini Search Engine Using Indexing and Searching Algorithms

CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '—' = no correlation.

CO	Program Outcome (PO)											PSO		
CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2	PSO 3
CO1	3	2	1	–	–	–	–	–	–	–	–	3	2	1
CO2	3	3	2	1	2	–	–	–	–	–	–	3	3	2
CO3	3	3	2	2	2	–	–	–	–	–	–	3	3	2
CO4	3	3	3	2	2	–	–	–	–	–	–	3	3	3
CO5	3	3	2	2	3	–	–	–	–	–	–	3	3	2
CO6	3	3	3	2	3	–	–	–	1	1	1	3	3	3
Avg	3	2	1	–	–	–	–	–	–	–	–	3	2	1

sCO – WK Mapping (Knowledge & Attitude Profiles)

'✓' indicates the knowledge profile drawn upon by the CO.

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1	–	✓	✓	✓	–	–	–	–	–
CO2	–	✓	✓	✓	✓	✓	–	–	–
CO3	–	✓	✓	✓	✓	✓	–	–	–
CO4	–	✓	✓	✓	✓	✓	–	–	–
CO5	–	✓	✓	✓	✓	✓	–	✓	–
CO6	–	✓	✓	✓	✓	✓	✓	✓	–

CO – WP Mapping (Complex Engineering Problems)

'✓' indicates the problem-solving attribute addressed by the CO.

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO1	✓	–	✓	✓	–	–	–
CO2	✓	✓	✓	✓	–	–	✓
CO3	✓	✓	✓	✓	–	–	✓
CO4	✓	✓	✓	✓	–	✓	✓
CO5	✓	✓	✓	✓	–	✓	✓
CO6	✓	✓	✓	✓	✓	✓	✓

CO – EA Mapping (Complex Engineering Activities)

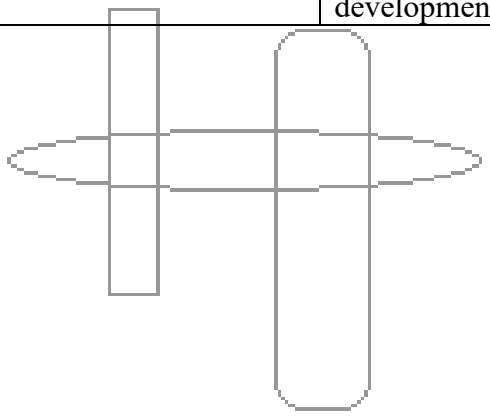
'✓' indicates the activity attribute addressed by the CO.

CO	EA1	EA2	EA3	EA4	EA5
CO1	✓	✓	–	–	–
CO2	✓	✓	✓	–	✓
CO3	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓
CO5	✓	✓	✓	✓	✓
CO6	✓	✓	✓	✓	✓

CO – SDG Mapping (Sustainable Development Goals)

CO	SDG(s) Addressed	Justification / Linkage
CO1	**SDG 4 – Quality Education**	Develops a strong foundation in algorithms and data structures, enabling students to acquire essential computational thinking and problem-solving skills required for lifelong learning.
CO2	**SDG 4 – Quality Education**, **SDG 9 – Industry, Innovation and Infrastructure**	Enables students to design efficient data organization techniques using arrays, sparse matrices, and polynomial representations,

		forming the basis for innovative software and digital infrastructure.
CO3	**SDG 8 – Decent Work and Economic Growth**, **SDG 9 – Industry, Innovation and Infrastructure**	Enhances the ability to select efficient searching and sorting algorithms, improving software performance, productivity, and computational efficiency in industrial applications.
CO4	**SDG 9 – Industry, Innovation and Infrastructure**, **SDG 12 – Responsible Consumption and Production**	Promotes the development of optimized dynamic data structures that improve memory utilization and computational resource efficiency in software systems.
CO5	**SDG 4 – Quality Education**, **SDG 9 – Industry, Innovation and Infrastructure**	Develops programming competence through stacks, queues, recursion, and backtracking, supporting the design of reliable and scalable software solutions.
CO6	**SDG 8 – Decent Work and Economic Growth**, **SDG 9 – Industry, Innovation and Infrastructure**, **SDG 17 – Partnerships for the Goals**	Equips students to develop efficient algorithmic solutions for real-world engineering problems, fostering innovation, employability, and collaborative software development practices.



FF No. : 654

CI2012: Database Management System

Teaching Scheme	Examination Scheme
Credits : 3	CP: 30 Marks
Lectures : 2 Hrs/week	Lab: 10 Marks
Practical : 2 Hrs/week	CVV: 30 Marks
Tutorial : NA Hrs/week	ESE TH(W):40 Marks

Prerequisites:	
●	Data structures, Discrete Mathematics
Course Objectives:	
●	To understand DBMS basics, data models, and user roles
●	To learn ER modeling and relational algebra for schema design.
●	To acquire skills in SQL and PL/SQL for database operations
●	To understand normalization techniques to normalize the database.
●	To become familiar with transactions, concurrency, and query optimization.
●	To gain knowledge of NoSQL, MongoDB, and modern data storage trends.
Course Outcomes:	
	After completion of the course, student will be able to
1.	Describe the fundamentals of DBMS, various data models, architectures, and the responsibilities of users and administrators.
2.	Make use of ER models to design and relational algebra to query database systems.
3.	Implement database operations using SQL and PL/SQL constructs such as procedures, functions, and triggers.
4.	Apply normalization techniques to eliminate redundancy and maintain data integrity.
5.	Demonstrate transaction properties, concurrency control, and query optimization methods for efficient database performance.
6.	Use knowledge of NoSQL databases, MongoDB operations, and distributed data systems to address modern storage requirements.

Section1:	Topics/Contents

Unit 1: Introduction to DBMS and Data Models (03 Hours)

Introduction, Database Users and Administrators, Need of Database Management Systems, Characteristics of the Database, DBMS Architecture, Data Independence, View of Data, Database System Applications, Data Models: Relational Model, Hierarchical Model, Network Model, Objectbased Model.

Unit 2: ER Modelling and Relational Model (04 Hours)

Entity-Relationship Model: Entities, Attributes, and Relationships; Cardinality of Relationships; Strong and Weak Entity Sets; Generalization, Specialization, and Aggregation; Translating your ER model into a relational model with integrity constraints over relations. Relational Model: Concept of relations, referential integrity, and foreign keys, Relational algebra operators: selection, projection, cross product, various types of joins, division, example queries, tuple relational calculus, domain relational calculus, and converting the database specification in E/R notation to the relational schema.

Unit 3:SQL and PLSQL (05 Hours)

SQL Basic, DDL, DML, DCL, DTL, Joins and Set Operations, SQL Constraints: Primary Key, Foreign Key, Unique, Not Null, Check, Default, Group by, Order by, Having Clauses, Subqueries, View, and Index. PL/SQL: Concept of Stored Procedures and Functions, Cursors, Triggers, Assertions, Roles, and Privileges.

Section2:	Topics/Contents

Unit 4: Dependencies and Normal forms (04 Hours)

Importance of a good schema design, problems encountered with bad schema designs, CODD's Rules, the Concept of Functional Dependencies and Closure, Minimal Cover of Functional Dependencies, Anomalies in unnormalized and partially normalized relations, Armstrong's axioms for FD's, Decomposition: Lossless-Join Decomposition, Dependency Preservation, Multi-Valued Dependencies. Normalization: 1NF, 2NF, 3NF, BCNF, 4NF.

Unit 5: Transaction Management and Query Optimization (04 Hours)

Transaction Concept and ACID Properties, States of Transactions, Schedules, Serializability: Conflict and View Serializability, Concurrency Control Techniques: Lock-Based Protocols, Timestamp-Based Protocols, Deadlock Handling and Prevention, Recovery Techniques: Log-Based Recovery, Shadow Paging, Indexing: Primary, Secondary, and Multilevel Indexes, Query processing and optimization: evaluation of relational algebra expressions, query equivalence, join strategies, and query optimization algorithms**.**

Unit 6: NoSQL Databases **(04 Hours)**

Types of Data: Structured, Unstructured, and Semi-Structured Data CAP Theorem, NoSQL Database Introduction, Need, Features, Types of NoSQL Databases: Key-Value, Column, Document, and Graph Databases, BASE Properties, ACID vs. BASE, and a Comparative Study of RDBMS and NoSQL. MongoDB (with syntax and usage): CRUD Operations, Indexing, Aggregation, MapReduce, Cloud Databases, Distributed Databases.

Text Books: ***(As per IEEE format)***

1	A. Silberschatz, H. F. Korth, and S. Sudarshan, *Database System Concepts*, 6th ed. New York, NY, USA: McGraw-Hill Education, 2011.
2	R. Elmasri and S. B. Navathe, *Fundamentals of Database Systems*, 7th ed. Boston, MA, USA: Pearson, 2016.

Reference Books: ***(As per IEEE format)***

1	T. M. Connolly and C. E. Begg, *Database Systems: A Practical Approach to Design, Implementation, and Management*, 6th ed. Harlow, U.K.: Pearson.
2	R. Ramakrishnan and J. Gehrke, *Database Management Systems*, 3rd ed. New York, NY, USA: McGraw-Hill Education
3	K. Chodorow, *MongoDB: The Definitive Guide*, 2nd ed. O'Reilly Publications, ISBN: 978-93-5110-269-4.
4	P. S. Deshpande, *SQL and PL/SQL for Oracle 10g Black Book*. DreamTech. .
5	I. Bayross, *SQL, PL/SQL: The Programming Language of Oracle*. New Delhi, India: BPB Publications
6	G. Reese, R. Yarger, T. King, and H. Williams, *Managing and Using MySQL*, 2nd ed. New Delhi, India: Shroff Publishers and Distributors Pvt. Ltd
7	P. Dalton, *SQL Server Black Book*. New Delhi, India: DreamTech Press.
8	E. Redmond and J. R. Wilson, *Seven Databases in Seven Weeks: A Guide to Modern Databases and the NoSQL Movement*. Dallas, TX, USA: Pragmatic Bookshelf
9.	J. Kreibich, *Using SQLite*, 1st ed. Sebastopol, CA, USA: O'Reilly Media.

List of Practical's:

1. Implementation of DDL: Design and Develop SQL DDL statements which demonstrate the use of SQL objects such as Table, Index,

2. Implementation of DML, and DQL commands: Create a table Employees, Insert records, Update a salary column, and retrieve records using SELECT.

3. SQL Joins and Set Operations: Join the Employees and Departments tables to show employee names with their department names.

4. Views and Nested Queries: Create a view called HighSalaryEmployees and write a nested query to find employees earning more than the average salary.

5. Functions and Stored Procedures: Write a function to calculate annual salary. Write a stored procedure to update salary based on designation.

6. Triggers Implementation: Create a trigger that logs insert operations into a separate Logs table.

7. ER to Relational Mapping: Convert an Entity-Relationship (ER) diagram to relational schema.

8. Cursors: (All types: Implicit, Explicit, Cursor FOR Loop, Parameterized Cursor).

.9. NoSQL Implementation using MongoDB: Create a MongoDB collection called Products and perform insert and query operations

10.MongoDB - Aggregation and Indexing: Design and Develop MongoDB Queries using aggregation and indexing with suitable example using MongoDB.

11 MongoDB - Map reduces operations: Implement Map reduces operation with suitable example using MongoDB.

12 Design a two-tier application using Java, Python, or PHP as the front-end, with MySQL or MongoDB as the back-end for data storage and retrieval.

List of Project areas:

1.Online Course Management System: Design a system to manage online courses, student registrations, assignments, grades, and faculty dashboards. Includes user roles, course prerequisites, and submission tracking.

2. Hospital Appointment & Patient Management System: Build a DBMS to manage patient records, doctor schedules, appointments, treatments, and billing. Involves relationships among departments, doctors, patients, and reports.

3. Library Management System with Fine Calculation: Manage book check-ins/check-outs, due dates, reservations, fines, and digital catalog search. Add features like genre-based search and overdue fine tracking.

4. E-commerce Order Processing System: Handle customers, products, orders, payments, and shipping. Include stock availability, order history, and product reviews.

5. Hotel Booking and Reservation System: Design a system for guests to search, reserve, and cancel rooms, with staff and service records. Include room types, seasonal pricing, and transaction logging.

6. Student Attendance & Result Management: Track attendance and academic results of students across subjects and semesters. Enable performance analytics per student and per subject.

7. Banking and Loan Management System: Develop a database to manage user accounts, transactions, loan applications, EMI schedules, and credit scores. Requires ACID compliance, trigger-based fraud alerts, and audit trails.

8. Multi-Branch Inventory & Sales System for a Retail Chain: Manage inventory, stock transfers, branch sales, suppliers, and demand forecasting. Involves complex joins, transactions, triggers, and partitioned tables.

9. University ERP (Enterprise Resource Planning) System: Create a unified system for managing departments, students, faculty, courses, fees, payroll, and exams. Requires large-scale schema planning and role-based data access.

10. Healthcare Analytics Dashboard: Design a DBMS to collect and analyze patient data for disease prediction and hospital analytics. Incorporates views, stored procedures, triggers, and indexed performance optimization.

11. Online Job Portal with Resume Matching: Store job postings, resumes, recruiter profiles, and use query logic to match candidates with job roles. Integrates full-text search, ranking algorithms, and resume parsing.

12. Social Media Platform DBMS: Simulate a social network backend—users, posts, likes, shares, friends, and messaging. Needs recursive queries (friends of friends), normalization, and analytics.

CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '-' = no correlation.

CO	Program Outcome (PO)											PSO		
CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2	PSO 3
CO1	2	1									2	2		
CO2	3	2	2		2			2		2	2	3	2	
CO3	3		2		3			2	2	2	2	3	2	2
CO4	3	2	2	2				2		2	2	3	2	
CO5	2	1									2	2		
CO6	2	2	2		3			2	2	2	2	3	2	2
Avg	2.67	1.67	2.00	2.00	2.67			2.00	2.00	2.00	2.00	2.67	2.00	

CO – WK Mapping (Knowledge & Attitude Profiles) Sample Table

'✓' indicates the knowledge profile drawn upon by the CO.

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1	✓	✓							
CO2	✓	✓	✓						
CO3		✓	✓	✓					
CO4	✓	✓	✓	✓					
CO5	✓	✓	✓	✓	✓				
CO6		✓	✓	✓	✓	✓			

CO – WP Mapping (Complex Engineering Problems) Sample Table

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO1	✓	✓					
CO2	✓	✓	✓				
CO3	✓		✓	✓			
CO4	✓	✓	✓	✓			
CO5	✓	✓	✓	✓	✓		
CO6	✓		✓	✓	✓	✓	✓

CO – EA Mapping (Complex Engineering Activities)

'✓' indicates the activity attribute addressed by the CO

CO	EA1	EA2	EA3	EA4	EA5
CO1	✓	✓			
CO2	✓	✓	✓		
CO3	✓		✓	✓	
CO4	✓	✓	✓	✓	
CO5	✓	✓	✓	✓	✓
CO6	✓		✓	✓	✓

CO – SDG Mapping (Sustainable Development Goals) Sample Table

CO	SDG(s) Addressed	Justification / Linkage
CO1	4, 9	Develops fundamental database knowledge and technical skills (SDG 4) while supporting digital infrastructure and technological innovation (SDG 9).
CO2	9	Database design and relational algebra promote innovation in software systems and digital infrastructure (SDG 9).
CO3	9	SQL and PL/SQL implementation enables reliable enterprise database applications and industrial digital transformation (SDG 9).
CO4	9, 16	Normalization ensures data quality, integrity, and trustworthy information systems, supporting SDGs 9 and 16.
CO5	9	Transaction management, concurrency control, and query optimization improve efficient and scalable database systems (SDG 9).
CO6	9, 11	NoSQL, MongoDB, and distributed databases support cloud computing, smart applications, and sustainable smart-city infrastructure (SDGs 9 and 11).

CI2013: Object Oriented Programming

Teaching Scheme	**Examination Scheme**
Credits :3	CP: Marks:30
Lectures : 2 Hrs/week	LAB Marks:10
Practical : 2 Hrs/week	ESE (TH(W)): Marks:60
Tutorial : Hrs/week	

Prerequisites:	
●	C programming
●	Problem solving skills using programming language
Course Objectives:	
●	Differentiate various programming paradigms.
●	Understand the concept of data abstraction and encapsulation, how to design C++ classes for code reuse.
●	Learn to implement class member functions, constructors and operators overloading in C++.
●	Learn Inheritance, virtual functions and dynamic binding with polymorphism.
●	Utilize file handling techniques in C++ and Java for sequential and random access, while incorporating robust, reliable and efficient data management.
●	To develop problem solving ability using Object Oriented programming constructs like multithreading
Course Outcomes:	
	After completion of the course, student will be able to
1	Understand object-oriented programming features.
2	Identify classes, objects, methods, and handle object creation, initialization, and destruction to model real-world problems.
3	Understand and implement various types of constructors and destructors.
4	Identify relationships among objects using inheritance and polymorphism principles.
5	Use files for persistent data storage to model real world applications.
6	Handle different types of exceptions, apply generic programming concepts

Section1 :	Topics/Contents

Unit 1: Fundamentals of Object Oriented Programming (4 Hours)

Need of object-oriented programming, Fundamentals of Object-Oriented Programming: Objects, Classes, Data Members, Methods, Namespaces, Data Encapsulation, Data Abstraction and Information Hiding, Inheritance, Polymorphism, Static and Dynamic Binding, Message Passing. Benefits of OOP, C++ as object-oriented programming Language. C++ Programming Basics, Data Types, Structure, Enumeration, Control structures, Arrays, Strings, Access specifier.

Unit 2: Function and Pointer (5 Hours)

Introducing Functions, Constructor, Destructor, Objects and Memory requirements, Static members: variables and functions, inline function, friend function. Characteristics of Constructors, Types of Constructors, Constructor Overloading, Dynamic Initialization of an Object, Constructor with Default Arguments.

Pointers: declaring and initializing pointers, indirection Operators, Memory management: new and delete, Pointers to Objects, this pointer, Arrays of pointers, Function Pointers, Pointers to pointers, Pointer to Derived classes, Passing pointers to functions.

Unit 3: Inheritance and Polymorphism (5 Hours)

Base class and derived Class, Protected members, Overriding member function, Class Hierarchies, Public and Private Inheritance, Types of Inheritance: Single, Multilevel, Hierarchical, Multiple, Hybrid, Ambiguity in multiple Inheritance, Virtual Base Class, Abstract Class, Friend Class. Introduction to polymorphism, Types of Polymorphism, Operator Overloading - concept of overloading, Overloading Unary Operators, Overloading Binary Operators, Function overloading, Runtime Polymorphism - Pointer to Base Class, Virtual function and its significance, pure virtual function, Virtual destructor

Section2 :	Topics/Contents

Unit 4: Exception Handling and Templates (5 Hours)

Exception Handling - Fundamentals, other error handling techniques, simple exception handling-Divide by zero, Multiple catching, rethrowing an exception, exception specifications, user defined exceptions, processing unexpected exceptions, constructor, destructor and exception handling, exception and inheritance. Templates- The power of templates, function template, overloading Function templates and class template and Nontype parameters, template and friend generic functions, The type name and export keyword

Unit 5: Files and Streams (5 Hours)

Data hierarchy, Stream and files, Stream Classes, Stream Errors, Disk File I/O with Streams, File Pointers, and Error Handling in File I/O, File I/O with Member Functions, Overloading the Extraction and Insertion Operators, memory as a Stream Object, Command-Line Arguments, Printer output File Processing, Primitive Data Processing, Object Data Processing,

Unit 6: Object Oriented Programming with Java (4 Hours)

Introduction to Java and its OOP Features, Java vs C++, Class and Object in Java, Access modifiers: public, private, protected, default, Single and Multilevel inheritance using extends keyword, Method overriding using super keyword, Exception Handling in Java: try, catch, throw, finally.

