Bachelor of Technology (Computer Science and Engineering)

Semester-III

L-2 T-1 P-2 C-4

CSE130 TR1 - Object Oriented Programming in Java

Course Objectives

- To understand concepts of Object-Oriented Programming (OOP) and their implementation in Java.
- Learn to implement inheritance, interfaces, and packages to design modular and reusable code.
- To Explore exception handling and multithreading to write robust, concurrent Java applications.
- Apply the Java Collections Framework and file handling to manage data effectively.
- Create graphical user interfaces (GUI), database connectivity (JDBC), and web development (Servlets) to build interactive Java applications.

Course Outcomes (COs)

1. Understand and apply OOP principles such as encapsulation, inheritance, and polymorphism, and implement basic Java programs using control structures, arrays, and string handling..
2. Apply inheritance, interfaces, and packages to design modular and maintainable Java programs.
3. Apply exception handling and multithreading to develop robust applications that can handle concurrency and runtime errors effectively.
4. Apply Java Collections Framework and perform file input/output operations to manage and process data efficiently in Java applications.
5. Create interactive Java applications using basic GUI components, connect to databases using JDBC, and create web applications with Servlets.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	1	1	2	3	-	-	-	-	-	-	-	-	-	-
CO2	1	2	2	2	-	3	-	-	-	-	-	-	2	-	-
CO3	1	2	3	2	-	2	-	-	-	-	-	-	1	-	-
CO4	1	2	2	2	1	2	-	-	-	-	-	-	2	-	-
CO5	1	1	1	2	2	2	-	-	1	1	-	-	1	-	-

High-3 Medium-2 Low-1

Unit I: Introduction to OOP and Basic Java Concepts

Object-Oriented Programming (OOP) Concepts: Principles of Object-Oriented Programming Introduction to Java: Overview of Java, History and Features, Java Development Kit (JDK), Java Runtime Environment (JRE), Integrated Development Environment (IDE), Writing, compiling, and running a Java program, ,

Classes and Objects: Defining a class, Creating objects, Constructors, Methods: Defining methods, Method overloading, Basic Syntax and Structure: Data types, Variables, Operators, Control Structures: if, if-else, switch,

Loops: for, while, do-while, **Arrays:** Single-dimensional arrays, Multi-dimensional arrays, Basic Array operations, String Handling: String class, String methods, String manipulation, Basic Input and Output.

Unit II: Inheritance, Interface and Packages

Inheritance: Introduction to inheritance, Types of inheritance, Single inheritance, Multilevel inheritance, Hierarchical inheritance, Multiple inheritance issues, Method overriding, super keyword, final keyword, Constructors in inheritance, Abstract classes and methods,

Interfaces: Introduction to interfaces, Defining interfaces, Implementing interfaces, Multiple inheritance through interfaces, Extending interfaces, Default methods in interfaces, Functional interfaces,

Packages: Introduction to packages, Built-in packages, User-defined packages, Creating and using packages, Importing packages, Static imports, Package naming conventions, Access modifiers and package visibility, Java API packages (java.lang, java.util, etc.).

Unit III: Exception Handling and Multithreading

Exception Handling: Introduction to exceptions, Types of exceptions, Checked exceptions, Unchecked exceptions, Try, catch, finally blocks, Throw keyword, Throws keyword, Creating custom exceptions, Nested try-catch blocks, Exception propagation, Commonly used exceptions (ArithmeticException, NullPointerException, etc.), Assertions,

Multithreading: Introduction to multithreading, Creating threads by extending Thread class, Creating threads by implementing Runnable interface, Thread life cycle, Thread methods (start, run, sleep, join, etc.), Synchronization, Inter-thread communication, Thread priorities, Daemon threads, Thread groups, Concurrency issues and solutions, Introduction to the java.util.concurrent package.

Unit IV: Collections and File I/O

Collections Framework: Introduction to collections, Advantages of collections over arrays, Collection interfaces (List, Set, Map), Collection classes (ArrayList, LinkedList, HashSet, TreeSet, HashMap), Iterating over collections (Iterator, ListIterator), Using Collections class methods, Generics in collections,

File I/O: Introduction to File I/O, File class, Reading and writing files, FileReader and FileWriter, BufferedReader and BufferedWriter, FileInputStream and FileOutputStream, ObjectInputStream and ObjectOutputStream, Serialization and deserialization, Handling file I/O exceptions, Working with directories, Random access files.

Unit V: Basic GUI, JDBC, and Servlets

GUI (Graphical User Interface): Introduction to GUI in Java, Basics of Swing, Creating a window using JFrame, Simple Swing components (JButton, JLabel, JTextField), Basic event handling (ActionListener),

Introduction to JavaFX

JDBC (Java Database Connectivity): Introduction to JDBC, Connecting to a database, Executing SQL queries (Statement), ResultSet, Handling SQL exceptions,

Servlets: Introduction to servlets, Servlet lifecycle, Creating a basic servlet, Handling HTTP requests and responses (GET and POST).Introduction to JSP

References

- *"Java Programming and Problem Solving" by Ernest K. Smith and Peter A. Darnell, Wiley, 1st Edition (2011)*
- *Java: How to Program" by Paul Deitel and Harvey Deitel, Pearson Publication, 11th Edition (2017)*
- *Head First Java" by Kathy Sierra and Bert Bates, O'Reilly Media, 2nd Edition (2005)*

- "Java: The Complete Reference" by Herbert Schildt, McGraw-Hill Education, 12th Edition (2021)

List of e-Learning Resources:

1. *Coursera - Java Programming and Software Engineering Fundamentals* by Duke University
2. *Udemy - Java Programming Masterclass for Software Developers* by Tim Buchalka
3. *edX - Introduction to Java Programming* by Microsoft
4. *Oracle Java Documentation*

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**

Bachelor of Technology (Computer Science and Engineering)

Semester-III

L-2 T-1 P-2 C-4

CSE130 P - Object oriented Programming in Java

Course Objectives

- To understand concepts of Object-Oriented Programming (OOP) and their implementation in Java.
- Learn to implement inheritance, interfaces, and packages to design modular and reusable code.
- To Explore exception handling and multithreading to write robust, concurrent Java applications.
- Apply the Java Collections Framework and file handling to manage data effectively.
- Create graphical user interfaces (GUI), database connectivity (JDBC), and web development (Servlets) to build interactive Java applications.

