Centre for Artificial Intelligence

DISCRETE STRUCTURES

(31251201/ 24251201/ 27251201/ 28251201)

COURSE OBJECTIVES

- To perceive the knowledge of basic algebra and propositional logic
- To know about the graph theory and its application in computer engineering
- To familiarize the discrete numeric function and generating function

UNIT-I: Finite and Infinite Sets, Mathematical Induction, Principles of Inclusion and Exclusion, Multisets, Functions and Relations, Binary Relations, Equivalence Relations and Partitions, Partial Ordering Relations and Lattices, Chains, Pigeonhole Principle.

UNIT-II: Propositional Logic, Syntax, Semantics of ATF (Atomic Formula), WFF (Well Formed Formulas),Validity and Satisfiability of WFF by Quine's Method, Normal and Closure Form of Propositional Calculus.

UNIT-III: Introduction and Basic Terminology of Graphs, Planar Graphs, Multigraphs and Weighted Graph, Shortest Path in Weighted Graph, Introduction to Eulerian Paths and Circuits, Hamiltonian Paths and Circuits. Introduction to Trees, Rooted Trees, Path Length in Rooted Trees, Spanning Trees and Cut Trees.

UNIT-IV: Introduction to Discrete Numeric Functions and Generating Functions, Introduction to Recurrence Relations and Recursive Algorithms, Linear Recurrence Relations With Constant Coefficients, Homogeneous Solutions, Particular Solutions and Total Solutions.

UNIT-V: Binary Group Codes, Communication system and its problems, Binary Symmetric Channel, Encoding and Decoding, Error detecting and correcting codes, Block codes, Distance between words, Matrix Encoding Technique, Groups codes, Construction of Decoding Table, Hamming codes.

RECOMMENDED BOOKS

1. J. Tremblay and Manohar: Discrete Mathematical Structures with Application to Computer science. Narsingh Deo: Graph Theory.
2. Kenneth Rosen: Discrete mathematics and its applications (6th edition), 2006. McGraw-Hill
3. C. Liu, D. Mohapatra: Elements of Discrete Mathematics. 2008. Tata McGraw-Hill.
4. T. Koshy: Discrete mathematics with applications, 2003. Academic Press.
5. J. Hein: Discrete structures, logic and computability, 2009. Jones & Bartlett Publishers.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1	explain the concepts of set theory, propositional logic, graph theory, discrete numeric function and algebraic structure.
CO2	apply mathematical reasoning and logical thinking to solve problems
CO3	determine the solutions of problems pertaining to computer sciences using graph theory concepts.
CO4	solve counting and recursive problems using combinatorial analysis.
CO5	analyze error control coding techniques to enhance communication system reliability.

CO-PO Mapping Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	2	3	2	1		1	2	2	2	3	2	2
CO2	2	3	3	2	3	2	1		2	2	3	2	3	2
CO3	2	2	2	2	2	1			1	1	2	3	2	2
CO4	3	3	2	3	2	2			1	1	2	2	2	1
CO5	2	3	2	2	3	2		2	2	1	2	2	3	3

Centre for Artificial Intelligence
MODERN COMPUTER ARCHITECTURE
(31251202/ 24251202/ 27251202/ 28251202)

COURSE OBJECTIVES

- To understand basic principles of Computer Systems.
- To understand various logic design techniques and their applications. They should be capable of using high performance computing architecture.

UNIT-I: Basic of Computer Architectures: Von Neumann Model, Various Subsystems, CPU, Memory, I/O, System Bus, Arithmetic Micro-Operation, Logic Micro Operation, Shift Micro-Operation, Register Transfer Micro Operations.

UNIT-II: Multi-core Architecture: Memory technologies, hierarchical memory systems, the locality principle and caching, direct- mapped caches, block size, cache conflicts, associative caches, write strategies, advanced optimizations, performance improvement techniques, DRAM – organization, access techniques, scheduling algorithms, and signal systems.

UNIT-III: Distributed Computing Systems and Concurrency: Relation to Parallel Multiprocessors/multicomputer Systems, Distributed and Concurrent Programs, Message Passing vs. Shared Memory Systems, Synchronous vs. Asynchronous Executions, Design Issues and Challenges, Distributed Computing Technologies, Clocks and Synchronization, Global State and Distributed Transactions.