Text Books: ***(As per IEEE format)***

1	*The C++ Programming Language, Bjarne Stroustrup, 4th Edition, Addison-Wesley Pearson Education .*
2	*E. Balaguruswamy, "Object Oriented Programming Using C++ ", Tata McGraw Hill.*
3	*An Introduction to Object Oriented Programming, by Timothy A. Budd, published by Addison-Wesley.*

Reference Books: ***(As per IEEE format)***

1	*Herbert Schildt , "Java: The Complete Reference", 4th Edition*
2	*Herbert Schildt , "C++: The Complete Reference", 4th Edition.*

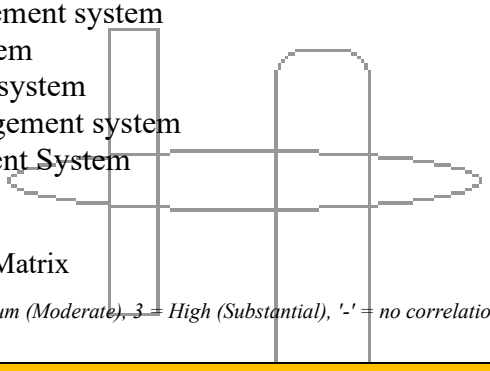
List of Practical's:

1. Write a C++ program to implement the concept of objects, classes, constructors, destructors.
2. Write a program to implement a class for Book details as follows Private: bookid, book_name, author, price Public: get_details(), print_details() Get details of some books from user and print in tabular form. Also find the total prize of all the books.
3. Write a program to extend Book class with dynamic memory allocation. (Use of "new" keyword and object pointers)
4. Design a class 'Complex' with data members for real and imaginary part. Provide default and Parameterized constructors. Write a program to perform arithmetic operations of two complex numbers.
5. Design and develop inheritance for a given case study, identify objects and relationships and implement inheritance wherever applicable. Employee class hasEmp_name, Emp_id, Address, Mail_id, and Mobile_no as members. Inherit the classes: Programmer, Team Lead, Assistant Project Manager and Project Manager from employee class. Add Basic Pay (BP) as the member of all the inherited classes with 52% of BP as DA, 27 % of BP as HRA, 12% of BP as PF, 0.1% of BP for staff club fund. Generate pay slips for the employees with their gross and net salary.
6. Design a base class shape with two double type values and member functions to input the data and compute_area() for calculating area of shape. Derive two classes: triangle and rectangle. Make compute_area() as an abstract function and redefine this function in the derived class to suit their requirements. Write a program that accepts dimensions of triangle/rectangle and displays calculated area. Implement dynamic binding for given case study.
7. Design and develop a context for given case study and implement an interface for Vehicles Consider the example of vehicles like bicycle, car and bike. All Vehicles have common functionalities such as Gear Change, Speed up and apply brakes. Make an interface and put all these common functionalities. Bicycle, Bike, Car classes should be implemented for all these functionalities in their own class in their own way.
8. Implement a program for maintaining a database of student records using Files. Students have Student_id,name, Roll_no, Class, marks and address. Display the data for a few students. 1. Create Database 2. Display Database 3. Delete Records 4. Update Record 5. Search Record
9. Implement a program to handle arithmetic exceptions and array index out of bounds.
10. Implement a generic program using any collection class to count the number of elements in a collection that have a specific property such as even numbers, odd number, prime number and palindromes.
11. Write a Java program to create a student class with details like Roll no, Name, Marks of five subjects, percentage, class (First, Second, etc) have following functions: parameterized constructor, destructor, Display and Calculate percentage and grade.

12. Identify commonalities and differences between Publication, Book and Magazine classes. Title, Price, Copies are common instance variables and saleCopy is a common method. The differences are, Bookclass has author and orderCopies(). Magazine Class has methods orderQty, Current issue, receiveissue(). Write a Java program to find how many copies of the given books are ordered and display total sale of publication.
13. Create a Java program to implement a Student Grading System that accepts student details such as Name, Roll No, Marks of 5 subjects, Calculates total, percentage, and grade, displays results in a tabular format. Use inheritance to separate basic student info from academic performance.
14. Demonstration of Object-Oriented Programming using Java and database through JDBC

sList of Project areas:

1. Electricity billing system
2. e-Healthcare management system
3. Library management system
4. Online bank management system
5. Online medical management system
6. Online quiz management system
7. Online Survey System
8. Stock management system
9. Supply chain management system
10. Hospital Management System



CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '-' = no correlation.

CO	Program Outcome (PO)											PSO		
CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2	PSO 3
CO1	3	2	2									3		
CO2	3	3	3	1	1	2	2	2			2	3		3
CO3	3	2	3	1	1		2					3		3
CO4	3	3	3	1							2	3	3	3
CO5	3	3	3	1	1						2	3		3
CO6	3	2	3									3		3
Avg	3	2.5	2.83	1	1	2	2	2			2	3	3	3

CO – WK Mapping (Knowledge & Attitude Profiles) Sample Table

'✓' indicates the knowledge profile drawn upon by the CO.

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1	✓	✓	✓						
CO2	✓	✓	✓	✓					
CO3	✓	✓	✓	✓	✓				
CO4	✓	✓	✓	✓	✓	✓			
CO5	✓	✓	✓	✓	✓	✓	✓		
CO6	✓	✓	✓	✓	✓	✓	✓	✓	

CO – WP Mapping (Complex Engineering Problems) Sample Table

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO1	✓						
CO2	✓	✓					
CO3	✓	✓	✓				
CO4	✓	✓	✓	✓			
CO5	✓	✓	✓	✓	✓		
CO6	✓	✓	✓	✓	✓	✓	✓

CO – EA Mapping (Complex Engineering Activities)

'✓' indicates the activity attribute addressed by the CO

CO	EA1	EA2	EA3	EA4	EA5
CO1	✓				
CO2	✓	✓			
CO3	✓	✓	✓		
CO4	✓	✓	✓		
CO5	✓	✓	✓	✓	
CO6	✓	✓	✓	✓	✓

CO – SDG Mapping (Sustainable Development Goals) Sample Table

CO	SDG(s) Addressed	Justification / Linkage
CO1	4, 9	Develops strong object-oriented programming fundamentals (SDG 4) that form the basis for designing innovative software systems and digital infrastructure (SDG 9). .
CO2	4,9	Enables students to model real-world problems using classes and objects, promoting software design skills and innovation in application development (SDG 4, SDG 9).
CO3	8, 9	Enhances software development practices through constructors, destructors, and memory management, improving employability and supporting industrial software development (SDG 8, SDG 9).
CO4	9	Encourages the development of reusable, maintainable, and scalable software using inheritance and polymorphism, contributing to sustainable industrial innovation (SDG 9)
CO5	9, 16	Develops reliable data storage and file management techniques that support secure information systems and robust digital infrastructure (SDG 9, SDG 16).
CO6	4, 9, 16	Applies exception handling, templates, and generic programming to build secure, reliable, and maintainable software while strengthening advanced programming competencies (SDG 4, SDG 9, SDG 16).

FF No.: 654-A

MM0903 : Discrete Mathematics

Teaching Scheme	**Examination Scheme**
Credits : 3	CP: 30 Marks
Lectures : 2 Hrs/week	GD/PPT: 20 Marks
Practical : NA	HA : 10 Marks
Tutorial : 1 Hrs/week	ESE ((TH)W): 40 Marks

Prerequisites:	
●	Basic knowledge of mathematics and logic
Course Objectives:	
CO1	To provide strong basic discrete structures in mathematical reasoning and logic essential for computer science problem-solving and theoretical computation
CO2	To develop recurrence relations for a wide variety of combinatorial problems and counting
CO3	To develop a comprehensive understanding of fundamental algebraic structures including groups, rings, and fields
CO4	To understand and apply solid foundation in the principles and techniques of number theory
CO5	To familiarize students fundamental concepts and structures in graph theory and develop analytical skills for modeling and solving real-world problems using graphs
CO6	To understand the theory and application of tree structures in graph theory
Course Outcomes:	
	After completion of the course, student will be able to
1.	Reason mathematically about elementary discrete structures (such as functions, relations, sets, graphs, and trees) used in computer algorithms and systems
2.	Express mathematical properties via the formal language of propositional and predicate logic
3.	Develop recurrence relations for a wide variety of combinatorial problems
4.	Demonstrate use of advanced combinatorial techniques
5.	Describe elementary concepts in modular arithmetic and their applications
6.	Exhibit understanding of basic graph theory, tree and its applications

Section1:	Topics/Contents	
Unit 1:	**Set Theory And Logic**	**(4 Hours)**
Sets: Sets and Subsets, Power Set, Cartesian Product, Set Operations, Venn Diagram, Inclusion-Exclusion Principle, Computer Representation of Sets. **Relations:** Product Sets, Pictorial Representatives of Relations, Composition of Relations, Types of Relations, Closure Properties, Equivalence Relations, Partial Orderings, partitions, Hasse Diagram, Lattices. **Logic:** Propositions, Truth Tables, Tautologies, Conditional Propositions and Logical Equivalence, Arguments and Rules of Inference. **Functions:** Domain, range, One-to-One, Onto, and Invertible Functions, Exponential and Logarithmic Functions, Cardinality, Algorithms and Functions		
Unit 2:	**Recursion and Counting Methods**	**(4 Hours)**
Mathematical Induction, Recursion, Recurrence Relation, Solving Recurrence Relations, The Basics of Counting, rule of Sum and Product, Permutations and Combinations, Binomial Coefficients and Identities, Algorithms for generating Permutations and Combinations, The Principle of Inclusion-Exclusion, The Addition and Multiplication Rules, The Pigeon-Hole Principle		
Unit 3:	**Algebraic Structures**	**(5 Hours)**
The structure of algebra, Algebraic Systems, Semi-Groups, Monoids, Groups, Subgroups And Their Properties -Cyclic Groups - Cosets - Permutation Groups - Lagrange's Theorem - Cayley's Theorem -Normal Subgroups - Homomorphism Of Groups - Quotient Groups –Introduction To Rings And Fields.		
Section2:	**Topics/Contents**	
Unit 4:	**Number Theory**	**(5 Hours)**
Number theory overview, Divisors - Properties of Divisibility, Division Algorithm, Greatest Common Divisor GCD and its Properties, Remainder Classes, Properties of Congruence, Solving Congruences, Euclidean Algorithm, Extended Euclidean Algorithm, Prime Factorization Theorem, Modular Arithmetic, Euler Phi Function, Euler's Theorem, Fermat's Little Theorem, Additive and Multiplicative Inverses, Chinese Remainder Theorem		
Unit 5:	**Graph Theory**	**(5 Hours)**
Graphs and Simple Graphs, Paths and Cycles, The Incidence and Adjacency Matrices, Vertex Degrees, Paths and Connection, Cycles, Hamiltonian Cycles, Subgraphs - Graph Isomorphism, Directed and Undirected graphs, Vertex types - Isolated and pendent vertices, In degree and out degree, The handshaking theorem, Types of graphs – Null graph, regular graph, complete graph, Operations of graphs – Union, Intersection, Sum of two graphs, Product of graphs, Composition, complement, Vertex And Edge Cuts - Vertex And Edge Connectivity		
Unit 6:	**Trees**	**(5 Hours)**
Trees - Acyclic graph, Terminology and Characterizations of Trees, Forest, Spanning tree, Minimal Spanning, Branch of tree, Chord, Rooted tree, Co tree, Binary tree - Path length of a binary tree, Binary tree representation of general trees, Counting trees, Tree traversal - Preorder traversal, Postorder traversal, Inorder traversal, Complete binary tree - Almost complete binary tree, Representation of algebraic structure of binary trees, Infix, prefix and postfix notation of an		

arithmetic expression - Infix notation, Prefix notation, Postfix notation, Evaluating prefix and postfix form of an expression, Shortest path algorithms - Dijkstra's algorithm, Minimal spanning trees - Weighted graph, Minimal spanning tree, Algorithm for minimal spanning tree - Kruskal's algorithm, Prim's algorithm	
Text Books: ***(As per IEEE format)***	
1	*"Discrete Mathematics and its applications" by Kenneth Rosen (William C Brown Publisher)*
2	*C. L. Liu and D. P. Mohapatra, "Elements of Discrete Mathematics", 4th Edition, McGraw-Hill.*
3	*"Applied Combinatorics" by Alan Tucker (Wiley Publishing company)*
4	*"Combinatorics: Topics, techniques, algorithms" by Peter J. Cameron (Cambridge University Press)*
5	*Graph Theory by Reinhard Diestel (Springer Verlag Publishing Company)*
6	*Introduction to Graph Theory by Douglas B. West (Prentice-Hall publishers)*
Reference Books: ***(As per IEEE format)***	
1	*K. H. Rosen,* Discrete Mathematics and Its Applications*, 8th ed. New York, NY, USA: McGraw-Hill Education, 2019.*
2	*R. P. Grimaldi,* Discrete and Combinatorial Mathematics: An Applied Introduction*, 5th ed. Boston, MA, USA: Pearson Education, 2004*

List of Tutorials

1. Problem solving based on propositional logic
2. Problem solving based on basic set theory
3. Problem solving based on relations and functions
4. Problem solving based on basic counting principles
5. Problem solving based on properties of binomial coefficients
6. Problem solving based on permutations, combinations
7. Problem solving based on combinatorial proof technique
8. Problem solving based on double counting
9. Problem solving based on Euler's and Fermat's theorems in calculations.
10. Problem solving based on Number Theory Theorems
11. Problem solving based on Paths, Cycles, and Connectivity
12. Problem solving based on Graph Theorems and Applications
13. Problem solving based on Tree Traversals
14. Problem solving based on Spanning Trees

CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '-' = no correlation.

CO	Program Outcome (PO)											PSO		
CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO 11	PS O1	PS O2	PS O3
CO1	3	1		1							2	2		
CO2	3	1				1			2		2	2	1	
CO3	3	2	3								2	2		
CO4	3	3	3	3							2	2		
CO5	3	3	1			1	2		1		2	2	1	
CO6	3	3	3	2		1		1			2	2	1	
Avg	3	2.3	1.6	1	0	0.5	0.3	0.1	0.5		2	2	0.5	0

CO – WK Mapping (Knowledge & Attitude Profiles) Sample Table

'✓' indicates the knowledge profile drawn upon by the CO.

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1		✓	✓						
CO2		✓	✓	✓					
CO3		✓	✓	✓					
CO4		✓	✓	✓	✓			✓	
CO5		✓	✓	✓	✓	✓	✓	✓	
CO6		✓	✓	✓	✓	✓		✓	

CO – WP Mapping (Complex Engineering Problems) Sample Table

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO1	✓		✓				
CO2	✓	✓	✓			✓	
CO3	✓	✓	✓			✓	

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO4	✓	✓	✓	✓		✓	
CO5	✓	✓	✓	✓	✓	✓	
CO6	✓	✓	✓	✓		✓	✓

S

CO – EA Mapping (Complex Engineering Activities)

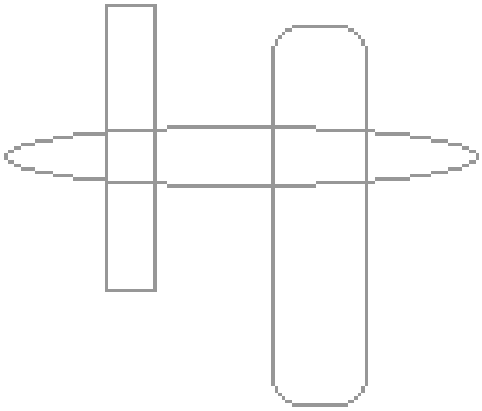
'✓' indicates the activity attribute addressed by the CO

CO	EA1	EA2	EA3	EA4	EA5
CO1	✓				
CO2	✓		✓		
CO3	✓		✓		
CO4	✓		✓		
CO5	✓		✓		
CO6	✓		✓	✓	✓

CO – SDG Mapping (Sustainable Development Goals) Sample Table

CO	SDG(s) Addressed	Justification / Linkage
CO1	4	Builds strong analytical and logical reasoning skills essential for lifelong learning and quality technical education
CO2	4	Enhances problem-solving and critical thinking abilities required in computing and engineering education
CO3	9	Provides mathematical foundations for cryptography, coding theory, and secure communication systems supporting innovation.
CO4	4	Number theory forms the basis of cybersecurity, encryption, and secure digital infrastructure.
CO5	11	Graph algorithms are widely used in transportation, communication, network optimization, and smart city applications

CO	SDG(s) Addressed	Justification / Linkage
CO6	9	Tree structures support efficient databases, file systems, AI, and network design, contributing to technological advancement.



FF No.: 654-A

CI2014: Course Name: Digital Systems and Microprocessor

Teaching Scheme	**Examination Scheme**
Credits : 3	CVV:20 Marks
Lectures : 30 Hrs/week	CP:30Marks
Practical : 14 Hrs/week	LAB:10 Marks
Tutorial : NA Hrs/week	MSE (W): 40 Marks

Prerequisites:	
●	Number systems and binary arithmetic, Boolean algebra and logic gates, Basic digital electronics, Fundamentals of computer organization ,Basic programming concepts
Course Objectives:	
CO1	To develop an understanding of number systems, Boolean algebra, logic minimization techniques, and digital circuit design methodologies.
CO2	To impart knowledge of combinational and sequential logic circuits, including adders, subtractors, flip-flops, registers, and counters.
CO3	To introduce Algorithmic State Machines (ASM) and Programmable Logic Devices (PLDs) for the design and implementation of digital systems.
CO4	To familiarize students with the architecture, features, operating modes, and addressing modes of modern microprocessors with emphasis on the 80386 processor.
CO5	To enable students to understand and apply various microprocessor instruction sets for arithmetic, logical, data transfer, and control operations.
CO6	To provide knowledge of system architecture concepts, system registers, memory management, interrupts, and exception handling mechanisms in microprocessor-based systems.
Course Outcomes:	
1.	Understand number systems and Simplify Boolean algebraic expressions using Minimization Technique.
2.	Design and Implement Combinational and Sequential circuits
3.	Design simple real-world applications using ASM and PLD.

4.	Illustrate the advanced features of the 80386 microprocessor and differentiate its various operating modes and addressing modes.
5.	Exhibit skill of assembly language programming for the application.
6.	Differentiate various types of system registers, interrupts, and exceptions.

Section1:	**Topics/Contents**

Unit 1: Fundamentals of Digital Electronics

Number System, Logic Minimization – Sum of Product (SOP) form, Product of Sum (POS) form, Representation and simplification of logical functions, Minimization of SOP and POS forms using K-Maps (up to 4 variables), Don't Care Condition, and Quine-McCluskey Technique.