Course Outcomes (COs)

1. Understand and apply OOP principles such as encapsulation, inheritance, and polymorphism, and implement basic Java programs using control structures, arrays, and string handling..
2. Apply inheritance, interfaces, and packages to design modular and maintainable Java programs.
3. Apply exception handling and multithreading to develop robust applications that can handle concurrency and runtime errors effectively.
4. Apply Java Collections Framework and perform file input/output operations to manage and process data efficiently in Java applications.
5. Create interactive Java applications using basic GUI components, connect to databases using JDBC, and create web applications with Servlets.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	1	1	2	3	-	-	-	-	-	-	-	-	-	-
CO2	1	2	2	2	-	3	-	-	-	-	-	-	2	-	-
CO3	1	2	3	2	-	2	-	-	-	-	-	-	1	-	-
CO4	1	2	2	2	1	2	-	-	-	-	-	-	2	-	-
CO5	1	1	1	2	2	2	-	-	1	1	-	-	1	-	-

High-3 Medium-2 Low-1

1. Create a class Calculator with overloaded methods to calculate the sum of two integers, three integers, and two double values. The method should return the sum based on the type and number of parameters.
2. Write a program that validates a password based on rules (at least 8 characters, must contain an uppercase letter, a number, and a special character). If the password is invalid, provide specific feedback on why it failed.
3. Implement a warehouse inventory system using a single-dimensional array where each index represents a product type. Update quantities based on user input for adding and removing items from the inventory and handle invalid inputs.
4. Create a Movie class with private attributes: title, director, genre, and rating. Create an array of five Movie objects and use constructors to initialize those objects. Write a static method getMoviesByDirector() in the Main class that takes an array of Movie objects and a director's name as input, and returns a list of movies directed by that director. Write the necessary getters to return the attributes.
5. Create a base class Product with attributes like productID, productName, and price. Extend the class into Clothing with additional attributes like size and material. Further, create a subclass MenClothing with attributes like type (e.g., formal, casual). Create methods in each class to handle actions like adding a product, displaying product details, and calculating discounts.Implement a program that creates

different types of clothing items for men, showing how the inheritance chain works in action.

6. Create a base class Person with attributes name, age, and gender. Extend the class to Student, Teacher, and Staff. Each subclass should have its own attributes (e.g., Student can have grade and rollNumber, Teacher can have subject and salary, Staff can have department and designation).

   Implement methods in each subclass to display details and specific information related to their roles (e.g., getGrade() for students, getSubject() for teachers).

   Use a list to store various types of people in the school and display their details..

7. Create a package employee that contains classes Employee, Manager, and HR. The Employee class should contain details like name, salary, and designation, while Manager and HR should extend the Employee class with additional functionalities. Create another package payroll where salary calculations and payslips are generated.

8. Create a banking application that allows users to withdraw and deposit money. Implement exception handling for scenarios like:

   `InsufficientFundsException` when a user tries to withdraw more money than available in their account.

   `InvalidAmountException` when the user tries to deposit or withdraw a negative or zero amount. Demonstrate how to handle these custom exceptions and notify the user with appropriate messages.

9. Design a system where multiple users (threads) can book movie tickets simultaneously. Implement multithreading to simulate the booking process, ensuring that:

   Thread safety is maintained when accessing shared resources (e.g., the total number of available tickets).

   Use synchronization to prevent multiple users from booking the same seat at the same time.

10. Create a library management system using a **List** to store book objects where:

    - Each book has attributes like title, author, and genre.
    - Implement functions to add new books, search for books by author or genre, and display all available books.
    - Use an ArrayList to maintain the collection of books and implement sorting functionality to display books in alphabetical order.

11. Develop a system to manage student grades using a **Map** where:

    The student name (or ID) is the key, and their grade is the value.

    Implement features to add, remove, update, and retrieve student grades.

    Use the `HashMap` collection to store student data and implement operations like searching for students with the highest grade, average grade, etc.

12. Design an address book application where contacts (name, phone number, address) are stored in a file. Allow users to add, remove, update, and search for contacts. Use file I/O to save the contacts to a text file and load them when the program starts.

13. Design a login form using Swings where users enter their username and password. On successful login, show a welcome message; on failure, show an error message. Implement a registration form and validate if the username is already taken.

14. Develop a currency converter using JavaFX where users can select the input and output currencies and enter the amount to convert. Use combo boxes for currency selection and a button to perform the conversion. Display the converted amount in a label.

15. Create a servlet-based product management system where users can:
- Add new products (name, price, quantity) via an HTML form.
- List all products on a separate page.