UNIT-IV: High Performance Computing (HPC): HPC Architecture, Parallel Processing, Parallel Memory Models, Data vs. Task Parallelism, High Throughput Computing, Vectorization, Multithreading.

UNIT-V (DYNAMIC CONTENTS): High Performance Computing with CUDA: CUDA programming model, Basic principles of CUDA programming, Concepts of threads and blocks, GPU and CPU data exchange

RECOMMENDED BOOKS

1. M. Morris Mano, Computer System & Architecture, Prentice Hall of India, 2002.
2. John L. Hennessy and David A Patterson, Computer Architecture-A quantitative approach, Morgan Kaufmann/ Elsevier, 4th Edition, 2007.
3. Hayes. J.P, Computer architecture and organization by McGraw-Hill Companies, 1998
4. Parallel Computer Architecture: A Hardware/Software Approach David Culler and J.P. Singh with Anoop Gupta, Morgan Kaufmann, 1998.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1	describe the organization of the Control unit, ALU, Memory and the I/O unit.
CO2	evaluate hierarchical memory systems, design effective caching strategies, and optimize memory performance using advanced techniques and scheduling algorithms.
CO3	compare distributed and parallel computing paradigms, address concurrency challenges, and implement synchronization techniques for distributed systems.
CO4	design parallel processing architectures and implement multithreading for high-performance computing applications.
CO5	develop parallel programs using the CUDA programming model, manage GPU-CPU data exchange, and optimize computations with threads and blocks.

CO-PO Mapping Matrix														
	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3	3	3	3	2	1		1		1		2	1	2
CO2	3	3	3	3	3	2	2	2	1	1	2	3	2	2
CO3	3	3	3	3	3	2		1	1	1		3	1	1
CO4	3	3	3	3	3	3	3	2	3	1	3	3	3	2
CO5	3	3	3	3	3	3	3	2	3	2	3	3	3	3

Centre for Artificial Intelligence
OBJECT ORIENTED PROGRAMMING
(31251203/ 24251203/ 27251203/ 28251203)

COURSE OBJECTIVES

- To study the concept of object oriented programming.
- To create programs that leverage the object oriented features of the C++ Language.
- To apply object oriented programming techniques for real world problem solving.

UNIT-I: Introduction to Object Oriented Programming: Unstructured & Structured Programming, Object Oriented Paradigm, features and comparison with Procedural Oriented Programming approach. Specification of Class, Abstract Data Types, Visibility Modes, Defining Member Functions, Object Creation, Characteristics of Object, Scope Resolution Operator, Static Data Member, Static Member Function.

UNIT-II: Constructors and Destructors: Introduction, Types of Constructors- Default Constructor, User Defined Constructor, Parameterized Constructor, Copy Constructor, Constructor with Default Arguments, Rules of Constructor Definition and Usage, Destructors. Array of Objects, Object as Arguments, Inline Function, Friend Function.

UNIT-III: Polymorphism: Introduction, Type of Polymorphism: Compile Time Polymorphism & Run Time Polymorphism, Function Overloading, Operator Overloading: Binary and Unary Operators, Rules for Operator Overloading, Pitfalls of Operator Overloading, Data Conversion, Type Casting.

UNIT-IV: Inheritance: Introduction to Code Reuse, Visibility Modes, Types of Inheritance: Single Level, Multilevel, Multiple, Hybrid, Multipath. this Pointer, Ambiguity in MultiLevel Inheritance, Constructors in Derived Classes, Virtual Functions, Virtual Base Classes, Abstract Classes and Pure Virtual Functions, Nesting of Classes, Overriding Member Function, Association, Type of Association.

UNIT-V (DYNAMIC CONTENTS): Templates and Exceptions: Function Templates, Function Templates with Multiple Arguments, Class Templates, Exception Syntax, Exceptions with Arguments. Standard Template Library: Containers, Algorithms, Iterators, Vectors and Lists.