Unit 2: Combinational and Sequential Circuits

Code converter: BCD, Excess-3, Gray code, Binary Code. Half- Adder, Full Adder, Half Subtractor, Full Subtractor, Binary Adder (IC 7483), BCD adder.

Flip-flop: SR, JK, D, T. Registers: Buffer register, shift register (SISO, SIPO, PISO& PIPO). Register Counters: Asynchronous Counter, Synchronous Counter, Ring counters, Johnson Counter, Modulus of the counter (IC 7490).

Unit 3: ASM & PLDs

Algorithmic State Machines: Finite State Machines (FSM) and ASM, ASM charts, notations, construction of ASM chart and realization for sequential circuits.

PLDS: PLD, Programmable Logic Array (PLA), Programmable Array Logic (PAL), Designing combinational circuits using PLDs.

Section2:	**Topics/Contents**

UNIT- 4 Introduction to 80386 - (5 Hours)

Introduction to Ideal Microprocessor – Data Bus, Address Bus, Control Bus, 8086 Architecture , 80386 DX Features and Architecture, Programmers Model, Operating modes, Addressing modes and data types.

Unit 5: Applications Instruction Set - (5 Hours)

Data Movement Instructions, Binary Arithmetic Instructions, Decimal Arithmetic Instructions, Logical Instructions, Control Transfer Instructions, String and Character Transfer Instructions, Instructions for Block Structured Language, Flag Control Instructions,, Segment Register Instructions, Miscellaneous Instructions.

Unit 6: System Architecture, Interrupts and Exception - (5 Hours)

Systems Registers - Systems flags, Memory Management registers, Control registers, Debug registers, Test registers.

Interrupts and Exceptions: Identifying Interrupts & Exceptions, Interrupt Tasks and Interrupt Procedures, Error Code, and Exception Conditions

Reference Books: Print Book ***(As per IEEE format)***

1	T. L. Floyd, "Digital Fundamentals", 9th edition, Pearson International Edition.
2	R. P. Jain, "Modern digital electronics", 4th edition, TMH Publication.
3	Anand Kumar, "Fundamentals of digital circuits" 4th edition, PHI publication
4	DouglasHall,"Microprocessors&Interfacing",McGrawHill,Revised2Edition,2006ISBN0- 07-100462-9
5	Chris H. Pappas, William H. Murray, "80386 Microprocessor Handbooks", McGraw-Hill Osborne Media, ISBN-10: 0078812429, 13:978-0078812422.
6	Brey,BarryB,"8086/8088,80286,80386and80486AssemblyLanguageProgramming", Prentice Hall, ISBN: 13:9780023142475.

Reference Books: Electronic Book ***(As per IEEE format)***

1	Intel 80386 Programmer's Reference Manual 1986, Intel Corporation, Order no.: 231630-011, December 1995
2	Intel 80386 Hardware Reference Manual 1986, Intel Corporation, Order no.: 231732-001, 1986.

List of Practicals:

1. Realize Full Adder and Sub tractor using a) Basic Gates and b) Universal Gates.
2. Design and implement Code converters - Binary to Grey and BCD to Excess-3.
3. Design and Realization of BCD Adder using 4-bit Binary Adder (IC 7483).
4. Design of Ripple Counter using JK-Flip Flops.

5. Design 3 bit Synchronous Up/Down Counter using JK-Flip Flop.
6. Design and implement Mod -N counter using IC-7490.
7. Study of Shift Registers (SISO,SIPO, PISO, PIPO)
8. Write an X86/64 ALP to accept five 64 bit Hexadecimal numbers from user and store them in an array and display the accepted numbers.
9. Write an X86/64 ALP to accept a string and to display its length.
10. Write an X86/64 ALP to find the largest of given Byte/Word/Dword/64-bit numbers.
11. Write a switch case driven X86/64 ALP to perform 64-bit hexadecimal arithmetic operations (+,-,*, /) using suitable macros. Define procedure for each operation.
12. Write an X86/64 ALP to count number of positive and negative numbers from the array.
13. Write X86/64 ALP to convert 4-digit Hex number into its equivalent BCD number and 5-digit BCD number into its equivalent HEX number. Make your program user friendly to accept the choice from user for: (a) HEX to BCD b) BCD to HEX (c) EXIT

CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '-' = no correlation.

CO	Program Outcome (PO)											PSO		
CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2	PSO 3
CO1	3	2	1	-	-	-	-	-	-	-	-	3	1	-
CO2	3	3	3	2	2	-	-	-	-	-	-	3	3	2
CO3	3	3	3	2	3	1	-	-	1	1	-	3	3	2
CO4	3	2	1	-	2	-	-	-	-	-	-	3	2	1
CO5	3	3	2	2	3	-	-	-	-	1	-	3	3	3
CO6	3	2	1	1	2	-	-	-	-	-	-	3	2	2
Avg	3	2.5	1.8	1.1	2	0.16	-	-	0.16	0.33	-	3	2.3	1.6

CO – WK Mapping (Knowledge & Attitude Profiles) Sample Table

'✓' indicates the knowledge profile drawn upon by the CO.

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1			✓	✓				✓	

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO2		✓		✓	✓	✓			
CO3		✓		✓		✓		✓	
CO4		✓		✓		✓			
CO5				✓		✓		✓	
CO6				✓	✓	✓		✓	

CO – WP Mapping (Complex Engineering Problems) Sample Table

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO1	✓						
CO2	✓		✓				
CO3	✓		✓				
CO4	✓	✓	✓				
CO5	✓	✓	✓				
CO6	✓		✓				✓

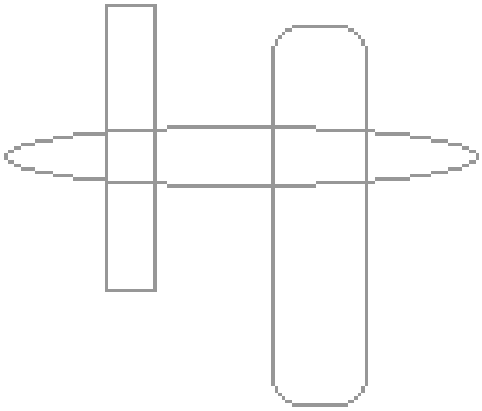
CO – EA Mapping (Complex Engineering Activities)

'✓' indicates the activity attribute addressed by the CO

CO	EA1	EA2	EA3	EA4	EA5
CO1	✓				
CO2	✓		✓		
CO3	✓		✓		
CO4	✓		✓		
CO5	✓		✓		
CO6	✓		✓	✓	✓

CO – SDG Mapping (Sustainable Development Goals) Sample Table

CO	SDG(s) Addressed	Justification / Linkage
CO1	4, 9	Builds the analytical foundation (SDG 4) that drives data-led innovation (SDG 9).
CO2	9	Supervised models power industrial and predictive applications (SDG 9).
CO3	3, 9	Statistical learning supports health diagnostics (SDG 3) and broader innovation (SDG 9).
CO4	8, 9	Classifiers enable fraud detection and economic applications (SDG 8, 9).
CO5	9	Decision-tree and ensemble models support data-driven industry solutions (SDG 9).
CO6	9, 11	Clustering / RL enable smart-city and resource applications (SDG 9, 11).



FF No.: 654-A

CI2016: Engineering Design and Innovations-I & II

Credits:.2 **Lab: 2 hours/Week**

s

Teaching Scheme	Examination Scheme
Credits : 2	CP: - Marks
Lectures : - Hrs/week	GD/PPT/HA: - Marks
Practical : - Hrs/week	MSE (W/O): 30 Marks
Tutorial : - Hrs/week	ESE (W/O): 70 Marks

Prerequisites:	Problem Based Learning
•	
Course Objectives:	
CO1	To develop critical thinking and problem-solving ability by exploring and proposing solutions to realistic/social problems.
CO2	To Evaluate alternative approaches, and justify the use of selected tools and methods,
CO3	To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
CO4	To engage students in rich and authentic learning experiences.
CO5	To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
CO6	To develop an ecosystem to promote entrepreneurship and research culture among the students
Course Outcomes:	
	After completion of the course, student will be able to
1.	Identify the real life problem from societal need point of view
2.	Choose and compare alternative approaches to select most feasible one
3.	Analyze and synthesize the identified problem from technological perspective
4.	Design the reliable and scalable solution to meet challenges
5.	Evaluate the solution based on the criteria specified
6.	Inculcate long life learning attitude towards the societal problems

Section1:	Topics/Contents

The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guideline mentioned herewith.

Group Structure: • There should be a team/group of 4-5 students. Computer Science & Engineering (Artificial Intelligence) –
A supervisor/mentor teacher assigned to individual groups. · It is useful to group students of different abilities and nationalities together.
Selection of Project/Problem:

- Students must focus to initiate the task/idea. The idea inception and consideration shall be from following areas as a real world problem:
- Health Care, Agriculture, Defense, Education, Smart City, Smart Energy, Swaccha Bharat Abhiyan, Environment, Women Safety.
- This is the sample list to start with. Faculty and students are free to include other areas which meet the society requirements at large.
- The model begins with the identifying of a problem, often growing out of a question or "wondering". This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.
- A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students' wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an interdisciplinary approach in both the analysis and solving phases.
- By exemplarity, a problem needs to refer back to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.

Teacher's Role in PCL :

- Teacher is not the source of solutions rather he will they act as the facilitator and mentor.
- To utilize the principles of problems solving, critical thinking and metacognitive skills of the students.
- To aware the group about time management.
- Commitment to devote the time to solve student's technical problems and interested in helping students to empower them better.

Student's Role in PCL:

- Students must have ability to initiate the task/idea .they should not be mere imitators.
- Students working in PCL must be responsible for their own learning.
- Students must quickly learn how to manage their own learning, Instead of passively receiving instruction.

- Students in PCL are actively constructing their knowledge and understanding of the situation in groups.
- Students in PCL are expected to work in groups.
- They have to develop interpersonal and group process skills, such as effective listening or coping creatively with conflicts.

Developing Inquiry Skills:

- Students in PCL are expected to develop critical thinking abilities by constantly relating: What they read to do? What they want to do with that information?
- They need to analyze information presented within the context of finding answers.
- Modeling is required so that the students can observe and build a conceptual model of the required processes.
- Use the following mechanism to maintain the track of moving towards the solution. How effective is …….? How strong is the evidence for ……? How clear is ……?
- What are the justifications for thinking? Why is the method chosen?
- What is the evidence given to justify the solution?

Literature Survey – To avoid reinvention of wheel:

- It is integral part of self- directed learning
- Identify the information needed to solve a given problem or issue
- Be able to locate the needed information
- Use the information to solve the given problem effectively. • Skills required by students in information literacy include:
- How to prepare the search? How to carry out the research • Sorting and assessing of information in general

Use of Research Methodology:
investigation, collaboration, comprehension, application, analysis, synthesize and evaluation
Focus on following skills while working in a team to reach to solution:

- Collaborative learning
- Interpersonal Skills
- Resources Evaluation
- Metacognitive Skills
- Reflection Skills

EDD Sample Case Studies : -
With the adaptation of industry communication standards, Raspberry Pi and Sensors, following projects
can be taken up:
1) Design a deployable product for soil moisture detection
2) Design a deployable product for temperature detection
3) Design a deployable product for pressure detection
3) Design a deployable product smoke detection
4) Design a deployable product for motion detection
5) Design a deployable product for collision detection
6) Design a deployable product for sound detection

Text Books: ***(As per IEEE format)***

1	By Terry Barrett. All Ireland Society for higher education (AISHE).

	A new model of problem-based learning. ISBN:978-0-9935254-6-9; 2017
2	By Mahnaz Moallem, woei hung and Nada Dabbagh, Problem Based Learning. Wiley Publishers. 2019.
3	By Robert Robart Capraro, Mary Margaret Capraro Stem Project based learning and integrated science, Technology, Engineering and mathematics approach.
Reference Books: *(As per IEEE format)*	
1	De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-based learning in engineering. Rotterdam: Sense Publishers. 2007.
2	Gopalan. Project management core textbook, second edition, Indian Edition , by

CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '-' = no correlation.

CO	Program Outcome (PO)											PSO		
CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2	PSO 3
CO1	2	3	1			2			2		1	2	1	1
CO2	2	3	2	2	1				2			2	2	1
CO3	2	3	2	3	2				2			2	3	1
CO4	2	2	3	2	2	2			2	1		2	3	2
CO5	2	2	2	3	2	2	1		2	1		1	3	2
CO6		1				2	2	1	2	1	3	1	2	3
Avg	1.83	2.33	2.2	2.4	2.5	2	1.5	2	2	1.33	2	1.67	2.5	1.83

CO – WK Mapping (Knowledge & Attitude Profiles) Sample Table

'✓' indicates the knowledge profile drawn upon by the CO.

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1	✓	✓		✓					✓
CO2		✓	✓	✓	✓				
CO3		✓	✓	✓	✓	✓			
CO4			✓	✓	✓	✓	✓		
CO5				✓	✓	✓	✓	✓	
CO6	✓			✓		✓		✓	✓

CO – WP Mapping (Complex Engineering Problems) Sample Table

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO1	✓	✓	✓		✓		✓
CO2	✓	✓	✓	✓	✓		
CO3	✓	✓	✓	✓	✓	✓	
CO4	✓	✓	✓	✓	✓	✓	✓
CO5	✓	✓	✓	✓	✓	✓	✓
CO6			✓		✓	✓	✓

CO – EA Mapping (Complex Engineering Activities)

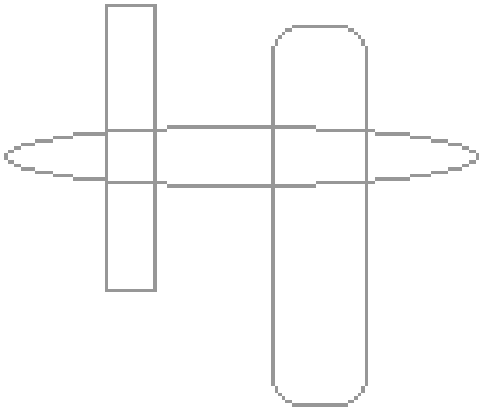
'✓' indicates the activity attribute addressed by the CO

CO	EA1	EA2	EA3	EA4	EA5
CO1	✓	✓			
CO2	✓	✓	✓		
CO3	✓	✓	✓	✓	
CO4		✓	✓	✓	✓
CO5			✓	✓	✓
CO6				✓	✓

CO – SDG Mapping (Sustainable Development Goals) Sample Table

CO	SDG(s) Addressed	Justification / Linkage
CO1	9, 11	Identifies real-life societal problems and community needs, promoting sustainable infrastructure and resilient communities (SDG 9, SDG 11).
CO2	9, 12	Compares alternative solutions to select feasible, efficient, and resource-conscious approaches that support innovation and responsible practices (SDG 9, SDG 12).
CO3	9	Analyzes technological aspects of the identified problem to develop innovative engineering solutions (SDG 9).

CO	SDG(s) Addressed	Justification / Linkage
CO4	9, 11	Designs reliable and scalable engineering solutions that improve infrastructure and address societal challenges (SDG 9, SDG 11).
CO5	9, 12	Evaluates solutions using defined criteria to ensure quality, efficiency, sustainability, and responsible resource utilization (SDG 9, SDG 12).
CO6	4, 17	Promotes lifelong learning, continuous professional development, and collaboration for solving societal problems (SDG 4, SDG 17).



FF No. : 654

HS201: Reasoning and Aptitude Development

Teaching Scheme	Examination Scheme
Credits : 1	CP: -- Marks
Lectures : 1 Hrs/week	GD/PPT/HA: -- Marks
Practical : -- Hrs/week	MSE (W/O): -- Marks
Tutorial : -- Hrs/week	ESE (W/O): 100 Marks

Prerequisites:	
Basic knowledge of English grammar and vocabulary.	
Fundamental understanding of sentence construction and communication skills.	
Basic arithmetic and mathematical concepts (Class XII level).	
Elementary logical and analytical reasoning skills.	
Familiarity with problem-solving techniques.	
Basic computer literacy and programming concepts.	
Exposure to at least one programming language (C/C++/Java/Python) is desirable.	
Course Objectives:	
CO1	Develop proficiency in English grammar, vocabulary, reading comprehension, and effective written communication.
CO2	Strengthen logical, analytical, critical, and reasoning abilities to solve diverse aptitude-based problems.
CO3	Apply quantitative and mathematical concepts to solve real-world numerical and analytical problems efficiently.
CO4	Interpret, analyze, and synthesize information from textual, graphical, tabular, and numerical data for informed decision-making.
CO5	Develop computational thinking and programming aptitude through logic building, pseudocode, algorithmic problem solving, and debugging techniques.
CO6	Enhance coding proficiency, optimization skills, and problem-solving abilities to perform effectively in programming assessments, campus recruitment tests, and technical interviews.
Course Outcomes:	
After completion of the course, student will be able to	
1.	Apply appropriate English language skills, including grammar, vocabulary, reading comprehension, and effective communication techniques to improve verbal and written proficiency.
2.	Analyze logical reasoning problems, critical thinking scenarios, and data interpretation tasks to arrive at accurate and evidence-based conclusions.

3.	Solve quantitative aptitude problems using mathematical concepts, algebraic reasoning, probability, geometry, and analytical techniques in real-life contexts.
4.	Develop efficient algorithmic solutions by applying computational thinking, programming fundamentals, debugging, optimization techniques, and time complexity analysis to solve coding problems.

Section1:	**Topics/Contents**

English Language and Communication Skills: English tenses (Present, Past, Future), active and passive voice, sentence correction, parts of speech, syntax, sentence structure, subject–verb agreement, reading comprehension, skimming and scanning, inference, tone, theme, intent identification, contextual vocabulary.

Logical Reasoning and Analytical Thinking: Deductive reasoning (syllogisms, coding–decoding, statement–conclusion, data sufficiency, directional sense, logical sequencing), inductive reasoning (analogies, classification, number series, sequence completion, trend identification), abductive and critical reasoning (assumptions, arguments, cause–effect, logical evaluation, decision making), information gathering, interpretation and synthesis using graphs, charts, tables and multi-source data.

Section2:	**Topics/Contents**

Quantitative Aptitude and Mathematical Reasoning: Number systems, integers, fractions, decimals, divisibility, HCF and LCM, prime and rational numbers, algebraic expressions, linear equations and inequalities, proportional reasoning, speed–time–distance, profit and loss, percentages, averages, ages, mixtures, permutations and combinations, probability, geometry, spatial reasoning, pattern recognition and quantitative problem solving.

Programming Aptitude and Automata Skills: Logic building, pseudocode, programming fundamentals, basic coding problems, program correctness, execution efficiency, intermediate coding challenges, data structures, algorithms, debugging, optimization techniques, time complexity analysis, industry best coding practices, Automata Pro IT, Automata Pro Max and comprehensive mock assessments.

Text Books: ***(As per IEEE format)***

1	R. S. Aggarwal, *Quantitative Aptitude for Competitive Examinations*, Revised and Enlarged Edition. New Delhi, India: S. Chand and Company Ltd., 2017.
2	R. S. Aggarwal, *A Modern Approach to Verbal & Non-Verbal Reasoning*. New Delhi, India: S. Chand Publishing.
3	Dr. M. B. Lal and A. Gupta, *CSAT Logical Reasoning & Analytical Ability*. Agra, India: Upkar Prakashan.
4	G. Laakmann McDowell, *Cracking the Coding Interview: 189 Programming Questions and Solutions*, 6th ed. Palo Alto, CA, USA: CareerCup, 2015.