References

- *"Java Programming and Problem Solving" by Ernest K. Smith and Peter A. Darnell, Wiley, 1st Edition (2011)*
- *Java: How to Program" by Paul Deitel and Harvey Deitel, Pearson Publication, 11th Edition (2017)*
- *Head First Java" by Kathy Sierra and Bert Bates, O'Reilly Media, 2nd Edition (2005)*
- "Java: The Complete Reference" by Herbert Schildt, McGraw-Hill Education, 12th Edition (2021)

List of e-Learning Resources:

- *Coursera - Java Programming and Software Engineering Fundamentals* by Duke University
- *Udemy - Java Programming Masterclass for Software Developers* by Tim Buchalka
- *edX - Introduction to Java Programming* by Microsoft
- *Oracle Java Documentation*

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**

MANDSAUR UNIVERSITY

Bachelor of Technology

Subject Code: CSE180 PR1

Semester : III[rd]

Subject Name: CSE180 PR1 -Relational database management systems

OBJECTIVES:

- Gain a good understanding of the architecture and functioning of Database Management Systems
- Understand the use of Structured Query Language (SQL) and its syntax.
- Apply Normalization techniques to normalize a database.
- Understand the need of transaction processing and learn techniques for controlling the Consequences of concurrent data access.

OUTCOMES:

- Describe basic concepts of database system
- Design a Data model and Schemas in RDBMS
- Competent in use of SQL
- Create the database using SQL
- Analyze functional dependencies for designing robust Database

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	-	-	-	-	-	-	-	-	-	2	2	-
CO2	2	1	2	1	3	-	-	-	-	-	-	-	1	3	-
CO3	2	1	2	1	2	-	-	-	-	-	-	-	1	2	-
CO4	3	3	-	-	1	-	-	-	-	-	-	-	2	3	-
CO5	1	1	2	1	-	-	-	-	-	-	-	-	1	2	-

Sr. No.	Aim
1	Delete duplicate rows from the table.
2	Display the alternate row from table..
3	Delete alternate row from table
4	Update multiple rows using a single update statement.
5	Find the third highest paid and third lowest paid salary.
6	Display the 3rd, 4th, 9th rows from the table.
7	Display the ename, which starts with j, k, l or m.
8	Show all employees who were hired the first half of the month.

9	Write a sql statement for rollback commit and save points.
10	Write a pl/sql for select, insert, update and delete statements.
11	Write a pl/sql block to delete a record. If delete operation is successful return 1 else return 0.
12	Display name, hire date of all employees using cursors.
13	Display details of first 5 highly paid employees using cursors.
14	Write a database trigger which fires if you try to insert, update, or delete after 7 'O' clock.
15	Write a database trigger, which acts just like primary key and does not allow duplicated values.
16	Create a database trigger, which performs the action of the on delete casssssscade.
17	Write a database trigger, which should not delete from emp table if the day is Sunday.

Text Books :

1. Korth, Silbertz,Sudarshan, "Fundamentals of Database System", McGraw Hill.
2. Elmasri, Navathe, "Fundamentals Of Database Systems", Pearson Educations.
3. Atul Kahate , " Introduction to Database Management System", Pearson Education.

Reference Books :

1. Date C J, "An Introduction To Database System", Pearson Educations
2. Rob, " Data Base System:Design Implementation & Management", Cengage Learning
3. Oracle 9i Database Administration Fundamental-I, Volume I, Oracle Press,TMH.

MANDSAUR UNIVERSITY

Bachelor of Technology

Subject Code: CSE180 TR1

Semester : III[rd]

Subject Name: CSE180 TR1 - Relational database management systems

OBJECTIVES:

- Gain a good understanding of the architecture and functioning of Database Management Systems
- Understand the use of Structured Query Language (SQL) and its syntax.
- Apply Normalization techniques to normalize a database.
- Understand the need of transaction processing and learn techniques for controlling the Consequences of concurrent data access.

OUTCOMES:

- Describe basic concepts of database system
- Design a Data model and Schemas in RDBMS
- Competent in use of SQL
- Create the database using SQL
- Analyze functional dependencies for designing robust Database

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	-	-	-	-	-	-	-	-	-	2	2	-
CO2	2	1	2	1	3	-	-	-	-	-	-	-	1	3	-
CO3	2	1	2	1	2	-	-	-	-	-	-	-	1	2	-
CO4	3	3	-	-	1	-	-	-	-	-	-	-	2	3	-
CO5	1	1	2	1	-	-	-	-	-	-	-	-	1	2	-

Contents:

Unit 1 :

Introduction to DBMS : Introduction to DBMS : Introduction to DBMS concepts and architecture, file system organization, advantages of DBMS, Data models, schemas and instances, Data dependency, functions of DBA, Entities and attributes, entity types, Key attributes, Relationships, ER data model: Entities and attributes, Entity types, Defining the E-R diagram, Concept of Generalization, Aggregation and Specialization.

Unit 2 :

Relational data models: Relational data models: Relational data models, Domains, tuples, attributes, relations, characteristics of relations, keys, key attributes of a relation, Relational database, schemas, Integrity constraints, intension and extension. Data vs. Information

Relational Query languages: SQL-DDL, DML, integrity constraints, Complex queries, various joins, indexing, triggers, Relationship Types Relational algebra and relational calculus, Relational algebra operations like select, Project, Join, Division, outer union. Logical Comparisons and Precedence Rules, Sorting Rows, Introduction to Functions

Unit 3 :

Database Design concepts, Database Design concepts, introduction to normalization, various normal forms, functional dependency, Decomposition, Dependency preservation and lossless join, problems with null valued and dangling tuples, multivalued dependencies. Introduction to query optimization, steps of optimization, heuristic based, cost estimation based optimization methods.