RECOMMENDED BOOKS

1. Object Oriented Programming in C++, Robert Lafore, Sams.
2. C++ How to Program, H M Deitel and P J Deitel, Prentice Hall.
3. The Complete Reference in C++, Herbert Schildt, TMH.
4. Fundamentals of Programming C++, Richard L. Halterman.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1	describe fundamental principles of object-oriented programming.
CO2	construct and manage object life cycles using various types of constructors and destructors, while employing inline and friend functions effectively in programming.
CO3	apply compile-time and runtime polymorphism using function and operator overloading, while addressing potential challenges of operator overloading and type casting.
CO4	develop well-structured, modular programs that leverage classes, objects, and inheritance to enhance code maintainability and reusability.
CO5	design reusable and generic code using function and class templates, handle runtime errors with exception handling, and utilize the Standard Template Library for efficient data manipulation.

CO-PO Mapping Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	3	3	3						3	1	1
CO2	3	3	3	3	3	2	2					3	2	2
CO3	3	3	3	3	3	3	3					3	3	3
CO4	3	3	3	3	3	3	3	3	3	1	2	3	3	2
CO5	3	3	3	3	3	3	2	3	3	1	3	3	3	3

Centre for Artificial Intelligence

DATA STRUCTURES

(31251204/ 24251204/ 27251204/ 28251204)

COURSE OBJECTIVES

- To be familiar with the use of data structures as the foundational base for computer solutions to problems.
- To understand various techniques of searching and sorting.
- To understand basic concepts about stacks, queues, lists, trees and graphs.

UNIT-I: Introduction to Data Structures: Algorithms & their characteristics, asymptotic notations. arrays and its representations, index to address translation. Link list: Introduction, implementation of linked list, operations, circular link list, doubly linked list, polynomial manipulation using linked list.

UNIT-II: Stacks: Concepts and implementation of stacks, operations on stack, conversion of infix to postfix notation, evaluation of postfix expression, recursion. Queues: Concepts and implementation, operations on queues, dequeue, priority queues, circular queues and application.

UNIT-III: Trees: Types, terminology, binary tree -representations, traversal, conversion of general tree to binary tree, binary search tree, threaded binary tree and height balanced tree. Heaps: binary heap, heap operations, heap sort.

UNIT-IV: Graphs: Background, graph theory terminologies, representation of graphs- sequential & linked representation, path matrix, graph traversals- BFS, DFS, spanning trees, applications of graph. Searching & Sorting: Linear search, binary search, bubble sort, selection sort, insertion sort, quick sort, merge sort.

UNIT-V (DYNAMIC CONTENTS): Hash tables: hash functions, collision resolution (chaining, open addressing) Load factor, rehashing, performance implications. Dynamic / Self-Adjusting Data Structures: Splay trees. Cache-Aware Data Structures: B-trees, van Emde Boas trees.

RECOMMENDED BOOKS

1. Data Structures, Algorithms and Applications in C++, Sartaj Sahni, 2nd Edition.
2. An Introduction to Data Structures with Applications, Jean-Paul Tremblay, Mcgraw hill.
3. Data Structures & Algorithms, Aho, Hopcroft & Ullman, original edition, Pearson Publication.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1	analyze algorithms using asymptotic notations and perform operations on arrays and linked lists.
CO2	construct stacks and queues, and use them to solve real world problems.
CO3	distinguish between different types of trees and identify the application of each.
CO4	apply graph based and sorting-searching algorithms.
CO5	compare various hashing techniques and advanced data structures

Centre for Artificial Intelligence

CO-PO Mapping Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	3	1							1	1	1
CO2	3	3	3	3	2				1			2	2	2
CO3	3	3	3	3	2		1		1	1		2	2	2
CO4	3	3	3	3	2	1	1	1	1	1	1	2	2	2
CO5	3	3	3	3	2	1			1	1		2	2	2

Centre for Artificial Intelligence

BASIC ELECTRICAL & ELECTRONICS ENGINEERING

(31251205/ 27251205/ 28251205)

COURSE OBJECTIVES

- To introduce the fundamental principles of electrical circuits and electronics.
- To provide insights into basic semiconductor devices, including diodes, transistors, and their applications.
- To demonstrate the practical use of electrical and electronic components in real-world systems.

UNIT-I: Basic Concepts of Electrical Engineering-Electrical Circuit Elements: Resistors, Capacitors, and Inductors, Ohm's Law, Kirchhoff's laws, Series and Parallel Circuits, star-delta circuits, charging and discharging of capacitor, series-parallel magnetic circuits, comparison between electric and magnetic circuit, Concept of induced EMFs..