Reference Books: *(As per IEEE format)*	
1	B. H. Dowden, *Logical Reasoning*. California State University, Sacramento, CA, USA, 2011.
2	*501 Challenging Logic and Reasoning Problems*, 2nd ed. New York, NY, USA: LearningExpress, LLC, 2005.
3	*Logic and Reasoning Problems*. LearningExpress Practice Series.
Link: https://drive.google.com/drive/folders/1fXLzuofNhMFq7skonrC0TpCdaCoT7_i0?usp=sharing	

CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '-' = no correlation.

CO	Program Outcome (PO)											PSO		
CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO 1	PSO 2	PSO 3
CO1	–	–	–	–	–	–	–	–	3	–	2			
CO2	2	2	–	–	–	–	–	–	1	–	2			
CO3	2	2	–	–	–	–	–	–	1	1	2			
CO4	2	2	3	3	3	–	1	–	1	1	3			
Avg	**2**	**2**	**3**	**3**	**3**	**–**	**1**	**–**	**1.5**	**1**	**2.25**			

CO – WK Mapping (Knowledge & Attitude Profiles)

'✓' indicates the knowledge profile drawn upon by the CO.

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1	–	✓	–	–	–	–	–	✓	✓
CO2	–	✓	–	–	–	–	–	✓	✓
CO3	–	✓	–	–	–	–	–	✓	✓
CO4	–	✓	–	–	–	–	–	✓	✓

CO – WP Mapping (Complex Engineering Problems)

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO1	–	–	–	–	–	–	–
CO2	✓	–	✓	✓	–	–	–
CO3	✓	–	✓	✓	–	–	–

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO4	✓	–	✓	✓	–	–	✓

CO – EA Mapping (Complex Engineering Activities)

'✓' indicates the activity attribute addressed by the CO

CO	EA1	EA2	EA3	EA4	EA5
CO1	–	–	–	–	–
CO2	–	–	✓	–	✓
CO3	–	–	✓	–	✓
CO4	✓	–	✓	–	✓

CO – SDG Mapping (Sustainable Development Goals)

CO	SDG(s) Addressed	Justification / Linkage
CO1	SDG 4	Develops proficiency in English communication, reading comprehension, and language skills, promoting inclusive and quality education and lifelong learning.
CO2	SDG 4	Enhances logical reasoning, critical thinking, and analytical decision-making skills, enabling responsible problem-solving and informed decision-making.
CO3	SDG 4, 8	Strengthens quantitative aptitude and analytical skills essential for employability, professional competence, and workplace productivity.
CO4	SDG 4, 8, 9	Develops computational thinking, programming, and algorithmic problem-solving skills required for innovation, digital transformation, and industry-ready engineering professionals.

- **SDG 4:** Quality Education
- **SDG 8:** Decent Work and Economic Growth
- **SDG 9:** Industry, Innovastion and Infrastructure

FF No.: 654-A

CI2015: Design Thinking

Credits:.1 **Lab: 2 hours/Week**

Teaching Scheme	**Examination Scheme**
Credits : 1	CP: - Marks
Lectures : Hrs/week	GD/PPT/HA: - Marks
Practical : Hrs/week	MSE (W/O): - Marks
Tutorial : Hrs/week	ESE (Direct Evaluation): 100 Marks

Prerequisites:	
●	Problem Based Learning, Project Centric Learning
Course Objectives:	
●	To provide ecosystem for students and faculty for paper publication and patent filing
Course Outcomes:	
	After completion of the course, student will be able to
1.	Understand the importance of doing Research
2.	Interpret and distinguish different fundamental terms related to Research
3.	Apply the methodology of doing research and mode of its publication
4.	Write a Research Paper based on project work
5.	Understand Intellectual property rights
6.	Use the concepts of Ethics in Research
7	Understand the Entrepreneurship and Business Planning

Section1:	**Topics/Contents**
What is research? Importance of Paper Publication and Patents Structure of Paper Journal Publication Publication in conference Literature Review Research Paper Writing Journal Ratings and Evaluation How to rate a Journal? Intellectual property (IP) Research Ethics Entrepreneurship	
Section2:	**Topics/Contents**
Structure of The paper Journal List (Top 50 Journals) Selection of the journal Use of various online journal selection tools Plagiarism checking Improving contents of the paper Patent drafting Patent search Filing of patent Writing answers to reviewer questions Modification in manuscript Checking of publication draft	

Text Books: *(As per IEEE format)*	
1	C. R. Kothari and G. Garg, *Research Methodology: Methods and Techniques*, 4th ed. New Delhi, India: New Age International Publishers, 2019
2	W. Booth, G. Colomb, J. Williams, J. Bizup, and W. FitzGerald, *The Craft of Research*, 5th ed. Chicago, IL, USA: The University of Chicago Press, 2024
3	R. M. Sarma, *Research Methodology*, 1st ed. New Delhi, India: PHI Learning Pvt. Ltd., 2019.
Reference Books: *(As per IEEE format)*	
1	A. Wallwork, *English for Writing Research Papers*, 2nd ed. Cham, Switzerland: Springer, 2016.
2	A. Wallwork, *English for Research: Usage, Style, and Grammar*, New York, NY, USA: Springer, 2013
3	P. Narayanan, *Patent Law*, 4th ed. Lucknow, India: Eastern Law House, 2017.

CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '-' = no correlation.

CO	**Program Outcome (PO)**											**PSO**		
CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO 1**	**PSO 2**	**PSO 3**
CO1	3	2	-	2	1	2	1	2	-	2	-	2	-	1
CO2	3	3	-	2	2	-	-	2	-	2	-	2	-	1
CO3	2	3	2	3	2	-	-	2	-	2	-	2	2	2
CO4	2	2	3	3	2	-	-	2	2	3	-	2	2	2
CO5	2	2	-	1	-	2	2	3	-	2	2	1	-	1
CO6	2	2	-	2	-	2	3	3	2	2	1	1	-	1
Avg	1	2	3	2	2	3	2	2	3	3	3	1	2	2

CO – WK Mapping (Knowledge & Attitude Profiles) Sample Table

'✓' indicates the knowledge profile drawn upon by the CO.

CO	**WK1**	**WK2**	**WK3**	**WK4**	**WK5**	**WK6**	**WK7**	**WK8**	**WK9**
CO1	✓	✓		✓					✓
CO2		✓	✓	✓	✓				
CO3		✓	✓	✓	✓	✓			

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO4			✓	✓	✓	✓	✓		
CO5				✓	✓	✓	✓	✓	
CO6	✓			✓		✓		✓	✓

CO – WP Mapping (Complex Engineering Problems) Sample Table

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO1	✓	✓	✓		✓		✓
CO2	✓	✓	✓	✓	✓		
CO3	✓	✓	✓	✓	✓	✓	
CO4	✓	✓	✓	✓	✓	✓	✓
CO5	✓	✓	✓	✓	✓	✓	✓
CO6			✓		✓	✓	✓

CO – EA Mapping (Complex Engineering Activities)

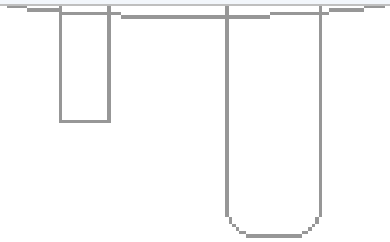
'✓' indicates the activity attribute addressed by the CO

CO	EA1	EA2	EA3	EA4	EA5
CO1	✓	✓			
CO2	✓	✓	✓		
CO3	✓	✓	✓	✓	
CO4		✓	✓	✓	✓
CO5			✓	✓	✓
CO6				✓	✓

CO – SDG Mapping (Sustainable Development Goals) Sample Table

CO	SDG(s) Addressed	Justification / Linkage
CO1	4, 9	Understanding the importance of research promotes quality education, scientific inquiry, innovation, and technological advancement for solving real-world problems.

CO	SDG(s) Addressed	Justification / Linkage
CO2	4, 9	Knowledge of research terminology and concepts strengthens analytical thinking, research competency, and innovation capabilities.
CO3	9, 17	Applying research methodology and publishing research fosters innovation, knowledge dissemination, and collaboration among researchers and institutions.
CO4	4, 9, 17	Writing research papers develops scientific communication, promotes lifelong learning, and contributes to global knowledge sharing through collaborative research.
CO5	9,12	Understanding Intellectual Property Rights encourages innovation, protection of inventions, responsible use of knowledge, and sustainable technological development.
CO6	4, 16	Applying research ethics promotes academic integrity, responsible conduct of research, transparency, fairness, and ethical decision-making.
CO7	8, 9	Entrepreneurship and business planning encourage innovation, job creation, sustainable economic growth, and technology-based enterprises.



FF No.: 654-A

CI2017: Advanced Data Structures and Algorithms

Teaching Scheme	Examination Scheme
Credits: 4	CP:30 Marks
Lectures: 3 Hrs/week	CVV: 20 Marks
Practical: 2 Hrs/week	Lab: 10 Marks
	ESE :40 Marks

Prerequisites:	Fundamentals of Data Structures
Course Objectives:	1. To interpret the concept of tree data structures. 2. To build graphs and represent them in various ways. 3. To apply hashing techniques on data. 4. To analyse algorithms and the working of the divide-and-conquer method. 5. To correlate greedy and dynamic programming methods. 6. To implement backtracking and branch-and-bound techniques.
Course Outcomes:	After completion of the course, student will be able to: CO1: Apply advanced tree and balanced search structures (AVL, RB-Tree, B/B+) to perform efficient searching, insertion, and deletion with complexity justification. CO2: Analyze and implement graph algorithms (MST, shortest path, connectivity) using correct representations and efficiency trade-offs. CO3: Apply heaps, priority queues, and hashing techniques (universal hashing, collision resolution) to design efficient dictionary and scheduling structures. CO4: Solve recurrences, analyze algorithms, and implement divide-and-conquer strategies with correctness and complexity proofs. CO5: Apply greedy and dynamic programming methods to solve optimization problems with appropriate problem decomposition and DP table formulation. CO6: Use backtracking and branch-and-bound techniques to solve combinatorial search problems with pruning and optimization strategies.

Course Relevance:

The course Advanced Data Structures and Algorithms is highly relevant to the Computer Engineering / Information Technology curriculum, as it provides in-depth knowledge of efficient data organization, algorithm design, and complexity analysis fundamental to solving large-scale and real-time computational problems. Students gain the ability to design, analyze, and implement advanced data structures and algorithms such as balanced trees, graphs, heaps, hashing techniques, and optimization strategies. These concepts underpin modern applications including databases, operating systems, compiler design, artificial intelligence, machine learning, networking, and distributed systems. By emphasizing algorithmic efficiency, correctness proofs, and problem-solving strategies, the course strengthens students' analytical and logical thinking skills, while preparing them for technical interviews, competitive programming, higher studies, and research, in alignment with NBA and Outcome-Based Education (OBE) requirements.

Section 1: Topics/Contents
Unit 1: Trees (05 Hours)
Linear and non-linear data structures, tree representation, tree and binary tree with their properties, converting a general tree to a binary tree, expression tree. Binary tree traversal: inorder, preorder and postorder. Binary Search Tree, binary search tree operations, Binary Search Tree – analysis, height, optimality. AVL Trees, Red-Black Trees, B-Trees & B+ Trees.
Unit II: Graphs (05 Hours)
Basic concepts, storage representation for graphs, adjacency matrix, adjacency list. Traversal techniques: Depth First Search, Breadth First Search, BFS & DFS (iterative + recursive). Minimum Spanning Tree – Prim's algorithm, Kruskal's algorithm; Dijkstra's shortest path algorithm; Bellman–Ford algorithm; Floyd–Warshall algorithm; A* search heuristic.
Unit III: Heap and Hashing Techniques (05 Hours)
Symbol table, heap data structure, min and max heap, heap sort, applications of heap (priority queue). Advanced heaps: Fibonacci heap. Hash table, scattered tables, hash functions, characteristics of a good hash function, methods to transform keys to hash values, collision resolution techniques, universal hashing, double hashing, load factor & rehashing.
Section 2: Topics/Contents
Unit IV: Introduction to Algorithms (05 Hours)
Algorithms: introduction, algorithm specification – pseudo-code conventions, algorithm analysis – asymptotic notations, Master theorem and substitution method to solve recurrence relations, recursion tree method. Divide and Conquer: general method, Quick sort, Merge sort and their time complexities.
Unit V: Greedy Algorithm and Dynamic Programming (05 Hours)

Greedy Algorithm: general method, knapsack problem, job sequencing with deadlines, coin change problem, Huffman coding. Dynamic Programming: general method, 0/1 knapsack problem, optimal binary search tree, travelling salesperson problem, matrix chain multiplication, longest common subsequence (LCS).
Unit VI: Backtracking and Branch and Bound (05 Hours)
Backtracking: general method, knapsack problem, 8-queens problem, Sudoku / constraint-based search, graph colouring. Branch and Bound: general method, bounding function design, knapsack problem, travelling salesperson problem, knapsack using B&B, TSP using B&B.

Text Books: (As per IEEE format)

1. E. Horowitz, S. Sahni, S. Rajasekaran; Computer Algorithms; 2nd ed., Universities Press.

2. A. V. Aho, J. E. Hopcroft, J. D. Ullman; Data Structures and Algorithms; Pearson Education, 1998, ISBN 0-201-43578-0.

3. R. Lafore; Data Structures and Algorithms in Java; 2nd ed., Pearson Education.

Reference Books: (As per IEEE format)

1. Y. Langsam, M. Augenstein, A. Tanenbaum; Data Structures Using C and C++; 2nd ed., Prentice Hall of India, 2002, ISBN 81-203-1177-9.

2. M. A. Weiss; Data Structures and Algorithm Analysis in C++; 4th ed., Pearson Education, 2002, ISBN 81-7808-670-0.

3. G. A. V. Pai; Data Structures and Algorithms; McGraw Hill, ISBN-13: 978-0-07-066726-6.

4. T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein; Introduction to Algorithms; 3rd ed., MIT Press, Cambridge, MA, USA, ISBN 978-81-203-4007.

MOOCs Links and Additional Reading Material:

1. NPTEL Course “Data Structures And Algorithms”, IIT Delhi, Prof. Naveen Garg – https://nptel.ac.in/courses/106102064

2. NPTEL Course “Programming and Data Structures”, IIT Kharagpur, Dr. P. P. Chakrabarti – https://nptel.ac.in/courses/106105085

List of Practical's:

Sr. No.	Title / Description
1	**Binary Tree & BST Operations** A. Create a binary tree and perform: (a) insert, (b) display, (c) find depth of the tree, (d) display leaf nodes, (e) create a copy of the tree.

	B. Implement a binary search tree and perform: (a) insert, (b) delete, (c) display, (d) search, (e) BFS (level-wise print). Construct a BST from a given set of keys; perform recursive and iterative traversals, compute tree height, and verify whether the BST is height-balanced.
2	**Expression Tree** Construct an expression tree from a postfix expression and perform recursive and non-recursive traversals.
3	**AVL Tree with Rotations** Implement insertion and deletion operations in an AVL tree. After each update, identify and display the rotation performed (LL, LR, RL, or RR).
4	**Graph Modelling – Social Network** Model a friend network (e.g., Facebook) as a graph, with each node representing a user and each link representing a friendship, storing attributes such as date of birth and comment count. (1) Find the user with the maximum number of friends. (2) Find the user(s) with the maximum and minimum comments. (3) Find users with a birthday in the current month. Hint: use adjacency-list representation and perform DFS and BFS traversals.
5	**B-Tree / B+ Tree Insertion** Insert a sequence of keys into a B-Tree or B+ Tree of a specified order. Show all node splits, promotions, and the final hierarchical structure.
6	**Minimum Spanning Tree** Given an undirected weighted graph, find the total weight of the minimum spanning tree and construct one such MST using Prim's or Kruskal's algorithm.
7	**Shortest Path Algorithms** A student wants to travel from source (A) to destination (B) and books a cab. Compute the shortest path avoiding real-time traffic using Dijkstra, Bellman–Ford, and Floyd–Warshall algorithms. Compare the resulting shortest-path costs and detect any negative-cycle conditions.
8	**Heap-based Max/Min Search** Read the marks obtained by second-year students in an online examination for a particular subject. Using a heap data structure, find the maximum and minimum marks obtained, and analyse the algorithm.
9	**Hashing – Telephone Directory**

<table>
<tr><td></td><td>Consider a telephone-book database of N clients. Use a hash table to quickly look up a client's telephone number. Implement two collision-handling techniques and compare them by the number of comparisons required to locate a set of telephone numbers.</td></tr>
<tr><td>10</td><td>Sorting – Merge Sort & Quick Sort
A. Read the credits obtained by students in various club activities and sort the credits using merge sort.
B. Implement quick sort to sort students by credits received over the last two years and display the names of the top three students.</td></tr>
<tr><td>11</td><td>Fractional Knapsack (Greedy)
Given a set of items with weights and values and a knapsack capacity, find the maximum total value obtainable by selecting fractions of items using the greedy strategy.</td></tr>
<tr><td>12</td><td>Job Sequencing with Deadlines
There are 5 jobs, each with a deadline and an associated profit. Select a subset of jobs that maximizes total profit while ensuring each selected job is completed within its deadline. Display the selected jobs with their deadlines.
<table>
<tr><th>Job</th><th>Deadline</th><th>Profit</th></tr>
<tr><td>A</td><td>2</td><td>100</td></tr>
<tr><td>B</td><td>1</td><td>19</td></tr>
<tr><td>C</td><td>2</td><td>27</td></tr>
<tr><td>D</td><td>1</td><td>25</td></tr>
<tr><td>E</td><td>3</td><td>15</td></tr>
</table></td></tr>
<tr><td>13</td><td>Travelling Salesperson Problem
Given a 2-D matrix tsp[][], where each row holds the distances from the indexed city to all other cities (–10" denotes no path between two indexed cities), print the minimum cost of the TSP cycle.
Sample input: tsp[][] = {{-1, 10, 15, 20}, {10, -1, 35, 25}, {15, 35, -1, 30}, {20, 25, 30, -1}}.</td></tr>
<tr><td>14</td><td>N-Queens (Backtracking)
Place each queen one by one in different rows, starting from the topmost row. While placing a queen in a row, check for clashes with already-placed queens. If a safe column is found, mark the row and column as part of the solution; otherwise backtrack (undo the placement of the most recent queen) and return false.</td></tr>
</table>

List of Project Areas:

(Suggested indicative areas – to be finalized by the course coordinator)

1. Design and performance comparison of self-balancing BSTs (AVL vs Red-Black) for a real-time indexing application.
2. Graph-based route optimization / campus navigation system using Dijkstra and A* search.
3. Distributed hash-table-based key-value store with collision handling and load-balancing analysis.
4. Huffman-coding-based file compression utility with compression-ratio benchmarking.
5. Job-scheduling simulator using greedy and dynamic-programming strategies with visual analytics.
6. Constraint-satisfaction solver (e.g., Sudoku / N-Queens) using backtracking and branch-and-bound with pruning-efficiency comparison.

CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '-' = no correlation.