Unit 4 :

Transaction Processing Concepts: Transaction Processing Concepts: Transaction System, Testing of Serializability, Serializability of schedules, conflict & view serializable schedule, recoverability, Recovery from transaction failures. Log based recovery. Checkpoints deadlock handling. Concurrency. Control Techniques: - Concurrency Control, locking Techniques for concurrency control, time stamping protocols, o, multiple granularity. Multi version schemes, Recovery with concurrent transaction. Introduction to Distributed databases, data mining, data warehousing, Object Technology and DBMS, Comparative study of OODBMS Vs DBMS . Temporal, Deductive, Multimedia, Web & Mobile database.

Unit 5 :

Study of Relational Database Management Systems through Oracle/Postgres.

SQL/MySQL: Introduction to PL/SQL, Benefits of PL/SQL, Creating PL/SQL Blocks, Defining Variables and Datatypes, Using SQL in PL/SQL, Program Structures to Control Execution Flow, Using Composite Data Types, Using Cursors and Parameters, Exception Handling, Using and Managing Procedures, Using and Managing Functions, Using and Managing Packages, Getting the Best out of Packages, Improving PL/SQL Performance, Using and Managing Triggers, Recognizing and Managing Dependencies, Using the PL/SQL Compiler

Course Outcome:

- Understand database concepts and structures and query language
- Design the E R model and relational model.
- To analysis and build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS.
- Understand Functional Dependency and Functional Decomposition.
- Apply various Normalization techniques
- Evaluate PL/SQL programming using concept of Cursor Management, Error Handling, Package and Triggers
- Create various advanced SQL queries related to Transaction Processing & Locking using concept of Concurrency control.
- Understand query processing and techniques involved in query optimization.
- Understand the principles of storage structure and recovery management.

Text Books :

1. Korth, Silbertz,Sudarshan, “Fundamentals of Database System”, McGraw Hill.
2. Elmasri, Navathe, “Fundamentals Of Database Systems”, Pearson Educations.
3. Atul Kahate , “ Introduction to Database Management System”, Pearson Education.

Reference Books :

1. Date C J, “An Introduction To Database System”, Pearson Educations
2. Rob, “ Data Base System:Design Implementation & Management”, Cengage Learning
3. Oracle 9i Database Administration Fundamental-I, Volume I, Oracle Press,TMH.

Bachelor of Technology (Computer Science and Engineering)

Semester-III

L-2 T-1 P-2 C-4

PCCSE0100A: Data Structure

Course Objectives

1. To introduce students to various fundamental data structures such as arrays, linked lists, stacks, queues, trees, graphs, and hash tables.
2. To analyze the efficiency of data structures in terms of time complexity and space complexity. They understand concepts like Big O notation and learn to evaluate the performance of different data structures for different operations.
3. To provide hands-on experience to students in writing code to create and manipulate different data structures.

Course Outcomes (COs)

1. Understand the basic concepts and terminology related to data structures and concepts of array.
2. Create and manipulate linear data structures such as arrays, linked lists, stacks, and queues.
3. Create and traverse various types of trees and graph data structure.
4. Apply hashing and file structures to efficiently store and retrieve data.
5. Create and analyze various sorting and searching algorithms.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	-	-	1	-	-	-	-	-	2	-	2	1
CO2	3	3	1	-	-	-	-	1	-	1	-	1	3	1	1
CO3	3	2	3	1	1	-	-	-	-	1	-	1	3	1	1
CO4	3	-	-	3		-	-	1	-	-	-	1	3	1	1
CO5	3	3		1	2		1	-	-	1	-	2	3	1	1

High-3 Medium-2 Low-1

UNIT I: Introduction to Data Structures **7Hours**

Data types – Primitive and non-primitive, Types of Data Structures - Linear & Non Linear Data Structures. Algorithms Definition and concepts, Space and Time Complexity of Algorithms, Array- Representation of arrays, Operations on Arrays, Applications of arrays.

UNIT II: Linear Data Structure **11Hours**

Stack-Definitions & Concepts, Operations on Stacks, Applications of Stacks-Polish Expression, Reverse Polish Expression, Recursion, Tower of Hanoi, Queue: Representation of Queue, Operations on Queue, Circular Queue, PriorityQueue, Array representation of Priority Queue, Double Ended Queue, Applications of Queue. Linked List: Singly Linked List, Doubly Linked list, Circular linked list, Linked Implementation of Stack, Linked implementation of Queue, Applications of linked list.

UNIT III: Nonlinear Data Structure **11 Hours**

Tree - Definitions, Representation of binary tree, Tree traversal - In order, post order, preorder, Threaded binary tree, Binary search trees, Conversion of General Trees To Binary Trees, Applications of Trees, Some balanced tree mechanism; e.g. Heap, AVL trees; 2-3 trees; Red black tree; Multi-way search tree: B and B+ tree.
Graph: Graph Terminologies, Adjacency Matrices and List Representations of Graphs; Elementary Graph Operations: Depth First Search & Breadth first Search, Spanning Trees: Shortest path, Minimal spanning tree using graphs.

UNIT IV: Hashing and File Structure **7 Hours**

Hashing: The symbol table, Hashing Functions, Collision Resolution Techniques, File Structure: Concepts of fields, records and files, Sequential, Indexed and Relative/Random File Organization, Indexing structure for index files, hashing for direct files, Multi-Key file organization and access methods.