UNIT-II: Single-phase AC Circuits-Generation of alternating emf, instantaneous, rms, peak, average values and related other terms, vector representation of AC quantities, Steady state analysis of R, L, C series and parallel circuits, power triangle. Three-phase AC Circuits-Generation of three-phase emf, star connection, delta connection, relationship between line and phase quantities

UNIT-III: Single Phase Transformer-Principle of Transformers: Construction and Working, Types of transformer, EMF equation of transformer, Transformation ratio, Applications of transformer.

UNIT-IV: Basic Electronics-Overview of Semiconductor Physics, PN Junction Diodes: Characteristics and Applications, Zener Diodes: Voltage Regulation, Half and full wave rectifiers, special purpose diodes.

UNIT-V (DYNAMIC CONTENTS): Smart Electrical Components & Digital Twin technology, smart grids, and high-efficiency power electronics. Wide-bandgap materials, high-frequency rectification and quantum electronic systems.

RECOMMENDED BOOKS

1. Basic Electrical Engineering, Authors: V.K. Mehta and Rohit Mehta, Publisher: S. Chand Publishing
2. Fundamentals of Electrical Engineering I, Author: Don H. Johnson, Publisher: Rice University (Free Online Resource)
3. Electrical Engineering 101: Everything You Should Have Learned in School...but Probably Didn't, Author: Darren Ashby, Publisher: Newnes (Elsevier), Edition: 3rd Edition
4. Principles of Electronics, Author: Colin Simpson, Publisher: Cengage Learning
5. Basic Electrical Engineering, Authors: D.P. Kothari and I.J. Nagrath, Publisher: McGraw Hill Education.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1	analyze electrical and magnetic circuits to evaluate transient responses in capacitive and inductive circuits.
CO2	analyze single-phase and three-phase AC circuits, including power measurements and power factor correction.
CO3	explain the generation of three-phase EMFs
CO4	describe the construction and working principles of transformers, analyze their performance using EMF equations and phasor diagrams, and evaluate their efficiency and applications.
CO5	apply semiconductor physics principles to analyze PN junction and Zener diodes, design rectifiers, and identify the applications of special-purpose diodes in circuits.

CO-PO Mapping Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	2	1	-	-	-	-	-	-	-	2	1	1
CO2	3	3	3	2	2	1	-	-	-	-	-	2	2	1
CO3	3	3	3	2	2	1	-	-	-	-	-	2	1	1
CO4	3	2	3	2	-	-	-	-	-	-	-	2	1	2
CO5	3	3	3	3	3	2	2	2	2	2	2	2	2	2

Centre for Artificial Intelligence
SENSORS AND ACTUATORS
(24251205)

COURSE OBJECTIVES

- Illustrate the working principles of transducers, sensors and actuators.
- Develop and exemplify basic programming skills in Virtual Instrumentation.
- Design and implement a system using sensor and instrumentation configuration.

UNIT-I: Introduction: Introduction, Input output configuration, generalized functional elements, advantages of electronic measurement, Errors in measurement, Gross errors and systematic errors, Absolute and relative errors, static characteristics, dynamic characteristics, calibration.

UNIT-II: Transducers: Introduction, Resistive transducer, Resistive position transducer, Strain gauges, Resistance thermometer, Thermistor, Inductive transducer, Capacitive transducers, Differential output transducers and LVDT. Piezoelectric transducer, photoelectric transducer, Photovoltaic transducer. Temperature transducers.

UNIT-III: Sensors: Introduction, principles, classification, characterization, Smart sensors: Introduction Primary sensors Information coding/ processing, Data communication, automation. Introduction to MEMS and Microsystems, Microsystems and Microelectronics Multidisciplinary nature of micro system design and manufacture applications of micro systems, Micro sensors, Humidity and Moisture Sensors.

UNIT-IV: Actuators: Functional components of an actuator, Performance Characteristics of Actuators, Thermo-mechanical Actuators, Optical Actuators, Electrical actuating systems: Solid-state switches, Solenoids, Electric Motors-D.C, AC motors; Piezoelectric Actuator. Magnetic Actuators, Capacitive Actuators,. Micro Actuators: Actuation principle, shape memory effects-one way, two way and pseudo elasticity. Types of micro actuators-Electrostatic, Magnetic, Fluidic, Inverse piezo effect, other principles.