CO	Program Outcome (PO)											PSO		
CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	2	2	2	2	1	-	-	-	1	-	2	-	-
CO2	3	3	3	3	2	1	-	-	-	1	-	3	2	-
CO3	3	2	2	2	2	1	-	-	-	1	-	2	2	-
CO4	3	3	3	3	2	1	-	-	-	1	-	3	2	2
CO5	3	3	3	2	1	2	-	-	1	2	-	3	-	2
CO6	3	2	3	2	2	2	-	-	1	2	-	3	-	2
Avg	3	3	3	2	2	1	-	-	1	1	-	3	2	2

CO – WK Mapping (Knowledge & Attitude Profiles)

'✓' indicates the knowledge profile drawn upon by the CO. (Drafted mapping – to be reviewed against the institute's WK1–WK9 definitions.)

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1	-	✓	✓	✓	-	-	-	✓	-
CO2	-	✓	✓	✓	✓	-	-	-	-

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO3	-	✓	✓	✓	-	-	-	✓	-
CO4	-	✓	✓	-	-	-	-	✓	-
CO5	-	✓	✓	✓	✓	-	-	-	-
CO6	-	✓	✓	✓	✓	-	-	✓	-

CO – WP Mapping (Complex Engineering Problems)

Drafted mapping – to be reviewed against the institute's WP1–WP7 definitions.

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO1	✓	-	-	-	-	-	-
CO2	✓	-	✓	-	-	-	-
CO3	✓	-	✓	-	-	-	-
CO4	✓	✓	-	-	-	-	-
CO5	✓	✓	✓	-	-	-	-
CO6	✓	✓	✓	-	-	-	✓

CO – EA Mapping (Complex Engineering Activities)

'✓' indicates the activity attribute addressed by the CO. (Drafted mapping – to be reviewed against the institute's EA1–EA5 definitions.)

CO	EA1	EA2	EA3	EA4	EA5
CO1	✓	-	-	-	-
CO2	✓	-	✓	-	-
CO3	✓	-	-	-	-
CO4	✓	-	✓	-	-
CO5	✓	-	✓	-	-
CO6	✓	-	✓	✓	-

CO – SDG Mapping (Sustainable Development Goals)

CO	SDG(s) Addressed	Justification / Linkage

CO1	4, 9	Builds the analytical foundation (SDG 4) for efficient search-and-storage structures used in digital infrastructure (SDG 9).
CO2	9, 11	Graph and shortest-path algorithms underpin transportation, network, and smart-city routing systems (SDG 9, 11).
CO3	9	Hashing and priority-queue structures power large-scale indexing and retrieval systems central to digital industry (SDG 9).
CO4	4, 9	Algorithmic analysis and correctness proofs strengthen computational literacy (SDG 4) and efficient software systems (SDG 9).
CO5	8, 9	Optimization via greedy/DP methods supports resource-efficient, cost-effective industrial and economic applications (SDG 8, 9).
CO6	9	Backtracking and branch-and-bound techniques enable constraint-based problem solving used in planning and design tools (SDG 9).

CO Attainment Levels:

Weights for attainment levels: L1 – Easy – 0.75, L2 – Comfortable – 0.70, L3 – Medium – 0.65, L4 – Somewhat difficult – 0.60, L5 – Difficult – 0.55.

CO1 – __, CO2 – __, CO3 – __, CO4 – __, CO5 – __, CO6 – __ (to be computed from actual assessment data)

Future Courses Mapping:

The knowledge and skills acquired in this course serve as a prerequisite and strong support for advanced courses such as Design and Analysis of Algorithms, Database Management Systems, Operating Systems, Computer Networks, Artificial Intelligence, Machine Learning, and Distributed Systems. Mastery of these concepts enables students to better understand algorithmic efficiency, system performance, and intelligent decision-making in subsequent courses.

Job Mapping:

The competencies developed through this course prepare students for roles that require strong algorithmic thinking and efficient data handling. Graduates are well-equipped for positions such as Software Engineer, Backend Developer, Data Engineer, Algorithm Engineer, System Software Developer, and Research Engineer, where optimized data structures and algorithms are critical.

FF No.: 654-A

CI2018: Machine Learning

Teaching Scheme	Examination Scheme
Credits : 3	CP: Marks
Lectures : 2 Hrs/week	GD/PPT/HA: Marks
Practical : 2 Hrs/week	MSE (W/O): Marks
Tutorial : Hrs/week	ESE (W/O): Marks

Prerequisites:	Probability and Statistics
•	
Course Objectives:	
CO1	Explain the fundamental concepts of Machine Learning, types of learning paradigms, data types, feature engineering techniques, and the role of data preprocessing in developing intelligent systems.
CO2	Apply data preprocessing, transformation, validation, and dimensionality reduction techniques to prepare datasets for effective machine learning model development.
CO3	Analyze and evaluate the performance of machine learning models using appropriate classification and regression evaluation metrics, validation techniques, and diagnostic measures.
CO4	Design and implement regression-based supervised learning models using suitable optimization and regularization techniques to solve real-world prediction problems.
CO5	Develop and compare supervised classification models such as Support Vector Machines, K-Nearest Neighbors, Decision Trees, Random Forests, and Naïve Bayes for solving classification problems.
CO6	Apply unsupervised learning techniques, including clustering algorithms, to discover meaningful patterns, group similar data instances, and interpret clustering results for decision-making.
Course Outcomes:	
	After completion of the course, student will be able to
	Explain the fundamental concepts of Machine Learning, learning paradigms, data preprocessing techniques, feature engineering methods, and dataset preparation for building intelligent systems. *(Bloom's Level: L2 – Understand)*

	Apply data preprocessing, transformation, validation, and dimensionality reduction techniques to prepare high-quality datasets for machine learning applications. *(Bloom's Level: L3 – Apply)* Analyze the performance of machine learning models using appropriate evaluation metrics for binary and multiclass classification, regression, and cross-validation techniques. *(Bloom's Level: L4 – Analyze)* Develop and evaluate supervised regression models using linear, polynomial, logistic, ridge, and lasso regression techniques, while addressing issues such as overfitting, underfitting, bias, and variance. *(Bloom's Level: L3 – Apply)* Design and implement supervised classification models using Support Vector Machines, K-Nearest Neighbors, Decision Trees, Random Forests, and Naïve Bayes for solving real-world classification problems. *(Bloom's Level: L3 – Apply)* Apply unsupervised learning techniques, including K-Means, K-Medoids, Gaussian Mixture Models, and Hierarchical Clustering, to identify hidden patterns and extract meaningful insights from unlabeled datasets. *(Bloom's Level: L3 – Apply)*

Section1:	**Topics/Contents**

Unit I: Introduction to Machine Learning (5 Hours)

Basics of Machine Learning: Definitions, ML vs AI vs Data Science, Real life applications.

Types of Learning: Supervised Learning Unsupervised Learning, Semi-Supervised Learning, Reinforcement Learning. Types of Data: Qualitative and Quantitative data, Data Sampling Techniques. Concept of Feature: Feature construction, Feature Selection and Transformation, Dataset Preparation: Training Vs. Testing Dataset.Dataset Preparation: Training Vs. Testing Dataset, Identifying Missing values and approaches, Noisy Data Extraction, Data Cleaning as a process, Data reduction

Unit II: Data Pre-processing and Transformation (5 Hours)

Data Transformation and Discretization: Data Transformation by Normalization, Discretization by Binning Discretization by Histogram Analysis Discretization by Cluster.

Dataset Validation Techniques: Hold-out, k-fold Cross validation, Leave-One-Out Cross-Validation (LOOCV).Dimensionality Reduction: Introduction, Subset Selection, Principal Components Analysis, Factor Analysis, Multidimensional Scaling, Linear Discriminant Analysis Models of Machine learning: Geometric model, Probabilistic Models, Logical Models, Grouping models, Parametric and non-parametric models

Unit III: Model Evaluation and Classification Metrics (4 Hours)

Binary Classification: Metrics for Evaluating Classifier Performance - Confusion Matrix, Accuracy, Precision, Recall, ROC Curves, AUC, F-Measure.Multi-class Classification: Metrics

for Evaluating Classifier Performance - Per-class Precision and Per-Class Recall, weighted average precision and recall -with example, Handling more than two classes

Section2:	Topics/Contents

Unit IV: Supervised Learning - Regression Models (05 Hours)

Introduction, Univariate Regression – Model Representation, Least-Square Method for finding values of the regression coefficients, Multiple Regression, Logistic Regression. Multivariate Regression –Model Representation, Polynomial Regression. Performance enhancement with Regularization techniques. Regularized Regression – Overview of Ridge and Lasso Regression Cost Functions: MSE, MAE, R-Square, Performance Evaluation, Optimizing Simple Linear Regression with Gradient Descent Algorithm. Generalization Issues - Overfitting Vs. Underfitting, Bias Vs. Variance.

Unit V: Supervised Models – Classification Models (05 Hours)

SVM: Introduction to SVM, Computing Support Vector Classifier for classification, Soft Margin SVM, Introduction to various SVM Kernel to handle non-linear data – RBF, Gaussian, Polynomial, Sigmoid: Introduction to K-Nearest neighbour. Decision trees: Overview, decision tree algorithm, evaluating a decision tree using Gini Index and Entropy, Random Forest: Bagging, Boosting. Naive Bayes: Bayes Theorem, Naïve Bayes Classifier, smoothing, diagnostics. Diagnostics of classifiers

Unit VI: Unsupervised Learning (04 Hours)

Cluster Analysis: Basic Concepts and Methods, Proximity Matrices, Clustering Algorithms: K-mean, Gaussian Mixtures as Soft K-means Clustering Partitioning Methods: k-Means, A Centroid Based Methods: k-Medoids, Hierarchical Methods: Agglomerative versus Divisive Hierarchical Clustering, Dendrogram for hierarchical clustering

Text Books: ***(As per IEEE format)***

1	Data Science and Machine Learning: Mathematical and Statistical Methods By D.P. Kroese, Z.I. Botev, T. Taimre, R. Vaisman, *Chapman and Hall/CRC, Boca Raton, 2019*.

Reference Books: ***(As per IEEE format)***

List of Practical's:

1. Perform the following operations using Python on suitable data sets:

a. Read data from different formats (like csv, xls)

b. Find Shape of Data
c. Find Missing Values
d. Find data type of each column
e. Finding out Zero's
f. Indexing and selecting data, sort data,
g. Describe attributes of data, checking data types of each column,
h. counting unique values of data, format of each column, converting variable data type (e.g. from long to short, vice versa)

2. Perform the following operations using Python on the data sets:

a. Compute and display summary statistics for each feature available in the dataset. (e.g. minimum value, maximum value, mean, range, standard deviation, variance and percentiles
b. Illustrate the feature distributions using histogram.
c. Data cleaning, Data integration, Data transformation, Data model building (e.g. Classification)

3. Visualize the data using Python by plotting the graphs for assignment no. 1 and 2. Consider a suitable data set. Use Scatter plot, Bar plot, Box plot, Pie chart, Line Chart.

4. Use a multifeature dataset Wine dataset. Apply normalization and Principal Component Analysis. Plot the first two principal components and interpret dimensionality reduction.UCI Machine Learning Repository : https://archive.ics.uci.edu/datasets/

5. Predict the price of the Uber ride from a given pickup point to the agreed drop-off location. Perform following tasks: 1. Pre-process the dataset. 2. Identify outliers. 3. Check the correlation. 4. Implement linear regression and ridge, Lasso regression models. 5. Evaluate the models and compare their respective scores like R2, RMSE, etc. Dataset link: https://www.kaggle.com/datasets/yasserh/uber-fares-dataset

6. Use the diabetes data set from UCI and Pima Indians Diabetes data set for performing the following: a. Univariate analysis: Frequency, Mean, Median, Mode, Variance, Standard Deviation, Skewness and Kurtosis b. Bivariate analysis: Linear and logistic regression modeling c. Multiple Regression analysis d. Also compare the results of the above analysis for the two data sets. Dataset link: https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

7. A bank wants to predict whether a customer will default on credit card payment based on age, income, education, and payment history.

a. Preprocess data (cleaning, encoding, scaling, train–test split).
b. Build SVM models using: Linear, Polynomial, RBF kernels.
c. Build KNN models with k = 3, 5, 7.
d. Compare model accuracy and confusion matrices.

Dataset: Default of Credit Card Clients : https://www.kaggle.com/datasets/uciml/default-of-credit-card-clients-dataset

8. Apply appropriate ML algorithm on a dataset collected in a cosmetics shop showing details of customers to predict customer response for special offer. Create confusion matrix based on above data and find

a. Accuracy

b. Precision
c. Recall
d. F-1 score

9. Write a program to do following:

This dataset gives the data of Income and money spent by the customers visiting a shopping mall. The data set contains Customer ID, Gender, Age, Annual Income, Spending Score. Therefore, as a mall owner you need to find the group of people who are the profitable customers for the mall owner. Apply at least two clustering algorithms (based on Spending Score) to find the group of customers.

a. Apply Data pre-processing
b. Perform data-preparation (Train-Test Split)
c. Apply Machine Learning Algorithm
d. Evaluate Model.

Data Set: https://www.kaggle.com/shwetabh123/mall-customers

10. Assignment on Regression technique.

Download temperature data from the link below.

https://www.kaggle.com/venky73/temperaturesof-india?select=temperatures.csv

This data consists of temperatures of INDIA averaging the temperatures of all places month wise. Temperatures values are recorded in CELSIUS

a. Apply Linear Regression using a suitable library function and predict the Month-wise temperature.
b. Assess the performance of regression models using MSE, MAE and R-Square
c. metrics
d. Visualize a simple regression model.

11. Assignment on Classification technique. Every year many students give the GRE exam to get admission in foreign Universities. The data set contains GRE Scores (out of 340), TOEFL Scores (out of 120), University Rating (out of 5), Statement of Purpose strength (out of 5), Letter of Recommendation strength (out of 5), Undergraduate GPA (out of 10), Research Experience (0=no, 1=yes), Admitted (0=no, 1=yes). Admitted is the target variable.

Data Set: https://www.kaggle.com/mohansacharya/graduate-admissions

The counsellor of the firm is supposed check whether the student will get an admission or not based on his/her GRE score and Academic Score. So to help the counsellor to make appropriate decisions, build a machine learning model classifier using a Decision tree to predict whether a student will get admission or not. Apply Data pre-processing (Label Encoding, Data Transformation….) techniques if necessary.

a. Perform data-preparation (Train-Test Split)
b. Apply Machine Learning Algorithm
c. Evaluate Model.

12.A telecom company wants to predict whether a customer will churn based on demographics, account info, and usage patterns.

a. Preprocess data (missing values, encoding, scaling, train-test split)

b. Train a Random Forest classifier and tune hyperparameters
c. Evaluate model: Confusion Matrix, Accuracy, Precision, Recall, F1-score
d. Analyse feature importance

Telco Customer Churn (Kaggle): https://www.kaggle.com/datasets/blastchar/telco-customer-churn

13. Implement K-Means clustering on Iris.csv dataset. Determine the number of clusters using the elbow method. Dataset Link: https://www.kaggle.com/datasets/uciml/iris

14. A retail company wants to group customers based on Annual Income and Spending Score for targeted marketing. Dataset: Mall Customer Segmentation (Kaggle): https://www.kaggle.com/datasets/shwetabh123/mall-customers

a. Preprocess data (missing values, encoding, scaling)
b. Apply Hierarchical Clustering and plot dendrography
c. Visualize clusters (Income vs Spending Score
d. Interpret and describe each customer segment

14) Mini project is to be performed in a group of 3 to 4 students.

Develop a mini project in a group using different predictive models' techniques to solve any real-life problem. Use different frameworks of ML Frameworks such as Tensorflow, PyTorch, Scikit-learn – Keras.

List of Course Project areas:

1. 1. Explainable AI-Based Credit Risk Assessment Using SHAP and LIME
2. Hybrid Ensemble Model for Traffic Accident Severity Prediction
3. Early Detection of Alzheimer's Disease Using ML on MRI-Derived Features
4. Imbalanced Data Handling for Fraud Detection Using Advanced Sampling Techniques
5. Time-Series Forecasting of Energy Consumption Using LSTM and ML Models
6. Multi-Class Skin Disease Classification Using Machine Learning and Feature Engineering
7. Anomaly Detection in Network Traffic Using Unsupervised Learning Techniques
8. Comparative Analysis of Feature Selection Techniques for High-Dimensional Datasets
9. Demand Forecasting System Using Machine Learning with Seasonal Decomposition
10. Fake News Detection Using Hybrid NLP and Machine Learning Models
11. Predictive Maintenance of Industrial Equipment Using Machine Learning
12. Emotion Recognition from Speech Using Machine Learning Techniques
13. Stock Market Volatility Prediction Using Machine Learning and Technical Indicators
14. Medical Diagnosis System Using Ensemble Learning Techniques
15. Personalized Recommendation System Using Hybrid Collaborative and Content-Based Filtering
16. Federated Learning Framework for Privacy-Preserving Medical Data Classification
17. Explainable Intrusion Detection System Using Machine Learning
18. Multi-Step Time Series Forecasting Using Hybrid ARIMA–LSTM Models
19. Bias and Fairness Analysis in Machine Learning Classification Models
20. Graph-Based Recommendation System Using Machine Learning
21. Automated Resume Skill Extraction and Matching Using Machine Learning
22. Predictive Analysis of Air Quality Index Using Machine Learning Models

23. Semi-Supervised Learning for Fake Review Detection
24. Machine Learning-Based Early Warning System for Financial Risk Prediction
25. Feature Optimization and Model Compression for Efficient ML Deployment

CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '-' = no correlation.

CO	Program Outcome (PO)											PSO		
CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	**3**	*2*	–	–	1	–	–	–	–	–	1	3	3	
CO2	2	**3**	2	2	**3**	–	–	–	–	–	1	3	3	
CO3	2	**3**	2	**3**	2	–	–	–	–	–	1	3	3	
CO4	**3**	**3**	**3**	2	**3**	–	–	–	–	1	1	3	2	
CO5	**3**	**3**	**3**	2	**3**	1	1	1	1	1	1	3	3	
CO6	2	**3**	2	**3**	**3**	2	1	1	1	–	**2**	3	3	
Avg	2.8	2.6	2.0					2.0	2.0	2.0	2.0	3.0	2.8	

CO – WK Mapping (Knowledge & Attitude Profiles) Sample Table

'✓' indicates the knowledge profile drawn upon by the CO.

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1	✓	✓	✓	✓			✓	✓	✓
CO2		✓	✓	✓	✓	✓	✓		
CO3		✓	✓	✓	✓	✓	✓	✓	
CO4		✓	✓	✓	✓	✓	✓	✓	
CO5		✓	✓	✓	✓	✓	✓	✓	
CO6		✓	✓	✓	✓	✓	✓	✓	

CO – WP Mapping (Complex Engineering Problems) Sample Table

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO1	✓		✓	✓			
CO2	✓	✓	✓	✓	✓		✓
CO3	✓	✓	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓	✓	✓
CO5	✓	✓	✓	✓	✓	✓	✓
CO6	✓	✓	✓	✓	✓	✓	✓

CO – EA Mapping (Complex Engineering Activities)

'✓' indicates the activity attribute addressed by the CO

CO	EA1	EA2	EA3	EA4	EA5
CO1	✓				✓
CO2	✓	✓	✓		✓
CO3	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓
CO5	✓	✓	✓	✓	✓
CO6	✓	✓	✓	✓	✓

CO – SDG Mapping (Sustainable Development Goals) Sample Table

CO	SDG(s) Addressed	Justification / Linkage
CO1	SDG4, SDG8, SDG9	Builds foundational knowledge and digital skills that support quality education, innovation, and employability.
CO2	SDG4, SDG8, SDG9, SDG10, SDG12	Enables efficient data management and responsible use of computational resources while preparing datasets for intelligent applications.
CO3	SDG4, SDG9, SDG11, SDG12, SDG13, SDG16	Promotes reliable model evaluation for applications in smart cities, environmental monitoring, healthcare, governance, and sustainable decision-making.
CO4	SDG8, SDG9, SDG11, SDG12, SDG13, SDG17	Supports predictive analytics for industrial optimization, resource management, environmental sustainability, and collaborative innovation.