UNIT V: Sorting and Searching **9 Hours**

Sorting – Need for sorting, Types of sorting algorithm -Stable sorting Algorithm, Internal & External sorting algorithm , Outline and offline algorithm, Sorting Techniques- Bubble Sort, Selection Sort, Quick Sort, Merge Sort, Insertion sort, Radix Sort, Shell sort, Heap sort, Bucket sort.
Searching – Sequential Search and Binary Search

Practical(s)

1. Implement a C program to create array and perform following operations- Traversal, sum of array elements, Average of all values, find Maximum and Minimum value, Search given value, reverse the elements and count the frequency of elements.
2. Implement a Program to merge two sorted arrays into a single sorted array.
3. Implement a C Program to implement a stack using an array.
4. Implement a C Program to check if a given expression containing parentheses is balanced using a stack.
5. Implement a C Program to convert an infix expression to postfix notation using a stack.
6. Implement a C Program to implement multiple stacks in a single array.
   Implement a C Program to implement a queue using an array with the following operations- traverse, find the size of a queue, find the front and rear elements of a queue.
7. Implement a C Program to implement a circular queue using an array with traverse, insertion and deletion operations.
8. Implement a C Program to implement a double-ended queue using an array with traverse, insertion and deletion operations.
9. Implement a C Program to implement a priority queue using an array.
10. Implement a C Program to create single linked list with different operations like- insert (begin, end and position), delete (begin, end and position), search and travers.
11. Implement a C program to implement stack using linked list.
12. Implement a C program to implement queue using linked list.
13. Implement a C Program to create circular linked list with different operations like- insert (begin, end and position), delete (begin, end and position), search and travers.
14. Implement a C Program to create and perform operations on a doubly linked list.
15. Implement a C Program to create a binary tree by adding nodes dynamically.
16. Implement a C Program to traverse and print the elements of a binary tree (in-order, pre-order, post-order traversal).
17. Implement a C Program to perform basic operations on a binary search tree, such as search, insertion, and deletion.
18. Implement a C Program to represent a graph using adjacency matrix or adjacency list.
19. Implement a C Program to perform file handling operations.
20. Implement a C Program to perform bubble sort.
21. Implement a C Program to perform insertion sort.
22. Implement a C Program to perform selection sort.
23. Implement a C Program to perform Quick sort.
24. Implement a C Program to perform Merge sort.

Total: 45 Hours

Reference(s)

1. Introduction to algorithm", by T.H. Coreman, PHI Publication.
2. Data Structures, by Trembley and Sorenson, TMH Publications
3. Data structure and algorithm, by Pai, TMH Publications.
4. An Introduction to Data Structures with Applications, by Jean-Paul Tremblay & Paul G. Sorenson Publisher-Tata McGraw Hill (Text Book)

List of e-Learning Resources:

1. https://onlinecourses.swayam2.ac.in/cec23_cs09/preview

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**

Mandsaur University

Bachelor of Technology (Computer Science and Engineering)

Semester-III

L-2T-1P-0C-3

CSE663 TR1: Introduction to Machine Learning

Course Objectives

- To give understanding the fundamentals of machine learning, its history, types, and common challenges.
- To learn building and evaluating supervised learning models using techniques like linear regression, SVMs, decision trees, and ensemble methods.
- To explore unsupervised learning techniques to discover patterns, clusters, and associations in data.
- To delve into advanced topics such as ensemble methods, neural networks, deep learning, NLP, and reinforcement learning.
- To learn the principles of genetic algorithms and apply them to solve optimization problems in machine learning.

Course Outcomes (COs): Upon completion of this unit students will be able to:

1. Understand the basic concepts, terminologies, and workflow of machine learning
2. Apply supervised learning algorithms for regression and classification tasks
3. Apply unsupervised learning algorithms for clustering tasks.
4. Apply theadvanced deep learning and Neural network concepts.
5. Create a basic understanding of genetic algorithms and its applications.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	1	-	3	-	-	-	-	-	-	-	-	1	1	-
CO2	1	3	1	-	-	-	-	-	-	-	-	-	3	1	-
CO3	1	1	-	-	3	-	-	-	-	-	-	-	2	2	-
CO4	1	-	2	-	-	-	-	-	-	-	-	-	2	3	2
CO5	1	-	-	-	3	-	-	-	-	-	-	-	3	1	1

High-3 Medium-2 Low-1

UNIT I: Introduction to Machine Learning **9 Hours**

Introduction to Machine Learning,Definition and scope of machine learning,Historical developments and key milestones,Machine learning workflow and process.Issues in machine learning, Types of learning: supervised, unsupervised and Reinforcement learning, Concept learning.

UNIT II: Supervised Learning **9 Hours**

Supervised Learning,Linear Regression- Simple linear regression, Multiple linear regression, Support Vector Machines (SVM),Decision trees and Random Forests,Ensemble methods: Bagging and Boosting.Polynomial regression, Evaluation metrics: MSE, RMSE, MAE, confusion matrix, ROC curve, AUC,accuracy, precision, recall, F1-score,Cross-validation techniques, Bias-Variance tradeoff

UNIT III: Unsupervised Learning **9 Hours**

Unsupervised Learning,K-means clustering,Hierarchicalclustering,Dimensionality Reduction:Principal Component Analysis (PCA),t-Distributed Stochastic Neighbor Embedding (t-SNE), Association Rules: Apriori algorithm, Market basket analysis.

UNIT IV: Advanced Topics: **9 Hours**

Ensemble Methods: Bagging (Random Forest) , Boosting: (AdaBoost, Gradient Boosting Machines (GBM), XGBoost), Support Vector Machines (SVM): Concepts and hyperplanes, Kernel trick, Neural Networks and Deep Learning: Basics of neural networks, Introduction to deep learning and neural network architectures,

Natural Language Processing (NLP): Text preprocessing, Sentiment analysis, Introduction to word embeddings and transformer models, Reinforcement Learning

UNIT V: Genetic Algorithms (GAs): 9 Hours

Genetic Algorithms (GAs): Motivation, Representing Hypotheses, Genetic operators, fitness Function and Selection, Working of Genetic Algorithm, Case studies of Machine Learning data sets.