UNIT-V (DYNAMIC CONTENTS): Design and fabrication process of Microsensors, Intelligent & Self sensing actuators, Smart Sensor Technologies: development trend, characteristics of smart sensors, application areas of smart sensors in IoT and Robotics. Sensors for Navigation and localization: LiDAR-SLAM (Simultaneous Localization and Mapping).

RECOMMENDED BOOKS

1. Patranabis.D, “Sensors and Transducers”, Wheeler publisher, 1994.
2. D.V.S. Murthy: “Transducers and Instrumentation”, 2nd Edition, PHI Ltd., 2014
3. Sergej Fatikow and Ulrich Rembold, “Microsystem Technology and Macrobiotics”, First edition, Springer –Verlag NEwyork, Inc, 1997.
4. Jacob Fraden, “Hand Book of Modern Sensors: Physics, Designs and Application” Fourth edition, Springer, 2010.

REFERENCE BOOKS

1. Robert H Bishop, “The Mechatronics Hand Book”, CRC Press, 2002.
2. Massood Tabib and Azar, “Microactuators Electrical, Magnetic, thermal, optical, mechanical, chemical and smart structures”, First edition, Kluwer academic publishers, Springer, 1997.
3. Manfred Kohl, “Shape Memory Actuators”, first edition, Springer.

Centre for Artificial Intelligence

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1	describe the functional elements of measurement systems, identify different types of errors, and evaluate the static and dynamic characteristics of measurement systems.
CO2	analyze various types of transducers for diverse measurement applications.
CO3	explain the principles and applications of sensors, characterize smart sensors, and analyze the design and operation of MEMS-based systems for multidisciplinary applications.
CO4	assess the performance and suitability of selected sensors and actuators for a specific control system.
CO5	apply the principles of micro-actuation, analyze the design and fabrication of microsensors and microactuators, and explore the integration of smart sensor technologies with IoT systems for innovative applications.

CO-PO Mapping Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	2	3	3	2	1						2		1
CO2	3	2	3	3	2	1				1		2	1	
CO3	3	3	3	3	2	2	2		1		1	2	2	2
CO4	3	3	3	3	3	2	2	2	3	1	2	2		2
CO5	3	3	3	3	3	2	3	2	3	1	2	3	2	2

Centre for Artificial Intelligence

SUSTAINABILITY & ENVIRONMENTAL SCIENCE

(31251211/ 24251211/ 27251211/ 28251211)

COURSE OBJECTIVES

- To equip students with a comprehensive understanding of environmental science, pollution control, sustainability, and global frameworks, enabling them to analyze environmental challenges and contribute to sustainable solutions through informed decision-making and responsible practices.

UNIT-I: Introduction to Environmental Science: definition, importance and its components. Ecosystem and its components. Water cycle, carbon cycle, food chain, energy flow in the ecosystem. Current state of environment in India and world; Underlying reasons (root causes) of modern environmental degradation (social, psychological, cultural).

UNIT-II: Environmental Pollution and Management: air, water, noise, soil, thermal and radioactive. Causes, impacts, pollution control techniques and mitigation strategies. Solid waste management: Principles of waste management, different components of waste management system and introduction to management of hazardous waste like e-waste, plastic waste. Global environmental Issues: Climate change, global warming, ozone layer depletion.

UNIT-III: Environmental policies and laws in India: Environmental Protection Act, Water Act, Air Act. Overview of global environmental policies and frameworks: Kyoto protocol, Montreal protocol, COP summits. Introduction to clean development mechanism, carbon credit, carbon trading.

UNIT-IV: Sustainability concepts: definition, importance, pillars of sustainability (economic, environmental, and social). Sustainable development. Overview of UN Sustainable Development Goals (SDGs) and their global relevance. Concept of circular economy, resource efficiency, energy conservation, green buildings and sustainable manufacturing.

UNIT-V: Sustainable Energy solutions: New Energy Sources: Need of new sources. Different types of new energy sources. Applications of- Hydrogen energy, Ocean energy resources, Tidal energy conversion. Concept, origin and power plants of geothermal energy. Introduction to sustainable transportation systems and sustainable water infrastructure.