CO	SDG(s) Addressed	Justification / Linkage
CO5	SDG8, SDG9, SDG10, SDG11, SDG12, SDG13, SDG16, SDG17	Develops AI solutions for healthcare, finance, agriculture, governance, and inclusive digital transformation while emphasizing ethical AI.
CO6	SDG9, SDG10, SDG11, SDG12, SDG13, SDG16, SDG17	Uses clustering techniques for smart infrastructure, climate analytics, public services, policy planning, and interdisciplinary collaboration.

FF No.: 654-A

CI2019: Data Communication and Networking

Teaching Scheme	**Examination Scheme**
Credits : 3	CP: 20 Marks
Lectures : 2 Hrs/week	GD/PPT: 30 Marks
Practical : 1 Hrs/week	LAB : 10 Marks
Tutorial : NA	ESE (W/O): 40 Marks

Prerequisites:	
●	Digital Electronics
Course Objectives:	
CO1	Apply fundamental concepts of data communication to identify different types of signals, transmission impairments, and performance metrics
CO2	Differentiate between guided and unguided transmission media. Select appropriate communication channels, networking devices, and addressing schemes for computer networks.
CO3	Describe and analyze layered network architectures (OSI & TCP/IP).
CO4	*Implement* framing techniques and *evaluate* error detection and correction mechanisms for reliable data communication.
CO5	Analyze network-layer functions, addressing mechanisms, and routing techniques.
CO6	Implement transport layer services and analyze application layer protocols for reliable end-user communication
Course Outcomes:	
	After completion of the course, student will be able to

1.	Reason mathematically about elementary discrete structures (such as functions, relations, sets, graphs, and trees) used in computer algorithms and systems
2.	Express mathematical properties via the formal language of propositional and predicate logic
3.	Develop recurrence relations for a wide variety of combinatorial problems
4.	Demonstrate use of advanced combinatorial techniques
5.	Describe elementary concepts in modular arithmetic and their applications
6.	Exhibit understanding of basic graph theory, tree and its applications

Section1:	**Topics/Contents**
Unit 1:	Introduction to Data Communications (5 Hrs)
Process of Data Communication and its components: Transmitter, Receiver, Medium, Message, Protocol. Analog and Digital: Analog and Digital Data, Analog and Digital Signals, Periodic and Non-periodic Signal. Transmission Impairment: Attenuation, Distortion, Noise Data Rate Limits: Noiseless Channel: Nyquist Bit Rate, Noisy Channel: Shannon Capacity, Using Both Limits Performance: Bandwidth, Throughput, Latency (delay), Bandwidth-delay Product, Jitter.Modes of Communication (Simplex, Half Duplex, Full Duplex).	
Unit 2:	Transmission Media and Addressing (5 Hrs)
Guided Media: Twisted-Pair, Coaxial and Fiber-Optic Cable Unguided Media: Radio Waves, Microwaves, Infrared Connecting devices: Hubs, Repeaters, Active hubs, Bridges, Switches, Routers, Gateways LAN, MAN, WAN, Topologies- star, mesh, bus, hybrid etc. Addressing: Physical &logical Addresses, Port Addresses, Specific Addresses	
Unit 3:	Layered Model and Topologies (3 Hrs)
The OSI Model: Layered Architecture, peer-to-peer Processes, Encapsulation of Layers in the OSI Model, TCP/IP Protocol Suite	
Section2:	Topics/Contents
Unit 4:	Data Link Layer (6 Hrs)
Design Issues: Services to Network Layer, Framing. Types of errors: Single bit error and burst error, Redundancy, detection versus correction, forward error correction versus retransmission strategies: Error Detection and correction, Parity Bits, Hamming Codes and CRC. Flow Control Protocols: Unrestricted Simplex, Stop and Wait, Go back to N, Selective repeat Point-to-point Protocol (PPP) ,Sliding Window Protocol. MAC Sub layer: Multiple Access Protocols: Pure and Slotted ALOHA, CSMA, WDMA, CSMA/CD, CSMA/CA	
Unit 5:	Network Layer (5 Hrs)
Introduction: Functions of Network layer. Switching Techniques: Circuit switching, Message Switching, Packet Switching. IP Protocol: Classes of IP (Network addressing),	

IPv4 , IPv6, Network Address Translation, Sub-netting, CIDR. Network layer Protocols: ARP, RARP, ICMP, IGMP.Network Routing and Algorithms: Shortest Path First, Distance Vector Routing	
Unit 6:	Transport and Application Layer (6 Hrs)
Process to Process Delivery, Services, Socket Programming. Elements of Transport Layer Protocols: Addressing, Connection establishment, Connection release, Flow control and buffering. Transport Layer Protocols: TCP and UDP.Application Layer- HTTP, DNS, Email: SMTP, MIME, POP3, Webmail, FTP, TELNET, DHCP,Application layer protocols interaction with end-user applications.	
Reference Books: (As per IEEE format)	
1	Andrew S. Tenenbaum ,"Computer Networks", Pearson
2	Behrouz A. Forouzan," Data Communications and Networking"
3	Douglas E. Comer, "Computer Networks and Internets", Pearson Education
4	Godbole Achyut, "Data Communication and Networks", Tata McGraw Hill Publication.

List of Assignments:

1. Network Representation on Network Simulator tool(packet Tracer)

2. Study of DCE, DTE, straight and cross cables

3.Identify MAC and IP addresses (physically as well as using packet tracer tool).

4.Demonstrate the different types of topologies and types of transmission media by using a packet tracer tool.

5. Test with basic network tools. Read the main pages of ifconfig, ping, traceroute, arp, dig and nslookup. Write the solution of the following problems into the report and also paste the screenshots of Terminal into the report.

(a) Find the IP address of your machine using 'ifconfig' command.

(b) Use 'ping ' command and find out
(i.) the average RTT(round trip time).
(ii.) the %packet loss.
(iii.) size of packet that is sent to www.vit.edu in server.
(iv.) size of packet that is received by your machine.

(c) Use 'dig www.vit.edu' command and find out i. the ip address of vit.edu ii. the ip addresses of local DNS servers of your institute

(d) Use 'traceroute www.vit.edu' and find out
i. number of hops in between your machine and vit.edu server.
ii. the ip address of your network gateway of your subnet.

(e) Use 'arp -a' command to find out the MAC address of the device that is performing as your network gateway.

6. Design and configure static and dynamic routing in a simulated network environment using packet
Tracer

7. Implement the data link layer framing methods such as character, character-stuffing and bit stuffing.

8. Write a program to compute CRC code for the polynomials.

9. Develop a simple data link layer that performs the flow control using the sliding window protocol, and loss recovery using the Go-Back-N mechanism.

10. Implement Dijsktra's algorithm to compute the shortest path through a network.

11. Implement distance vector routing algorithm for obtaining routing tables at each node.

12. Wireshark
i. Packet Capture Using Wire shark
ii. Starting Wire shark
iii. Viewing Captured Traffic
iv. Analysis and Statistics & Filters.

13. Write the client server programs using TCP Berkeley socket primitives for wired /wireless network for following
a. to say Hello to Each other
b. File transfer

14. Write the client server programs using UDP Berkeley socket primitives for wired /wireless network for following
a. to say Hello to Each other
b. Calculator (Trigonometry)

NPTEL / MOOC course:

"Data Communication", IIT Kharagpur Prof. Ajit Pal
https://nptel.ac.in/courses/106105082

CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '-' = no correlation.

CO	Program Outcome (PO)											PSO		
CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	1	–	–	–	–	–	1	3	1
CO2	2	3	2	-	-	3	–	–	–	–	–	1	3	2
CO3	3	2	-	-	-	2	–	–	–	–	–	1	3	2
CO4	2	3	2	2	2	3	–	–	–	–	–	1	2	3
CO5	2	3	2	2	2	3	–	–	–	–	–	1	3	2
CO6	2	3	3	3	2	3	–	–	–	1	–	1	2	3
Avg														

CO – WK Mapping (Knowledge & Attitude Profiles) Sample Table

'✓' indicates the knowledge profile drawn upon by the CO.

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1		✓	✓		✓				
CO2		✓	✓	✓	✓	✓			
CO3		✓		✓	✓				
CO4		✓	✓	✓	✓			✓	
CO5		✓	✓	✓	✓	✓	✓	✓	
CO6		✓	✓	✓	✓	✓		✓	

CO – WP Mapping (Complex Engineering Problems) Sample Table

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO1	✓			✓	✓		✓
CO2	✓	✓	✓	✓	✓	✓	✓
CO3	✓	✓	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓	✓	✓
CO5	✓	✓	✓	✓	✓	✓	✓
CO6	✓	✓	✓	✓	✓	✓	✓

CO – EA Mapping (Complex Engineering Activities)

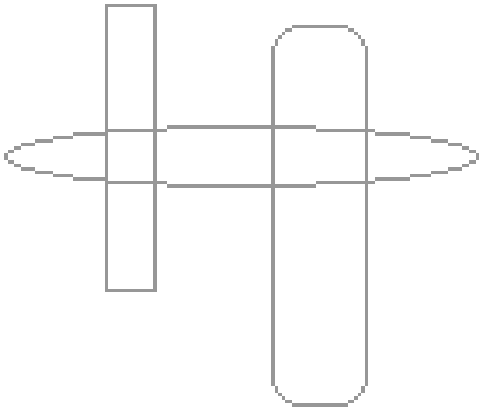
'✓' indicates the activity attribute addressed by the CO

CO	EA1	EA2	EA3	EA4	EA5
CO1	✓	✓			✓
CO2	✓	✓	✓	✓	✓
CO3	✓	✓	✓	✓	✓
CO4	✓	✓	✓	✓	✓
CO5	✓	✓	✓	✓	✓
CO6	✓	✓	✓	✓	✓

CO – SDG Mapping (Sustainable Development Goals) Sample Table

CO	SDG(s) Addressed	Justification / Linkage
CO1	4	Provides foundational knowledge in communication technologies essential for engineering education and lifelong learning
CO2	9	Supports the design and deployment of reliable communication infrastructure and promotes technological innovation.
CO3	9	Enhances understanding of standardized communication architectures that form the backbone of modern digital infrastructure.

CO	SDG(s) Addressed	Justification / Linkage
CO4	9	Contributes to the development of robust, efficient, and reliable communication systems for industrial and societal applications.
CO5	11	Enables efficient network connectivity and smart communication systems essential for sustainable urban infrastructure and digital services.
CO6	9	Facilitates secure and reliable communication services supporting digital innovation, online learning, collaboration, and information exchange.



FF No.: 654

CI2020: Full Stack Development

Teaching Scheme	Examination Scheme
Credits : 04	CA: 10 Marks
Lectures : 2 Hrs/week	CP: 30 Marks
Practical : 2 Hrs/week	CVV: 20 Marks
Tutorial : NA	TH: 40 Marks

Prerequisites:	
•	**Object-Oriented Programming Concepts, Fundamentals of Web Technologies, Database Management Systems**
Course Objectives:	
CO1	To introduce basic web concepts, HTML5 structure, CSS styling, and responsive layouts.
CO2	To develop the ability to write dynamic and interactive web applications using JavaScript, ES6 features, and DOM manipulation techniques.
CO3	To build interactive and responsive web applications using jQuery and Bootstrap.
CO4	To equip students with backend development skills using Node.js/Express.js, including API creation and database integration.
CO5	To introduce modern component-based frameworks such as Angular/React and TypeScript to build scalable and modular front-end applications.
CO6	To enable students to develop dynamic, component-based web applications using React.js.
Course Outcomes:	
	After completion of the course, student will be able to
1.	To implement structured and responsive web pages using HTML5, CSS, and layout techniques.
2.	To use JavaScript, ES6 features, and DOM manipulation techniques for constructing interactive and dynamic web content.
3.	To employ jQuery and Bootstrap to design responsive and interactive user interfaces.
4.	To analyze backend processes by building APIs and integrating databases using Node.js and Express.js.
5.	To construct modular, component-driven applications using Angular or React, along with TypeScript constructs.
6.	To implement dynamic, component-based web applications using React.js.

Course Relevance:

The **Full Stack Development** course equips students with the knowledge and practical skills required to design, develop, test, and deploy complete web applications. It covers both front-end and back-end technologies, database management, server-side programming, and application deployment, enabling students to build end-to-end software solutions. The course emphasizes industry-relevant tools, frameworks, and best practices, preparing students for roles such as Full Stack Developer, Web Developer, Software Engineer, and Application Developer. It also enhances problem-solving, collaborative development, and software engineering skills required in modern IT industries.

Section1:	**Topics/Contents**

Unit 1: (Web Essentials) (05 Hours)

Introduction to the Web: Evolution of the Web: Web 1.0, Web 2.0, Web 3.0, Internet basics: Clients, servers, IP, DNS, hosting, browser architecture & rendering process, HTTP & HTTPS: Request/Response, Methods, Status Codes, URL Structure, and Query Parameters.
Developer Tools: Inspecting elements, understanding the console & network tab, working with sources & debugging.
Introduction to Web Project Structure: Folder organization: assets, scripts, styles, linking external files.
HTML5 Basics: HTML5 elements: headings, colors, fonts, links, frames, lists, tables, images, Forms & **default validation,** Block vs inline elements, Semantic elements (header, nav, section, footer), Difference between HTML & HTML5, HTML5 media: audio, video.
CSS Basics: Why CSS? Types of CSS, Applying CSS: inline, internal, external, properties, classes, nested (child) selectors, Colors, text, background, border, margin, padding, box model.
Layout + Animation: Positioning: flex, grid, inline, block, CSS animation & transition.

Unit 2: (JavaScript Development) (05 Hours)

JavaScript Fundamentals: Syntax, variables, datatypes, operators, conditionals, loops, functions, objects, arrays, and built-in objects (Math, Date, and String)
ES6 Modern JavaScript Features: let, const, arrow functions, spread/rest operator, for/of, map & set, classes, promises, symbols, default parameters, String methods: includes(), startsWith(), endsWith()
DOM Manipulation: Document Object Model, selecting and modifying elements, Event handling, modifying CSS dynamically, DOM tree traversing.

Unit 3: (JQuery and Bootstrap) (05 Hours)

jQuery: Introduction to jQuery: Need for jQuery, jQuery Syntax: Selectors, filters, DOM traversal, DOM manipulation: adding/removing elements, modifying CSS & attributes, Event Handling: Click, hover, keyboard, and form events. Effects & Animations: hide/show, fade, slide.

Bootstrap (Industry Requirement): Why Bootstrap? CSS vs. Bootstrap, Bootstrap grid system, responsive design, Bootstrap components: button, table, lists, cards, cross-platform UI using Bootstrap

Section2:	**Topics/Contents**

Unit 4: (Backend Development with Node.js or Express.js) (06 Hours)

Node.js Core: Introduction to Node.js, environment setup, Node.js events, Node.js functions, Built-in modules (fs, http, path, os, events)
Express.js: Introduction to Express, Routing, Middleware, Creating APIs
Database Integration: Express with MongoDB or Express with MySQL/SQLite
Applications: Front-end + back-end integration, Building a chat application, Introduction to task managers (npm scripts, nodemon, and PM2) and unit testing basics.

Unit 5: (Frontend Framework: Angular) (05 Hours)

Angular: Angular Architecture, Project Structure, Lifecycle, Modules & Components, Data Binding, Directives & Pipes, Services & Dependency Injection, Routers, Angular Forms
TypeScript: Type annotations & inference, basic types, interfaces, classes, inheritance, enums, generics, Type guards, unions, intersections, decorators, modules & namespaces, advanced types, declaration files, and tooling.

Unit 6: (Frontend Framework: React.js) (04 Hours)

React.js: Introduction to React, React Components, JSX, Render Function, Component API, Component Lifecycle, State, Props, Passing Data from Parent to Child, Passing Data from Child to Parent, Component Styling, Refs, Keys, React Router.

Text Books: ***(As per IEEE format)***

1	*Achyut Godbole, Atul Kahate; Web Technologies: TCP/IP to Internet Application Architectures; 1st Edition, McGraw Hill Education.*
2	*Chris Northwood; The Full Stack Developer; 2nd Edition, Apress, 2023*

Reference Books: ***(As per IEEE format)***

1	*Ethan Brown; Web Development with Node and Express; 2nd Edition, O'Reilly Media, 2022*
2	*Alex Banks, Eve Porcello; Learning React; 3rd Edition, O'Reilly Media.*
3	*Brad Dayley, Brendan Dayley, Caleb Dayley, "Node.js, MongoDB and Angular Web Development", 2nd Edition, Addison-Wesley*
4	*Ethan Brown; Learning JavaScript: JavaScript Essentials for Modern Application Development; 4th Edition, O'Reilly Media.*
5	*Bear Bibeault, Yehuda Katz, Aurelio De Rosa; jQuery in Action; 3rd Edition, Manning Publications*
6	*Jeremy McPeak& Paul Wilton," Beginning JavaScript", Wrox Publication, ISBN-13: 978- 0470525937*
7	*Maximilian Schwarzmüller; React Key Concepts; 2nd Edition, Packt Publishing*

List of Practical's:

1. Design a multipage personal portfolio website using HTML5 semantic elements and CSS. The website should include headings, images, tables/lists, navigation links, forms with default validation, and proper use of classes, colors, margins, padding, and box model.
2. Build a multimedia webpage that includes HTML5 audio and video elements, styled using external CSS. Add captions, controls, and create a clean layout using padding, borders, and background properties.
3. Write JavaScript programs demonstrating variables, datatypes, operators, conditionals, loops, functions, arrays, and objects.
4. Create a webpage and use DOM manipulation to change text, style elements, and handle button click events.
5. Build a Random Quote Generator using JavaScript that displays a new quote each time a button is clicked.
6. Create an interactive webpage using jQuery that performs DOM manipulation, handles click/hover events, and applies effects such as hide/show, fade, and slide on selected elements.
7. Build a fully responsive webpage using Bootstrap with a grid layout and components (button, table, list, card).
8. Create a Node.js server using built-in modules (http, fs, path) that serves static HTML pages and demonstrates event handling and file operations.
9. Build a RESTful API using Express.js integrated with MongoDB/MYSQL/SQLite, implement routing and middleware, and connect it to a simple front-end to perform CRUD operations.
10. Create a simple Angular project with two components and display data using interpolation and property/event binding.
11. Create an Angular service to share data between components and set up routing with at least two navigable pages.
12. Create a React application with multiple components, demonstrating state and props for dynamic content rendering
13. Implement event handling in a React app, such as button clicks, form submissions, and input change events.
14. Build a multi-page React application using React Router and apply CSS or inline styling to components for a polished UI.