Total: 45 Hours

References

1. "Pattern Recognition and Machine Learning", by Christopher M. Bishop, Springer, 1st Edition, 2006
2. "Machine Learning: A Probabilistic Perspective", by Kevin P. Murphy, The MIT Press, 1st Edition, 2012.
3. "Machine Learning Yearning", by Andrew Ng, (Self-Published), (eBook), 2018 (updated edition)
4. "Deep Learning", by Ian Goodfellow, Yoshua Bengio, and Aaron Courville, The MIT Press, 1st Edition, 2016
5. "Introduction to Machine Learning with Python: A Guide for Data Scientists", by Andreas C. Müller and Sarah Guido, O'Reilly Media, 1st Edition, 2016
6. "Machine Learning: An Algorithmic Perspective", by Stephen Marsland, CRC Press, 2nd Edition, 2015
7. "Genetic Algorithms and Machine Learning for Programmers", by John P. Miller, Apress, 1st Edition, 2010
8. "Machine Learning: A Bayesian and Optimization Perspective", by Sergios Theodoridis, Academic press, 1st Edition, 2021

List of e-Learning Resources:

1. Introduction to Machine Learning - Course (nptel.ac.in)
2. https://www.coursera.org/

Subject Tr. Academic Coordinator HoD Sr. Faculty Nominated by DOAA

Mandsaur University

Bachelor of Technology (Computer Science and Engineering)

Semester-III

L-0 T-0 P-2 C-1

CSE663 PR1: Introduction to Machine Learning

Course Objectives

- To give understanding the fundamentals of machine learning, its history, types, and common challenges.
- To learn building and evaluating supervised learning models using techniques like linear regression, SVMs, decision trees, and ensemble methods.
- To explore unsupervised learning techniques to discover patterns, clusters, and associations in data.
- To delve into advanced topics such as ensemble methods, neural networks, deep learning, NLP, and reinforcement learning.
- To learn the principles of genetic algorithms and apply them to solve optimization problems in machine learning.

Course Outcomes (COs): Upon completion of this unit students will be able to:

1. Understand the basic concepts, terminologies, and workflow of machine learning
2. Apply supervised learning algorithms for regression and classification tasks
3. Apply unsupervised learning algorithms for clustering tasks.
4. Apply the advanced deep learning and Neural network concepts.
5. Create a basic understanding of genetic algorithms and its applications.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	1	1	-	3	-	-	-	-	-	-	-	-	1	1	-
CO2	1	3	1	-	-	-	-	-	-	-	-	-	3	1	-
CO3	1	1	-	-	3	-	-	-	-	-	-	-	2	2	-
CO4	1	-	2	-	-	-	-	-	-	-	-	-	2	3	2
CO5	1	-	-	-	3	-	-	-	-	-	-	-	3	1	1

High-3 Medium-2 Low-1

1. Implement a linear regression model to predict housing prices based on features like area, number of rooms, etc.
2. Develop a program to analyze and predict stock market trends using linear regression.
3. Build a spam email classifier using logistic regression to classify emails as spam or non-spam.
4. Create a program to predict customer churn for a subscription-based service using logistic regression.
5. Construct a decision tree model to classify different species of flowers based on their features.
6. Develop a program that predicts whether a credit card applicant is likely to default or not using decision trees.
7. Build a random forest classifier to classify images into different categories.
8. Develop a program that predicts customer satisfaction using gradient boosting and ensemble learning.
9. Implement an SVM model for sentiment analysis to classify movie reviews as positive or negative.
10. Build a program that detects anomalies in network traffic using SVM.

11. Develop a program to segment customer data into different groups using K-means clustering.
12. Implement hierarchical clustering to analyze and group documents based on their similarity.
13. Apply t-SNE (t-Distributed Stochastic Neighbor Embedding) to visualize high-dimensional data in a lower-dimensional space.
14. Build a convolutional neural network (CNN) for image recognition, such as classifying handwritten digits.
15. Develop a recurrent neural network (RNN) for sentiment analysis on text data.

Total: 30 Hours

References

1. "Pattern Recognition and Machine Learning", by Christopher M. Bishop, Springer, 1st Edition, 2006
2. "Machine Learning: A Probabilistic Perspective", by Kevin P. Murphy, The MIT Press, 1st Edition, 2012.
3. "Machine Learning Yearning", by Andrew Ng, (Self-Published), (eBook), 2018 (updated edition)
4. "Deep Learning", by Ian Goodfellow, Yoshua Bengio, and Aaron Courville, The MIT Press, 1st Edition, 2016
5. "Introduction to Machine Learning with Python: A Guide for Data Scientists", by Andreas C. Müller and Sarah Guido, O'Reilly Media, 1st Edition, 2016
6. "Machine Learning: An Algorithmic Perspective", by Stephen Marsland, CRC Press, 2nd Edition, 2015
7. "Genetic Algorithms and Machine Learning for Programmers", by John P. Miller, Apress, 1st Edition, 2010
8. "Machine Learning: A Bayesian and Optimization Perspective", by Sergios Theodoridis, Academic press, 1st Edition, 2021

List of e-Learning Resources:

1. Introduction to Machine Learning - Course (nptel.ac.in)
2. https://www.coursera.org/

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**

Mandsaur University

Bachelor of Technology (Computer Science and Engineering)

Semester-III

MAT220 PR1: Discrete Mathematics

Course Objective:

L-0 T-0 P-2 C-1

- To introduce the concepts of sets, relations, and functions.
- To perform the operations associated with sets, functions, and relations.
- To introduce generating functions and recurrence relations.
- To provide students with an introduction to number theory.
- To introduce generating functions and recurrence relations.