RECOMMENDED BOOKS

1. D. K. Asthana, Meera Asthana, A Text Book of Environmental Studies, S Chand & Co., New Delhi.
2. S. K. Dhameja, Environmental Engineering & Management, S K Kataria & Sons, New Delhi
3. C. S. Rao, Environmental Pollution Control Engineering, C.S. Rao, New Age International Publishers
4. A. K. Gupta, Environmental Sustainability and Green Technologies, PHI Learning.

Centre for Artificial Intelligence

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1	Explain the fundamental concepts of environmental science, including ecosystems and the causes of environmental degradation.
CO2	Analyze the sources, causes, and impacts of air, water, and solid waste pollution and propose appropriate mitigation strategies.
CO3	Evaluate the effectiveness of environmental policies and global frameworks in addressing environmental challenges.
CO4	Explain the concepts of sustainability and sustainable development goals.
CO5	Apply various solutions for achieving sustainable development.

CO-PO Mapping Matrix

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	2	-	-	-	-	-	1	-	-	-	-	1	-	-
CO2	2	2	2	-	-	-	3	-	-	-	-	2	-	-
CO3	-	-	1	-	-	2	2	-	-	-	-	2	2	2
CO4	-	-	-	-	-	-	2	-	-	-	-	2	-	-
CO5	2	2	2	-	-	1	3	-	-	-	-	2	2	2

ANNEXURE - III

Experiment list/ Lab manual and project list (Micro Project-II) of all the Laboratory Courses
B.Tech.
[Artificial Intelligence (AI)/ Information Technology (Artificial Intelligence and Robotics)/ Artificial Intelligence (AI) and Data Science/ Artificial Intelligence (AI) and Machine Learning]
II Semester
for batch admitted in 2025-26

Centre for Artificial Intelligence
OBJECT ORIENTED PROGRAMMING LAB
(31251206/ 24251206/ 27251206/ 28251206)

List of Programs

1. Define a class representing a basic entity (e.g., a car, a person) with member variables and member functions. Create objects of the class and demonstrate their usage.
2. Implement a class with a constructor and destructor. Illustrate their roles and demonstrate how they are called during object creation and destruction.
3. Create a base class and a derived class. Demonstrate inheritance by inheriting properties and methods from the base class to the derived class using different types of inheritances.
4. Implement a polymorphic behavior using function overloading or overriding. Show how a function can behave differently based on the type of object.
5. Overload a few operators (e.g., +, -, *, =) for a custom class. Show how these operators can be used with objects of that class.
6. Implement a template class that can work with different data types. Demonstrate how to create objects of the template class with different data types.
7. Design a program that uses exception handling to catch and handle runtime errors, such as division by zero or array out-of-bounds.
8. Utilize STL containers like vectors or lists in a program. Perform basic operations like insertion, deletion, and iteration.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1	apply the principles of object-oriented programming paradigm to programming problems.
CO2	use classes and objects to model real-world entities in a program.
CO3	apply the concepts of base classes, derived classes, and method overriding.
CO4	implement exception handling to manage errors and unexpected situations in a program.
CO5	create generic and reusable code using STL.

CO-PO Mapping Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	3	3							3	1	1
CO2	3	3	3	3	3							3	2	2
CO3	3	3	3	3	3						2	3	3	3
CO4	3	3	3	3	3	2	2	2	2	1	2	3	3	2
CO5	3	3	3	3	3	3	2	3	3	1	3	3	3	3

Centre for Artificial Intelligence
DATA STRUCTURES LAB
(31251207/ 24251207/ 27251207/ 28251207)

List of Programs

1. Implement Singly, Doubly and Circular Linked List.
2. Implement Stack and Queue using arrays.
3. Implement Stack and Queue using Singly linked lists.
4. Create a program which creates a Binary search tree and traverse it using Inorder, Pre-order and Post-order traversal techniques.
5. Write a program to implement Threaded Binary trees.
6. Implement a graph data structure using Adjacency matrix and list respectively.
7. Write a program which finds the spanning tree of a graph.
8. Implement BFS and DFS graph traversal techniques.
9. Implement following searching and sorting algorithms:
    a. Linear and Binary Search.
    b. Bubble sort.
    c. Selection sort
    d. Insertion sort
    e. Merge sort
    f. Quick sort
    g. Heap sort
10. Define function for performing insertion, deletion, searching, and traversal operations on B-tree and Splay tree.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1	analyze the complexity of different data structures.
CO2	choose appropriate data structures for specific scenarios.
CO3	design solutions using tree and graph based data structures.
CO4	solve real world problems using appropriate data structures.