List of Project areas:

1. **AI-Driven Adaptive Learning Platform:** Design and develop a full-stack platform that dynamically personalizes learning content using user behavior analytics and performance metrics.
2. **Scalable Microservices-Based E-Commerce System:** Build an e-commerce application using microservices architecture with service discovery, API gateway, distributed database, and fault tolerance.
3. **Real-Time Collaborative Code Editor:** Develop a browser-based collaborative coding platform with real-time synchronization, conflict resolution, version history, and user authentication.

4. **Blockchain-Based Secure Document Management System:** Design a decentralized web application for document storage and verification using blockchain to ensure immutability and traceability.
5. **Smart Traffic Monitoring and Prediction System:** Build a full-stack system that collects real-time traffic data, applies predictive analytics, and visualizes congestion patterns on interactive maps.
6. **Multi-Tenant SaaS Project Management Tool:** Develop a secure, multi-tenant project management application with tenant isolation, role-based access control, billing, and usage analytics.
7. **IoT-Enabled Smart Energy Management Platform:** Create a full-stack solution that integrates IoT sensor data with cloud services to monitor, analyze, and optimize energy consumption in real time.
8. **High-Availability Online Examination System:** Design a fault-tolerant exam platform supporting thousands of concurrent users with load balancing, auto-scaling, and anti-cheating mechanisms.
9. **Real-Time Financial Trading Dashboard:** Build a full-stack web application that streams live market data, performs analytics, and visualizes trends with low-latency updates.
10. **Secure Healthcare Data Exchange System:** Develop a full-stack platform for interoperable healthcare data exchange using encryption, audit trails, and compliance-aware access controls.
11. **Intelligent Supply Chain Optimization System:** Design a web-based system that integrates demand forecasting, inventory optimization, and logistics tracking using data analytics.
12. **Federated Authentication and Authorization System:** Build a full-stack identity management system supporting SSO, OAuth 2.0, JWT, multi-factor authentication, and fine-grained access control.

CO-PO Mapping*Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '—' = no correlation.*

CO		Program Outcomes (PO)											PSO		
CO/PO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	2		2				2	2	2	2	3	2	2
CO2	3	2	2	1	2				1	1	1	1	3	2	2
CO3	3	2	2	1	2				2	2	2	2	3	2	2
CO4	3	3	3	2	3			1	2	2	2	2	3	2	2
CO5	3	2	2	2	3				2	2	2	2	3	2	2
CO6	3	2	2	2	2	1	1		2	2	2	2	3	2	2
Average	3.00	2.17	2.17	1.60	2.33	1.00	1.00	1.00	1.83	1.83	1.83	1.83	3.00	2.00	2.00

CO – WK Mapping (Knowledge & Attitude Profiles)

'✓' indicates the knowledge profile drawn upon by the CO.

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1	✓	–	✓	✓	–	✓	–	–	–
CO2	✓	✓	✓	✓	–	✓	–	✓	–
CO3	✓	–	✓	✓	✓	✓	–	–	–
CO4	✓	✓	✓	✓	✓	✓	–	✓	–
CO5	✓	✓	✓	✓	✓	✓	–	✓	–
CO6	✓	✓	✓	✓	✓	✓	–	✓	–

CO – WP Mapping (Complex Engineering Problems)

'✓' indicates the problem-solving attribute addressed by the CO.

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO1	✓	–	–	–	–	–	–
CO2	✓	–	✓	–	–	–	–
CO3	✓	–	✓	–	–	–	✓
CO4	✓	✓	✓	✓	–	✓	✓
CO5	✓	✓	✓	✓	✓	✓	✓
CO6	✓	–	✓	✓	–	✓	✓

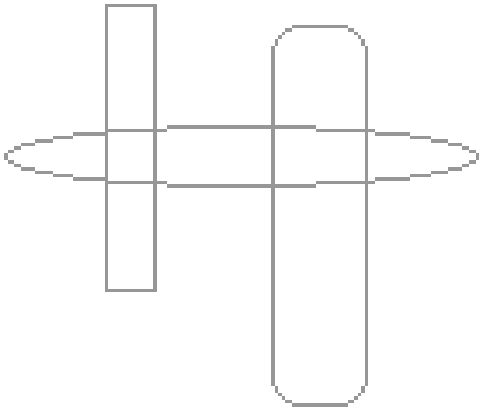
'✓' indicates the activity attribute addressed by the CO.

CO	EA1	EA2	EA3	EA4	EA5
CO1	✓	–	–	–	✓
CO2	✓	✓	✓	–	✓
CO3	✓	✓	✓	–	✓
CO4	✓	✓	✓	✓	✓
CO5	✓	✓	✓	✓	✓
CO6	✓	✓	✓	✓	✓

CO – SDG Mapping (Sustainable Development Goals)

CO	SDG(s) Addressed	Justification / Linkage

CO1	**SDG 4, SDG 8, SDG 9**	Develops foundational skills in responsive web design, enabling learners to build modern web applications that support digital education, employability, and technological infrastructure.
CO2	**SDG 4, SDG 8, SDG 9**	Enhances programming and problem-solving skills through JavaScript and DOM manipulation, preparing students for software development careers and digital innovation.
CO3	**SDG 4, SDG 9, SDG 10**	Promotes the development of responsive and accessible user interfaces using jQuery and Bootstrap, contributing to inclusive digital services and innovation.
CO4	**SDG 8, SDG 9, SDG 16**	Enables the development of secure backend systems, APIs, and database-driven applications that support reliable digital infrastructure and trustworthy online services.
CO5	**SDG 4, SDG 8, SDG 9, SDG 17**	Develops industry-relevant expertise in modern frontend frameworks, encouraging collaborative software development and innovation using contemporary technologies.
CO6	**SDG 4, SDG 8, SDG 9, SDG 17**	Facilitates the implementation of scalable React.js applications, fostering digital innovation, professional competencies, and teamwork in software development projects.



FF No.: 654-A

MM0902: Industrial Automation 5.0

Teaching Scheme	Examination Scheme
Credits: 3	MSE:35 Marks
Lectures: 2 Hrs/week	ESE: 35 Marks
Tutorials : 1 Hrs/week	Tutorial: 30 Marks

Prerequisites:	
•	**Basic knowledge of electronics and sensors, Fundamentals of computer networks, Basic programming understanding (C/Python)**
Course Objectives:	
CO1	To introduce concepts of Industrial Automation and Industry 5.0.
CO2	To explain IoT architecture, sensors, actuators, and communication technologies.
CO3	To understand industrial protocols, cloud monitoring, and data handling.
CO4	To explore industrial automation systems such as SCADA, predictive maintenance, and digital twins.
CO5	To understand security, privacy, and safety in IIoT environments.
CO6	To analyse real-world case studies of Industry 5.0 applications.
Course Outcomes:	
	After completion of the course, student will be able to
1.	Explain the concepts of Industrial Automation and Industry 5.0.
2.	Describe IoT fundamentals and related architectures used in industrial environments.
3.	Identify and compare industrial sensors, actuators, and communication modules.
4.	Apply industrial IoT protocols and cloud platforms for data monitoring.
5.	Explain automation systems such as SCADA, predictive maintenance, and hardware selection.
6.	Analyze security, safety, and case studies in Industrial IoT applications.

Course Relevance:

The **Industry Automation 5.0** course equips students with the knowledge and skills required to understand modern industrial automation systems and emerging Industry 5.0 technologies. The course introduces Industrial Internet of Things (IIoT), smart sensors, actuators, industrial communication protocols, cloud-based monitoring, SCADA systems, predictive maintenance, digital twins, and cybersecurity in industrial environments. It emphasizes human-centric, sustainable, and resilient manufacturing by integrating automation with Artificial Intelligence (AI), data analytics, edge computing, and cyber-physical systems. Through practical case studies and industrial applications, students develop the ability to design, monitor, and analyze intelligent

automation solutions used in manufacturing, healthcare, logistics, energy, and smart infrastructure. The course prepares graduates for next-generation industrial ecosystems by bridging theoretical concepts with real-world industrial practices, thereby enhancing employability in Industry 5.0-enabled organizations.

Topics/Contents
Section1:
UNIT 1: Introduction to Industrial Automation & Industry 4.0 (5 Hr)
Industrial Automation: Meaning, Types, Need, Evolution: Industry 1.0 → 5.0, Features of Industry 4.0: Smart Sensors, Connectivity, Automation, Big Data, Role of IoT, AI, Cyber-Physical Systems, Real industrial examples (manufacturing, robotics, energy)
UNIT 2: IoT Fundamentals for Industrial Automation (5 Hr)
IoT Definition, Characteristics, Architecture, IoT layers: Device → Network → Cloud, M2M (Machine-to-Machine) communication, IoT Frameworks & platforms, Identifiers in IoT (UID, EPC, IP Addressing) Overview of simple hardware: Arduino, ESP32, Basics of sensor interfacing, Introduction to Raspberry Pi board: architecture, features, GPIO pins, operating system support, networking capability, role as edge device / IoT gateway in industrial automation.
UNIT 3: Industrial Sensors, Actuators & Communication (4 Hr)
Sensors: Temperature, Proximity, Vibration, Pressure, Load, Angle, Speed, Non-contact: IR, Ultrasonic, Laser sensors, Actuators: Motors, Relays, Solenoids, Industrial Mechanisms, Communication Technologies: Wi-Fi, Bluetooth, ZigBee (Basic), MQTT Publish/Subscribe model, Edge Controllers & Industrial Gateways
Section2: Topics/Contents
UNIT 4: IoT Protocols & Cloud in Industry 5.0 (5 Hr)
IoT Protocols: IPv6 basics, lightweight protocols – **6LoWPAN, RPL, REST, AMQP, MQTT, CoAP**, simple overview of text/image transfer protocols. Cloud & IIoT Platforms: **IIoT basics,** cloud monitoring using **ThingSpeak, Blynk, Firebase** – dashboards, data logs, alerts, remote device control. **Arduino Cloud and Node-RED:** Device connectivity, cloud dashboards, flow-

based programming for industrial IoT applications**. Cloud Service Providers:** Study of **AWS, Azure, GCP** IoT platforms.
UNIT 5: Automation Systems in Industry 5.0 (5 Hr)
Smart Manufacturing Concepts, Predictive Maintenance (Basic Idea), Smart Energy Monitoring, Digital Twin (Concept Only): SCADA: Architecture, Components, Use Cases Selection of industrial computational hardware: Raspberry Pi, Jetson Nano, ESP32, Microcontrollers
UNIT 6: Security, Safety & Industrial Case Studies (4 Hr)
Security in IioT: Threats & Challenges, Authentication mechanisms, Basic encryption & password protection, Privacy regulations and policies, Safety in Industry 5.0: Safety sensors, Emergency switches, alarms. Case Studies: Industry 5.0 use cases in manufacturing, healthcare, logistics, Breakdown of real IIoT systems, Students analyze and present case studies (tutorial-based)

Text Books:

1.Alasdair Gilchrist, “Industry 4.0: The Industrial Internet of Things”, Apress.
2. Rajesh Singh, Anita Gehlot, “Internet of Things”, McGraw Hill.
3.William Buchanan, “Industrial Control Systems & SCADA Security”, CRC Press.
4.Arshdeep Bahga and Vijay Madisetti, Internet of Things: A Hands-on Approach", (Universities Press)
5.Pethuru Rajand Anupama C. Raman, "The Internet of Things: Enabling Technologies, Platforms, and Use Cases", (CRC Press)

Reference Books:

1.Jay Lee, Behrad Bagheri, Hung-An Kao, “A Cyber-Physical Systems Architecture for Industry 5.0”.
2.Adrian McEwen, Hakim Cassimally, “Designing the Internet of Things”, Wiley.
3.Salesforce IoT & IBM Industrial IoT documentation (online resources).
4.Adrian McEwen, Hakim Cassimally “Designing the Internet of Things”, Wiley
5.OvidiuVermesan & PeterFriess “Internet of Things Applications From Research and Innovation to Market Deployment”, ISBN:987-87-93102-94, River Publishers.
6.Joe Biron and Jonathan Follett, "Foundational Elements of an IoT Solution," by Joe Biron

List of Tutorials:

1. Industry 1.0 to 5.0 Evolution
2. IoT Architecture & Components
3. Smart Sensors & Connectivity Features in Industry 5.0
4. Classification of Industrial Sensors
5. Actuators & Microcontroller Interfacing Basics
6. IoT Protocol Comparison

7. Short-Range & Long-Range Communication Technologies
8. MQTT Publish/Subscribe Model
9. Industrial Sensors & Actuators Study
10. Cloud Monitoring Platforms
11. IIoT Threats & Vulnerabilities
12. Privacy & Security Policies in IIoT
13. Predictive Maintenance Techniques
14. Digital Twin & Industry 4.0 Case Studies
15. Sensor & protocol demonstration
16. Cloud dashboard activity
17. Case study analysis
18. Simple device communication tasks
19. Mini conceptual assignments

CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '-' = no correlation.

CO-PO Mapping:

	Program Outcomes (PO)												PSO		
CO/PO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**	**PSO3**
CO1	3	2	–	–	–	2	–	–	–	–	–	1	2	1	1
CO2	3	2	–	–	2	–	–	–	–	–	–	1	2	2	1
CO3	3	2	2	–	3	–	–	–	–	–	–	–	3	2	1
CO4	2	3	3	–	3	–	–	–	–	–	–	–	3	3	1
CO5	2	2	3	–	2	1	–	–	–	–	–	–	2	3	2
CO6	1	3	–	2	–	3	2	2	2	2	–	2	1	2	3
Average	**2.33**	**2.33**	**2.67**	**2.00**	**2.60**	**2.00**	**2.00**	**2.00**	**2.00**	**2.00**	**–**	**1.40**	**2.17**	**2.17**	**1.50**

CO – WK Mapping (Knowledge Profile)

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1	✓	✓	✓						
CO2	✓	✓	✓	✓					
CO3	✓	✓	✓	✓	✓				
CO4	✓	✓	✓	✓	✓	✓			
CO5	✓	✓	✓	✓	✓	✓	✓		
CO6	✓	✓	✓	✓	✓	✓	✓	✓	✓

CO – WP Mapping (Complex Engineering Problems)

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO1	✓						
CO2	✓	✓					
CO3	✓	✓	✓				
CO4	✓	✓	✓	✓			
CO5	✓	✓	✓	✓	✓		
CO6	✓	✓	✓	✓	✓	✓	✓

CO – EA Mapping (Complex Engineering Activities)

CO	EA1	EA2	EA3	EA4	EA5
CO1	✓				
CO2	✓	✓			
CO3	✓	✓	✓	S	
CO4	✓	✓	✓	✓	
CO5	✓	✓	✓	✓	✓
CO6	✓	✓	✓	✓	✓

CO – SDG Mapping (Sustainable Development Goals)

CO	SDG(s) Addressed	Justification / Linkage
CO1	4, 9	Introduces Industrial Automation and Industry 5.0 concepts, strengthening engineering knowledge and supporting quality education and industrial innovation.
CO2	4, 9	Develops understanding of IoT architecture and industrial communication technologies for smart industrial infrastructure.
CO3	8, 9	Builds competencies in industrial sensors, actuators and communication systems supporting employability and sustainable industry.
CO4	9, 11	Applies IIoT protocols and cloud platforms for intelligent monitoring and sustainable industrial systems.
CO5	9, 12	Explains SCADA, predictive maintenance and digital twins to improve efficiency and sustainable production.
CO6	9, 16	Analyzes IIoT security, safety and privacy for secure and reliable industrial automation systems.

CO Attainment Levels:

Weights for attainment levels: L1 – Easy – 0.75, L2 – Comfortable – 0.70, L3 – Medium – 0.65, L4 – Somewhat difficult – 0.60, L5 – Difficult – 0.55.

CO1 – __, CO2 – __, CO3 – __, CO4 – __, CO5 – __, CO6 – __ (to be computed from actual assessment data)

Future Courses Mapping

The knowledge and skills acquired in this course provide a strong foundation for advanced courses in Industrial Automation, Industrial Internet of Things (IIoT), Smart Manufacturing, and Intelligent Systems. The concepts learned in this course support higher-level understanding of Internet of Things (IoT), Embedded Systems, Cyber-Physical Systems, Artificial Intelligence, Machine Learning, Robotics and Automation, Industrial Robotics, Cloud Computing, Edge Computing, Digital Twin Technology, Industrial Cybersecurity, Data Analytics, Smart Manufacturing Systems, Industry 5.0 Technologies, and Industrial Control Systems. The course also prepares students to design intelligent, connected, secure, and sustainable industrial solutions using modern automation technologies.

Job Mapping

The competencies developed through this course prepare students for careers in industrial automation, smart manufacturing, and Industrial IoT ecosystems. Graduates will be equipped with the knowledge required for roles such as Industrial Automation Engineer, IIoT Engineer, Automation System Engineer, Industrial IoT Developer, SCADA Engineer, PLC Programmer, Industrial Control Engineer, Smart Manufacturing Engineer, Edge Computing Engineer, Digital Twin Engineer, Industrial Cybersecurity Engineer, Embedded Systems Engineer, Robotics Engineer, Production Automation Engineer, Industrial Data Analyst, AI for Manufacturing Engineer, Smart Factory Engineer, and Industry 5.0 Solution Architect. These roles require expertise in industrial communication, automation technologies, intelligent monitoring systems, cloud-based industrial platforms, cybersecurity, and human-centric smart manufacturing solutions.

FF No.: 654-A

HS2004: REASONING & APTITUDE DEVELOPMENT-4

Teaching Scheme	**Examination Scheme**
Credits : 1	CP: -- Marks
Lectures : 1 Hrs/week	GD/PPT/HA: -- Marks
Practical : -- Hrs/week	MSE (W/O): -- Marks
Tutorial : -- Hrs/week	ESE (W/O): 100 Marks

Prerequisites:	
●	Basic knowledge of English grammar and vocabulary.
●	Fundamental understanding of sentence construction and communication skills.
●	Basic arithmetic and mathematical concepts (Class XII level).
●	Elementary logical and analytical reasoning skills.
●	Familiarity with problem-solving techniques.
●	Basic computer literacy and programming concepts.
●	Exposure to at least one programming language (C/C++/Java/Python) is desirable.
Course Objectives:	
CO1	Develop proficiency in English grammar, vocabulary, reading comprehension, and effective written communication.
CO2	Strengthen logical, analytical, critical, and reasoning abilities to solve diverse aptitude-based problems.
CO3	Apply quantitative and mathematical concepts to solve real-world numerical and analytical problems efficiently.
CO4	Interpret, analyze, and synthesize information from textual, graphical, tabular, and numerical data for informed decision-making.
CO5	Develop computational thinking and programming aptitude through logic building, pseudocode, algorithmic problem solving, and debugging techniques.
CO6	Enhance coding proficiency, optimization skills, and problem-solving abilities to perform effectively in programming assessments, campus recruitment tests, and technical interviews.
Course Outcomes:	
	After completion of the course, student will be able to
1.	**Apply** appropriate English language skills, including grammar, vocabulary, reading comprehension, and effective communication techniques to improve verbal and written proficiency.

