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)

- Understand and apply OOP principles such as encapsulation, inheritance, and polymorphism, and implement basic Java programs using control structures, arrays, and string handling..
- Apply inheritance, interfaces, and packages to design modular and maintainable Java programs.
- Apply exception handling and multithreading to develop robust applications that can handle concurrency and runtime errors effectively.
- Apply Java Collections Framework and perform file input/output operations to manage and process data efficiently in Java applications.
- 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:

- *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**

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)

- Understand and apply OOP principles such as encapsulation, inheritance, and polymorphism, and implement basic Java programs using control structures, arrays, and string handling..
- Apply inheritance, interfaces, and packages to design modular and maintainable Java programs.
- Apply exception handling and multithreading to develop robust applications that can handle concurrency and runtime errors effectively.
- Apply Java Collections Framework and perform file input/output operations to manage and process data efficiently in Java applications.
- 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

- 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.
- 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.
- 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.
- 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.
- 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.

- 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..

- 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.

- 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.

- 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.

- 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.

- 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.

- 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**

PRACTICAL

1 Write a C program to generate the first n terms of the Fibonacci sequence.

2 Write a C program to Check whether given number is Armstrong Number or Not.

3 Write a C program to evaluate algebraic expression (ax+b)/(ax-b).

4 Write a C program to find the roots of a quadratic equation.

5 Write a C program perform arithmetic operations using switch statement.

6 Write a C program to find factorial of a given integer using recursive function.

7 Write C program to find GCD of two integers by using recursive function.

8 Write a C Program to Sort the Array in an Ascending Order.

9 Write a C Program to find whether given matrix is symmetric or not.

10 Write a C program that uses functions to delete n Characters from a given position in a given string.

11 Write a C program using user defined functions to determine whether the given string is palindrome or not.

12 Write a C program that displays the position or index in the main string S where the sub string T begins, or - 1 if S doesn't contain T.

13 Write a C program to find the length of the string using Pointer.

14 Write a C program that uses functions and structures to perform the following operations:

i) Reading a complex number ii) Writing a complex number

iii) Addition of two complex numbers iv) Multiplication of two complex numbers

15 Write a C program to display the contents of a file.

Text Books:

E. Balaguruswamy, “Programming in ANSI C”, 8th Edition, 2019, McGraw Hill Education, ISBN: 978-93-5316-513-0.

Reference Books:

1. Pradip Dey, Manas Ghosh, “Programming in C”, 2nd Edition, 2018, Oxford University Press, ISBN: 978-01-9949-147-6.
2. Kernighan B.W and Dennis M. Ritchie, “The C Programming Language”, 2nd Edition, 2015, Pearson

Education India, ISBN: 978-93-3254-944-9.
3. Yashavant P. Kanetkar, “Let Us C”, 16th Edition, 2019, BPB Publications, ISBN: 978- 93-8728-449-4.
4. Jacqueline A Jones and Keith Harrow, “Problem Solving with C”, Pearson Education. ISBN: 978-93-325-3800-9.
5. Dr. Guruprasad Nagraj, “C Programming for Problem Solving”, Himalaya Publishing House. ISBN-978-93-5299-361-1.

Weblinks and Video Lectures (e-Resources):

1. http://elearning.vtu.ac.in/econtent/courses/video/BS/14CPL16.html
2. https://nptel.ac.in/courses/106/105/106105171