CO-PO Mapping Matrix

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	3	3	1							1	1	1
CO2	3	3	3	3	2				1			2	2	2
CO3	3	3	3	3	2		1		1	1		2	2	2
CO4	3	3	3	3	2	1	1	1	1	1	1	2	2	2

Centre for Artificial Intelligence

BASIC ELECTRICAL & ELECTRONICS ENGINEERING LAB

(31251208/ 27251208/ 28251208)

List of Experiments

1. Verify Ohm's law by measuring current across different resistors with varying applied voltages.
2. Verify Kirchhoff's Voltage Law (KVL) and Kirchhoff's Current Law (KCL) in a simple electrical circuit.
3. Analyze and measure the voltage and current in series and parallel resistor configurations
4. Solve the star-delta circuit using practical measurements on the breadboard.
5. Study the charging and discharging behavior of a capacitor.
6. Analyze the behavior of R, L, and C components in series and parallel combinations in AC circuits.
7. Verify the power triangle in an RLC circuit and measure power factor.
8. Observe the forward and reverse characteristics of a PN junction diode.
9. Study the voltage regulation characteristics of a Zener diode.
10. Study the behavior of LEDs and photodiodes.
11. Study a Zener diode shunt voltage regulator and observe how output voltage changes when the load current varies.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1	Apply the fundamental concepts of electrical engineering to analyze and solve basic electrical circuits.
CO2	Analyze and measure electrical parameters (voltage and current) in series and parallel configurations
CO3	Explain the concepts related to AC circuits, including the generation of alternating current, AC circuit analysis, power calculation, and resonance in RLC circuits.
CO4	Analyze the working of Bipolar Junction Transistors (BJT)
CO5	Design and construct basic electronic circuits on breadboards

CO-PO Mapping Matrix

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3	2	2	1	-	-	-	-	-	-	-	2	1	1
CO2	3	3	3	2	2	1	-	-	-	-	-	2	2	1
CO3	3	3	3	2	2	1	-	-	-	-	-	2	1	1
CO4	3	3	3	2	2	1	-	-	-	-	-	2	2	1
CO5	3	3	3	3	3	2	2	2	2	2	2	2	2	2

Centre for Artificial Intelligence
SENSORS AND ACTUATORS LAB
(24251208)

List of Programs/Experiments

1. Study of static and dynamic characteristics of sensors.
2. Displacement measurement using LVDT.
3. Using a strain gauge transducer, strain is measured.
4. Displacement measurement with a potentiometer.
5. RTD is used to measure temperature, and its properties are shown.
6. Temperature measurement with a thermistor.
7. Pressure measurement with a load cell.
8. Speed measurement with a magnetic sensor.
9. Speed measurement with photoelectric sensors.
10. Use a pressure transducer to measure pressure.
11. Liquid level measurement with a capacitive sensor.
12. Interface of DC motors using H-bridge circuit
13. Implement and analyze speed control of a DC motor by varying the PWM duty cycle.
14. Understand the working of servo motors and control their position using PWM.
15. Study of AC Motor for wide range of speeds and loads
16. Working of stepper motor under speed and torque control.
17. Virtual demonstration of hydraulic and pneumatic actuators and their industrial applications.
18. Design and application of Micro actuators and micro valves.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1	apply appropriate sensors and transducers to measure physical quantities.
CO2	design interfacing circuits for electromechanical actuators.
CO3	create application-oriented solutions integrating suitable sensors and actuators for industrial or automation scenarios.

CO-PO Mapping Matrix

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3	3	2	3	3			2		1		2		2
CO2	3	3	2	3			3	2					2	2
CO3	3	3	3	3	3	3			1	2	3	3	2	3

Centre for Artificial Intelligence

MICRO PROJECT-II

(31251210/ 24251210/ 27251210/ 28251210)