.

Course Outcomes (COs): Upon completion of this unit students will be able to:

1.Understand sets, relations, functions and discrete structure and apply the concept to computer science

2. To develop understanding of Logic Sets and Functions.

3. Analyze the concepts of set theory to apply t in group , field and ring theory

4.Analyze the concepts of Recurrence Relation and find solution.

5.Apply the concept of number theory and apply it in computer programming

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	-	-	-	1	-	1		2	-	1		2
CO2	-	1	-	1	3	-	-	1	2	-	-	1	-	2	-
CO3	2	-	3	-	-	-	3	1	-	-	-	-	-	1	-
CO4	-	3	2	2	1	3	-	-	-	1	-	-	1	-	-
CO5	1	-	-	1	2	1	-		2	-	3	1	-	1	1

Practical's list :

1 Create a python program to perform following set operation

- **Creation of sets**
- **Union**
- **Intersection**
- **Difference**
- **Symmetric Difference**
- **Set Cardinality**
- **Powerset**
- **Cartesian Product**
- **Set Complement**

2.Find the demonstrate principal of Pigeonhole Principle in python programming.

3.Create a python program including a function inclusion_exclusion that takes a list of sets as input. It uses bitwise operations to generate all possible subsets of the sets and calculates the intersection of each subset. By applying the inclusion-exclusion formula, it computes the sum of the lengths of the intersections with alternating signs. Finally, it subtracts this sum from the sum of the lengths of the original sets to obtain the size of the union.

4.Build a following that performs basic operations related to First-Order Logic.

5.Develop Python programs related to Universal and Existential Quantifiers in First-Order Logic with following operations:

Evaluating Universal Quantifiers:

Evaluating Existential Quantifiers
Combining Universal and Existential Quantifiers:
Quantifier Negation:

6.Creating an algebraic structure Group and some function for checking associativity, identity and inverse properties for python programming.

7. Find to the algebraic structure Ring and two function for checking Associative (Addition), Associative (Multiplication).

8.Implement python program for algebraic structure field, involves defining the operations of addition, subtraction, multiplication, and division, along with satisfying several properties such as associativity, commutativity, and the existence of identity and inverses.

9.Apply to the Python program that can solve linear recurrence relations with constant coefficients using matrix exponentiation.

10.Find all prime numbers in a given range in Python program

11.Find the greatest common divisor (GCD) of two numbers to Python

12.Build the factorial of given number using recursion.

13.Find the number given is Armstrong number or not using recursion

14.Develop Python program to calculate minimum weight spanning tree for the given graph.

15.Build to the Python program to calculate shortest path of given graph.

Total: 30 Hours

References

9. C.L.Liu "Elements of Discrete Mathematics" 4th ed.,Tata Mc Graw-Hill Edition.2017
10. Kenneth H. Rosen, "Discrete Mathematics and its applications"7th ed, McGraw Hill.1 July 2017
11. D. M. Burton, Elementary Number Theory, 7th ed., Tata McGraw-Hill, New Delhi, 2012.

List of e-Learning Resources:

1. https://nptel.ac.in/
2. https://www.coursera.org/
3. https://www.udemy.com/course/introduction-to-number-theory/

Subject Tr. Academic Coordinator HoD Sr. Faculty Nominated by DOAA

Bachelor of Technology (Computer Science and Engineering)
Semester-III

L-2 T-1 P-0 C-3

MAT220 TR1: Discrete Mathematics

Course Objectives

- To introduce the concepts of sets, relations, and functions
- To introduce the concepts of mathematical logic
- To perform the operations associated with sets, functions, and relations
- To introduce generating functions and recurrence relations
- To provide students with an introduction to graph theory

Course Outcomes (COs)

1. Understand sets, relations, functions and discrete structure and apply the concept to computer science.
2. Understand of Logic Sets and Functions
3. Analyze the concepts of set theory to apply t in group , field and ring theory
4. Analyze the concepts of Recurrence Relation and find solution
5. Apply the concept of graph theory to evaluate the shortest path

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation)

CO/PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	-	-	-	1	-	-		-	-	-	2	-	-
CO2	3	2	1	1	2	1	-	-	-	1	1	-	2	1	-
CO3	2	3	2	2	1	-	-	-	-	-	-	1	2	-	-
CO4	1	3	2	2	1	1	-	-	-	-	-	1	2	-	-
CO5	1	3	2	2	2	1	-	-	-	-	-	-	2	-	-

High-3 Medium-2 Low-1

Unit I: Set Theory, Theorem proving Techniques, Recurrence relation

12 Hours

Introduction to Sets, Finite and Infinite Sets, Unaccountably Infinite Sets. Introduction to Functions and relations, Properties of Binary relations, Partial Ordering Relations. Pigeonhole Principle, Mathematical Induction, Principle of Inclusion and Exclusion,

Unit II: Logic set and Theory

12 Hours

Proposition, First order logic, Basic logical operation, truth tables, tautologies, Contradictions, Algebra of Proposition, logical implications, logical equivalence, predicates, Normal Forms, Universal and existential quantifiers.