2.	**Analyze** logical reasoning problems, critical thinking scenarios, and data interpretation tasks to arrive at accurate and evidence-based conclusions.
3.	**Solve** quantitative aptitude problems using mathematical concepts, algebraic reasoning, probability, geometry, and analytical techniques in real-life contexts.
4.	**Develop** efficient algorithmic solutions by applying computational thinking, programming fundamentals, debugging, optimization techniques, and time complexity analysis to solve coding problems.

Section1:	**Topics/Contents**

English Language and Communication Skills: English tenses (Present, Past, Future), active and passive voice, sentence correction, parts of speech, syntax, sentence structure, subject–verb agreement, reading comprehension, skimming and scanning, inference, tone, theme, intent identification, contextual vocabulary.

Logical Reasoning and Analytical Thinking: Deductive reasoning (syllogisms, coding–decoding, statement–conclusion, data sufficiency, directional sense, logical sequencing), inductive reasoning (analogies, classification, number series, sequence completion, trend identification), abductive and critical reasoning (assumptions, arguments, cause–effect, logical evaluation, decision making), information gathering, interpretation and synthesis using graphs, charts, tables and multi-source data.

Section2:	**Topics/Contents**

Quantitative Aptitude and Mathematical Reasoning: Number systems, integers, fractions, decimals, divisibility, HCF and LCM, prime and rational numbers, algebraic expressions, linear equations and inequalities, proportional reasoning, speed–time–distance, profit and loss, percentages, averages, ages, mixtures, permutations and combinations, probability, geometry, spatial reasoning, pattern recognition and quantitative problem solving.

Programming Aptitude and Automata Skills: Logic building, pseudocode, programming fundamentals, basic coding problems, program correctness, execution efficiency, intermediate coding challenges, data structures, algorithms, debugging, optimization techniques, time complexity analysis, industry best coding practices, Automata Pro IT, Automata Pro Max and comprehensive mock assessments.

Text Books: ***(As per IEEE format)***

1	R. S. Aggarwal, *Quantitative Aptitude for Competitive Examinations*, Revised and Enlarged Edition. New Delhi, India: S. Chand and Company Ltd., 2017.
2	R. S. Aggarwal, *A Modern Approach to Verbal & Non-Verbal Reasoning*. New Delhi, India: S. Chand Publishing.
3	Dr. M. B. Lal and A. Gupta, *CSAT Logical Reasoning & Analytical Ability*. Agra, India: Upkar Prakashan.
4	G. Laakmann McDowell, *Cracking the Coding Interview: 189 Programming Questions and Solutions*, 6th ed. Palo Alto, CA, USA: CareerCup, 2015.
Reference Books: ***(As per IEEE format)***	
1	B. H. Dowden, *Logical Reasoning*. California State University, Sacramento, CA, USA, 2011.
2	*501 Challenging Logic and Reasoning Problems*, 2nd ed. New York, NY, USA: LearningExpress, LLC, 2005.
3	*Logic and Reasoning Problems*. LearningExpress Practice Series.
Link: *https://drive.google.com/drive/folders/1fXLzuofNhMFq7skonrC0TpCdaCoT7_i0?usp=sharing*	

CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '-' = no correlation.

CO	Program Outcome (PO)											PSO		
CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PSO1**	**PSO2**	**PSO3**
CO1	–	–	–	–	–	–	–	–	3	–	2			
CO2	2	2	–	–	–	–	–	–	1	–	2			
CO3	2	2	–	–	–	–	–	–	1	1	2			
CO4	2	2	3	3	3	–	1	–	1	1	3			
Avg	**2**	**2**	**3**	**3**	**3**	**–**	**1**	**–**	**1.5**	**1**	**2.25**			

CO – WK Mapping (Knowledge & Attitude Profiles)

'✓' indicates the knowledge profile drawn upon by the CO.

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1	–	✓	–	–	–	–	–	✓	✓
CO2	–	✓	–	–	–	–	–	✓	✓
CO3	–	✓	–	–	–	–	–	✓	✓
CO4	–	✓	–	–	–	–	–	✓	✓

CO – WP Mapping (Complex Engineering Problems)

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO1	–	–	–	–	–	–	–
SSCO2	✓	–	✓	✓	–	–	–
CO3	✓	–	✓	✓	–	–	–
CO4	✓	–	✓	✓	–	–	✓

CO – EA Mapping (Complex Engineering Activities)

'✓' indicates the activity attribute addressed by the CO

CO	EA1	EA2	EA3	EA4	EA5
CO1	–	–	–	–	–
CO2	–	–	✓	–	✓
CO3	–	–	✓	–	✓
CO4	✓	–	✓	–	✓

CO – SDG Mapping (Sustainable Development Goals)

CO	SDG(s) Addressed	Justification / Linkage
CO1	SDG 4	Develops proficiency in English communication, reading comprehension, and language skills, promoting inclusive and quality education and lifelong learning.
CO2	SDG 4	Enhances logical reasoning, critical thinking, and analytical decision-making skills, enabling responsible problem-solving and informed decision-making.
CO3	SDG 4, 8	Strengthens quantitative aptitude and analytical skills essential for employability, professional competence, and workplace productivity.
CO4	SDG 4, 8, 9	Develops computational thinking, programming, and algorithmic problem-solving skills required for innovation, digital transformation, and industry-ready engineering professionals.

- **SDG 4:** Quality Education
- **SDG 8:** Decent Work and Economic Growth
- **SDG 9:** Industry, Innovation and Infrastructure

CI2015: Design Thinking

Credits:.1 **Lab: 2 hours/Week**

Teaching Scheme	**Examination Scheme**
Credits : 1	CP: - Marks
Lectures : Hrs/week	GD/PPT/HA: - Marks
Practical : Hrs/week	MSE (W/O): - Marks
Tutorial : Hrs/week	ESE (Direct Evaluation): 100 Marks

Prerequisites:	
●	Problem Based Learning, Project Centric Learning
Course Objectives:	
●	To provide ecosystem for students and faculty for paper publication and patent filing
Course Outcomes:	
	After completion of the course, student will be able to
1.	Understand the importance of doing Research
2.	Interpret and distinguish different fundamental terms related to Research
3.	Apply the methodology of doing research and mode of its publication
4.	Write a Research Paper based on project work
5.	Understand Intellectual property rights
6.	Use the concepts of Ethics in Research
7	Understand the Entrepreneurship and Business Planning

Section1:	**Topics/Contents**
What is research? Importance of Paper Publication and Patents Structure of Paper Journal Publication Publication in conference Literature Review Research Paper Writing Journal Ratings and Evaluation How to rate a Journal? Intellectual property (IP) Research Ethics Entrepreneurship	
Section2:	**Topics/Contents**

Structure of The paper Journal List (Top 50 Journals) Selection of the journal Use of various online journal selection tools Plagiarism checking Improving contents of the paper Patent drafting Patent search Filing of patent Writing answers to reviewer questions Modification in manuscript Checking of publication draft	
Text Books: ***(As per IEEE format)***	
1	C. R. Kothari and G. Garg, *Research Methodology: Methods and Techniques*, 4th ed. New Delhi, India: New Age International Publishers, 2019
2	W. Booth, G. Colomb, J. Williams, J. Bizup, and W. FitzGerald, *The Craft of Research*, 5th ed. Chicago, IL, USA: The University of Chicago Press, 2024
3	R. M. Sarma, *Research Methodology*, 1st ed. New Delhi, India: PHI Learning Pvt. Ltd., 2019.
Reference Books: ***(As per IEEE format)***	
1	A. Wallwork, *English for Writing Research Papers*, 2nd ed. Cham, Switzerland: Springer, 2016.
2	A. Wallwork, *English for Research: Usage, Style, and Grammar*, New York, NY, USA: Springer, 2013
3	P. Narayanan, *Patent Law*, 4th ed. Lucknow, India: Eastern Law House, 2017.

CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '-' = no correlation.

CO	**Program Outcome (PO)**											**PSO**		
CO	**PO1**	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO 11**	**PSO1**	**PSO2**	**PSO3**
CO1	3	2	-	2	1	2	1	2	-	2	-	2	-	1
CO2	3	3	-	2	2	-	-	2	-	2	-	2	-	1
CO3	2	3	2	3	2	-	-	2	-	2	-	2	2	2
CO4	2	2	3	3	2	-	-	2	2	3	-	2	2	2
CO5	2	2	-	1	-	2	2	3	-	2	2	1	-	1
CO6	2	2	-	2	-	2	3	3	2	2	1	1	-	1
Avg	1	2	3	2	2	3	2	2	3	3	3	1	2	2

CO – WK Mapping (Knowledge & Attitude Profiles) Sample Table

'✓' indicates the knowledge profile drawn upon by the CO.

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1	✓	✓		✓					✓
CO2		✓	✓	✓	✓				
CO3		✓	✓	✓	✓	✓			
CO4			✓	✓	✓	✓	✓		
CO5				✓	✓	✓	✓	✓	
CO6	✓			✓		✓		✓	✓

CO – WP Mapping (Complex Engineering Problems) Sample Table

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO1	✓	✓	✓		✓		✓
CO2	✓	✓	✓	✓	✓		
CO3	✓	✓	✓	✓	✓	✓	
CO4	✓	✓	✓	✓	✓	✓	✓
CO5	✓	✓	✓	✓	✓	✓	✓
CO6			✓		✓	✓	✓

CO – EA Mapping (Complex Engineering Activities)

'✓' indicates the activity attribute addressed by the CO

CO	EA1	EA2	EA3	EA4	EA5
CO1	✓	✓			
CO2	✓	✓	✓		
CO3	✓	✓	✓	✓	
CO4		✓	✓	✓	✓
CO5			✓	✓	✓
CO6				✓	✓

CO – SDG Mapping (Sustainable Development Goals) Sample Table

CO	SDG(s) Addressed	Justification / Linkage
CO1	4, 9	Understanding the importance of research promotes quality education, scientific inquiry, innovation, and technological advancement for solving real-world problems.
CO2	4, 9	Knowledge of research terminology and concepts strengthens analytical thinking, research competency, and innovation capabilities.
CO3	9, 17	Applying research methodology and publishing research fosters innovation, knowledge dissemination, and collaboration among researchers and institutions.
CO4	4, 9, 17	Writing research papers develops scientific communication, promotes lifelong learning, and contributes to global knowledge sharing through collaborative research.
CO5	9,12	Understanding Intellectual Property Rights encourages innovation, protection of inventions, responsible use of knowledge, and sustainable technological development.
CO6	4, 16	Applying research ethics promotes academic integrity, responsible conduct of research, transparency, fairness, and ethical decision-making.
CO7	8, 9	Entrepreneurship and business planning encourage innovation, job creation, sustainable economic growth, and technology-based enterprises.

FF No.: 654-A

CI2016: Engineering Design and Innovations-I & II

Credits:.2 **Lab: 2 hours/Week**

s

Teaching Scheme	Examination Scheme
Credits : 2	CP: - Marks
Lectures : - Hrs/week	GD/PPT/HA: - Marks
Practical : - Hrs/week	MSE (W/O): 30 Marks
Tutorial : - Hrs/week	ESE (W/O): 70 Marks

Prerequisites:	Problem Based Learning
•	
Course Objectives:	
CO1	To develop critical thinking and problem-solving ability by exploring and proposing solutions to realistic/social problems.
CO2	To Evaluate alternative approaches, and justify the use of selected tools and methods,
CO3	To emphasize learning activities those are long-term, inter-disciplinary and student-centric.
CO4	To engage students in rich and authentic learning experiences.
CO5	To provide every student the opportunity to get involved either individually or as a group so as to develop team skills and learn professionalism.
CO6	To develop an ecosystem to promote entrepreneurship and research culture among the students
Course Outcomes:	
	After completion of the course, student will be able to
1.	Identify the real life problem from societal need point of view
2.	Choose and compare alternative approaches to select most feasible one
3.	Analyze and synthesize the identified problem from technological perspective
4.	Design the reliable and scalable solution to meet challenges
5.	Evaluate the solution based on the criteria specified
6.	Inculcate long life learning attitude towards the societal problems

Section1:	Topics/Contents

The content and process mentioned below is the guideline document for the faculties and students to start with. It is not to limit the flexibility of faculty and students; rather they are free to explore their creativity beyond the guideline mentioned herewith.

Group Structure: • There should be a team/group of 4-5 students. Computer Science & Engineering (Artificial Intelligence) –
A supervisor/mentor teacher assigned to individual groups. · It is useful to group students of different abilities and nationalities together.
Selection of Project/Problem:

- Students must focus to initiate the task/idea .The idea inception and consideration shall be from following areas as a real world problem:
- Health Care, Agriculture, Defense, Education, Smart City, Smart Energy, Swaccha Bharat Abhiyan, Environment, Women Safety.
- This is the sample list to start with. Faculty and students are free to include other areas which meet the society requirements at large.
- The model begins with the identifying of a problem, often growing out of a question or "wondering". This formulated problem then stands as the starting point for learning. Students design and analyze the problem/project within an articulated disciplinary subject frame/domain.
- A problem can be theoretical, practical, social, technical, symbolic, cultural, and/or scientific and grows out of students' wondering within different disciplines and professional environments. A chosen problem has to be exemplary. The problem may involve an interdisciplinary approach in both the analysis and solving phases.
- By exemplarity, a problem needs to refer back to a particular practical, scientific, social and/or technical domain. The problem should stand as one specific example or manifestation of more general learning outcomes related to knowledge and/or modes of inquiry.

Teacher's Role in PCL :

- Teacher is not the source of solutions rather he will they act as the facilitator and mentor.
- To utilize the principles of problems solving, critical thinking and metacognitive skills of the students.
- To aware the group about time management.
- Commitment to devote the time to solve student's technical problems and interested in helping students to empower them better.

Student's Role in PCL:

- Students must have ability to initiate the task/idea .they should not be mere imitators.
- Students working in PCL must be responsible for their own learning.
- Students must quickly learn how to manage their own learning, Instead of passively receiving instruction.
- Students in PCL are actively constructing their knowledge and understanding of the situation in groups.
- Students in PCL are expected to work in groups.

- They have to develop interpersonal and group process skills, such as effective listening or coping creatively with conflicts.

Developing Inquiry Skills:

- Students in PCL are expected to develop critical thinking abilities by constantly relating: What they read to do? What they want to do with that information?
- They need to analyze information presented within the context of finding answers.
- Modeling is required so that the students can observe and build a conceptual model of the required processes.
- Use the following mechanism to maintain the track of moving towards the solution. How effective is …….? How strong is the evidence for ……? How clear is ……?
- What are the justifications for thinking? Why is the method chosen?
- What is the evidence given to justify the solution?

Literature Survey – To avoid reinvention of wheel:

- It is integral part of self- directed learning
- Identify the information needed to solve a given problem or issue
- Be able to locate the needed information
- Use the information to solve the given problem effectively. • Skills required by students in information literacy include:
- How to prepare the search? How to carry out the research • Sorting and assessing of information in general

Use of Research Methodology:
investigation, collaboration, comprehension, application, analysis, synthesize and evaluation
Focus on following skills while working in a team to reach to solution:

- Collaborative learning
- Interpersonal Skills
- Resources Evaluation
- Metacognitive Skills
- Reflection Skills

EDD Sample Case Studies : -
With the adaptation of industry communication standards, Raspberry Pi and Sensors, following projects
can be taken up:
1) Design a deployable product for soil moisture detection
2) Design a deployable product for temperature detection
3) Design a deployable product for pressure detection
3) Design a deployable product smoke detection
4) Design a deployable product for motion detection
5) Design a deployable product for collision detection
6) Design a deployable product for sound detection

Text Books: ***(As per IEEE format)***

1	By Terry Barrett. All Ireland Society for higher education (AISHE). A new model of problem based learning. ISBN:978-0-9935254-6-9; 2017

2	By Mahnazmoallem, woei hung and Nada Dabbagh, Problem Based Learning. Wiley Publishers. 2019.
3	By Robert Robart Capraro, Mary Margaret Capraro Stem Project based learning and integrated science, Technology, Engineering and mathematics approach.
Reference Books: *(As per IEEE format)*	
1	De Graaff E, Kolmos A., red.: Management of change: Implementation of problem-based and project-based learning in engineering. Rotterdam: Sense Publishers. 2007.
2	. Gopalan.Project management core textbook, second edition, Indian Edition , by

CO – PO / PSO Articulation Matrix

Correlation: 1 = Low (Slight), 2 = Medium (Moderate), 3 = High (Substantial), '-' = no correlation.

CO	Program Outcome (PO)											PSO		
CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PSO1	PSO2	PSO3
CO1	2	3	1			2			2		1	2	1	1
CO2	2	3	2	2	1				2			2	2	1
CO3	2	3	2	3	2				2			2	3	1
CO4	2	2	3	2	2	2			2	1		2	3	2
CO5	2	2	2	3	2	2	1		2	1		1	3	2
CO6		1				2	2	1	2	1	3	1	2	3
Avg	1.83	2.33	2.2	2.4	2.5	2	1.5	2	2	1.33	2	1.67	2.5	1.83

CO – WK Mapping (Knowledge & Attitude Profiles) Sample Table

'✓' indicates the knowledge profile drawn upon by the CO.

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO1	✓	✓		✓					✓
CO2		✓	✓	✓	✓				
CO3		✓	✓	✓	✓	✓			
CO4			✓	✓	✓	✓	✓		

CO	WK1	WK2	WK3	WK4	WK5	WK6	WK7	WK8	WK9
CO5				✓	✓	✓	✓	✓	
CO6	✓			✓		✓		✓	✓

CO – WP Mapping (Complex Engineering Problems) Sample Table

CO	WP1	WP2	WP3	WP4	WP5	WP6	WP7
CO1	✓	✓	✓		✓		✓
CO2	✓	✓	✓	✓	✓		
CO3	✓	✓	✓	✓	✓	✓	
CO4	✓	✓	✓	✓	✓	✓	✓
CO5	✓	✓	✓	✓	✓	✓	✓
CO6			✓		✓	✓	✓

SCO – EA Mapping (Complex Engineering Activities)

'✓' indicates the activity attribute addressed by the CO

CO	EA1	EA2	EA3	EA4	EA5
CO1	✓	✓			
CO2	✓	✓	✓		
CO3	✓	✓	✓	✓	
CO4		✓	✓	✓	✓
CO5			✓	✓	✓
CO6				✓	✓

CO – SDG Mapping (Sustainable Development Goals) Sample Table

CO	SDG(s) Addressed	Justification / Linkage
CO1	9, 11	Identifies real-life societal problems and community needs, promoting sustainable infrastructure and resilient communities (SDG 9, SDG 11).

CO	SDG(s) Addressed	Justification / Linkage
CO2	9, 12	Compares alternative solutions to select feasible, efficient, and resource-conscious approaches that support innovation and responsible practices (SDG 9, SDG 12).
CO3	9	Analyzes technological aspects of the identified problem to develop innovative engineering solutions (SDG 9).
CO4	9, 11	Designs reliable and scalable engineering solutions that improve infrastructure and address societal challenges (SDG 9, SDG 11).
CO5	9, 12	Evaluates solutions using defined criteria to ensure quality, efficiency, sustainability, and responsible resource utilization (SDG 9, SDG 12).
CO6	4, 17	Promotes lifelong learning, continuous professional development, and collaboration for solving societal problems (SDG 4, SDG 17).