1. Develop a system to manage student information with classes for students, courses, and grades. Implement features like adding and removing students, enrolling in courses, and calculating GPA.
2. Create a system to manage inventory with classes for products, categories, and orders. Implement features like adding and removing products, updating stock levels, and processing orders.
3. Develop a system for managing hospital records with classes for patients, doctors, and appointments. Implement features like scheduling appointments, updating patient information, and generating reports.
4. Create a quiz game with classes for questions, quizzes, and players. Implement a scoring system, random question selection, and a leaderboard for tracking player performance.
5. Design a simple social media network with classes for users, posts, and comments. Implement features like creating posts, adding friends, and commenting on posts.
6. Develop a ticket booking system for a cinema or an event with classes for shows, venues, and tickets. Include features like booking tickets, checking seat availability, and generating tickets.
7. Use a heap to implement a priority queue.
8. Develop a basic hash table with collision resolution
9. Implement a program that can parse and evaluate mathematical expressions.
10. Use a stack to implement an expression evaluator.
11. Implement a Phone directory application using doubly linked lists.
12. Implement a flight reservation system using a combination of data structures like priority queues for seat reservations and graphs for flight connections.
13. Library Management System-Create classes for books, members, and transactions. Implement book issue/return, overdue fine calculation, and search by author/title.
14. Bank Account Management System-Include classes for accounts, customers, and transactions. Implement deposit, withdrawal, fund transfer, and mini-statement generation.
15. Restaurant Ordering System-Build classes for menu items, orders, and customers. Include order placement, bill generation, and applying discounts or taxes.
16. Hotel Room Booking System-Use classes for rooms, customers, and bookings. Implement check-in/check-out, room availability checks, and billing.
17. Online Shopping Cart System-Include classes for users, items, carts, and payments. Implement add/remove items from cart, order summary, and checkout.
18. Task Management Application-Build classes for tasks, users, and categories. Implement adding, editing, deleting tasks, deadlines, and reminders.
19. Expense Tracker Application-Include classes for expenses, categories, and users. Implement monthly summaries, filtering by category, and budget tracking.
20. Music Playlist Manager-Use classes for songs, playlists, and users. Implement adding/removing songs, sorting playlists, and shuffle mode.
21. Employee Payroll System-Create classes for employees, salaries, and attendance. Implement salary calculation, tax deduction, and payslip generation.
22. Weather Monitoring System-Include classes for sensors, readings, and alerts. Implement data logging, average computation, and extreme weather alerts.
23. Parking Lot Management System-Build classes for vehicles, parking slots, and tickets. Implement slot allocation, time-based fee calculation, and automatic exit logs.
24. Chat Application (Local Simulation)-Create classes for users, messages, and chats. Implement sending messages, chat logs, and user online/offline status.

25. Road Navigation System-Use graphs to represent cities/roads. Implement shortest path search (Dijkstra), route suggestions, and distance calculation.
26. Movie Recommendation System-Classes for users, movies, ratings. Implement similarity-based recommendations and generate top-rated lists.
27. File Compression Tool (Basic)-Implement Huffman Coding using trees and priority queues to compress and decompress text files.
28. Scheduling System Using Priority Queue-Create a CPU scheduling simulator with classes for processes. Implement FCFS, SJF, Priority Scheduling, and Round-Robin.
29. E-Voting System-Include classes for voters, candidates, and ballots. Implement voter authentication, vote casting, and result tallying.
30. Student Attendance Tracker-Create classes for students, subjects, and attendance records. Implement marking attendance, monthly reports, and attendance percentage calculation.

COURSE OUTCOMES

After completion of this course, the students would be able to:

CO1	Explain the principles of OOP and their applications in solving real-world problems.
CO2	Develop and manipulate data structures like stacks and queues to solve problems.
CO3	Examine the relationships between OOP principles and data structure implementations in program design.
CO4	Design and develop a micro-project that integrates OOP concepts and appropriate data structures to solve a real-world problem.
CO5	Construct reusable and modular code using OOP principles for solving a complex programming challenge.

CO-PO Mapping Matrix

	PO1	**PO2**	**PO3**	**PO4**	**PO5**	**PO6**	**PO7**	**PO8**	**PO9**	**PO10**	**PO11**	**PO12**	**PSO1**	**PSO2**
CO1	3	3	3	3	3	3					1	3	1	1
CO2	3	3	3	3	3	2					1	3	2	2
CO3	3	3	3	3	3	3					1	3	3	3
CO4	3	3	3	3	3	3	3	3	3	1	2	3	3	2
CO5	3	3	3	3	3	3	2	3	3	1	3	3	3	3