Unit III: Algebraic structures 12 Hours

Definition, Properties, types: Semi Groups, Monoid, Groups, Abelian group, properties of groups, Subgroup, cyclic groups, Cosets, factor group, Permutation groups, Normal subgroup, Homomorphism and isomorphism of Groups, example and standard results. Rings and Fields: definition and standard results.

Unit IV: Recurrence Relation 12 Hours

Recurrence Relation, Linear recurrence relations with constant coefficients, Homogeneous solutions, Particular solutions, Total solutions, Generating functions, Solution by method of generating functions

Unit V: Graph Theory **12 Hours**

Introduction of graphs, pseudographs, complete graphs, bi-partite graphs, isomorphism of graphs, paths And circuits, Eulerian circuits, Hamiltonian cycles, the adjacency matrix, weighted graph, travelling Salesman's problem, shortest path, Dijkstra's algorithm, Floyd-Warshall algorithm. Trees, Properties of Trees, Distance and Centres in a tree, Spanning trees of a graph and weighted graph.

Total: 60 Hours

Reference(s):

1 C.L.Liu, "Elements of Discrete Mathematics" Tata Mc Graw-Hill Edition.
2 Kenneth H. Rosen, "Discrete Mathematics and its applications", McGraw Hill.
3 Deo, Narsingh, "Graph Theory With application to Engineering and Computer.Science", PHI.
4 D. M. Burton, Elementary Number Theory, 7th ed., Tata McGraw-Hill, New Delhi, 2012.
5 Nar Singh Deo, Graph Theory, PHI

List of e-Learning Resources:

4. https://nptel.ac.in/
5. https://www.coursera.org/
6. https://www.youtube.com/watch?v=xlUFkMKSB3Y&list=PL379707A07F33C18B
7. https://www.youtube.com/watch?v=y-S7FyPD-uA&list=PL25JkDhpbDsfPBy3Cci5Lp_y-7qFgDSHt
8. https://www.udemy.com/course/introduction-to-number-theory/
9. https://www.udemy.com/course/number-theory-modular-arithmetic/

Subject Tr. **Academic Coordinator** **HoD** **Sr. Faculty Nominated by DOAA**

B.Tech(Computer Science Engineering)

MUC 010 Quantitative Aptitude- I
Sem –III (CSE-Plain, AI, BCT)

L-4 T-0 P-0 C-4

Course Objectives

- To cater to the needs of outgoing students.
- To prepare students for various examinations and campus interviews.
- To acquaint students with frequently asked patterns in quantitative aptitude and logical reasoning.
- To provide students with strategies and techniques to excel in these areas.
- To give exposure on Logical reasoning.
- To create deeper understanding in data interpretation and progression

Course Outcomes:

1. Understand the concepts of quantitative ability.
2. Understand the basic concepts of logical reasoning Skills.
3. Apply the concept of logical reasoning.
4. Analyze campus placements aptitude papers covering Quantitative Ability, Logical Reasoning Ability and compete in various competitive exams like CAT, CMAT, GATE, GRE, GATE, UPSC, GPSC etc..
5. Evaluate the concept of Logical and Verbal Reasoning.

Articulation Matrix

(Program Articulation Matrix is formed by the strength of correlation of COs with POs and PSOs. The strength of correlation is indicated as 3 for substantial (high), 2 for moderate (medium) correlation, and 1 for slight (low) correlation).

CO/PO/PSO	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3	2	1	1	-	-	-	-	-	-	-	-	1	-	-
CO2	3	2	1	1	2	-	-	-	-	-	1	-	1	1	-
CO3	2	3	2	2	1	1	-	-	-	-	1	1	2	1	-
CO4	1	3	2	2	1	1	-	-	-	1	1	1	1	2	1
CO5	1	3	2	2	2	1	-	-	-	1	1	1	2	1	1

High-3 Medium-2 Low-1

Unit- I: Quantitative Ability-I **12 Hours**

Number Systems, LCM and HCF, Decimal Fractions, Simplification, Square Roots and Cube Roots, Average, Problems on Ages, Surds & Indices, Percentages, Problems on Numbers

Unit- II: Quantitative Aptitude-Number Theory **12 Hours**

Place value, Face Value, Divisibility test, Prime and Co- prime numbers, number series based on basic formula, Fractions and factors

Unit- III: Basic Data Interpretation **12 Hours**

Tables, Column, Graphs, Bar Graphs, Line Charts, Pie Char,Venn Diagrams

Unit- IV: Logical Reasoning level -I **12 Hours**

Analogy, Blood Relation, Directional Sense, Number and Letter Series, Coding – Decoding

Unit- V: Verbal Reasoning **12 Hours**

Verification of Truth, Logical sequence of Words, Character Puzzels, Series Competition

Total: 60 Hours

Reference Books:

1. Aggarwal, R. S. (2022). A Modern Approach to Verbal & Non Verbal Reasoning. *S. Chand & Company Pvt Limited* .
2. Aggarwal, R. S. (2012). Quantitative Aptitude for Competitive Examinations. *S. Chand & Company Pvt Limited (Unit II, III)*.
3. Praveen, R. V. (2016). *Quantitative Aptitude and Reasoning*. PHI Learning Pvt. Ltd..
4. Allwein, G., & Barwise, J. (Eds.). (1996). *Logical reasoning with diagrams*. Oxford University Press.
5. Sharma, M., & Basu, S. (2024). BELL THE "CAT". *Managing India: The Idea of IIMs and its Changing Contexts*.

List of e-Learning Resources:

1. https://prepinsta.com/
2. https://www.indiabix.com/
3. https://www.javatpoint.com/