

OBJECT ORIENTATED PROGRAMMING THROUGH JAVA

Semester	Course Code	Course Title	Hours	Credits
IV	C4	OBJECT ORIENTATED PROGRAMMING THROUGH JAVA	60	3

Objectives:

To introduce the fundamental concepts of Object-Oriented programming and to design & implement object oriented programming concepts in Java.

Course Learning Outcomes: At the end of this course student will:

1. Understand the benefits of a well-structured program
2. Understand different computer programming paradigms
3. Understand underlying principles of Object-Oriented Programming in Java
4. Develop problem-solving and programming skills using OOP concepts
5. Develop the ability to solve real-world problems through software development in high-level programming language like Java

UNIT – I

Introduction to Java: Features of Java, The Java virtual Machine, Parts of Java

Naming Conventions and Data Types: Naming Conventions in Java, Data Types in Java, Literals

Operators in Java: Operators, Priority of Operators

Control Statements in Java: if... else Statement, do... while Statement, while Loop, for Loop, switch Statement, break Statement, continue Statement, return Statement

Input and Output: Accepting Input from the Keyboard, Reading Input with Java.util.ScannerClass, Displaying Output with System.out.printf(), Displaying Formatted Output with String.format()

UNIT – II

Arrays: Types of Arrays, Three Dimensional Arrays (3D array), arrayname.length, Command Line Arguments

Strings: Creating Strings, String Class Methods, String Comparison, Immutability of Strings

Introduction to OOPs: Problems in Procedure Oriented Approach, Features of Object- Oriented Programming System (OOPS)

Classes and Objects: Object Creation, Initializing the Instance Variables, Access Specifiers, Constructors

Methods in Java: Method Header or Method Prototype, Method Body, Understanding Methods, Static Methods, Static Block, The keyword 'this', Instance Methods, Passing Primitive Data Types to Methods, Passing Objects to Methods, Passing Arrays to Methods, Recursion, Factory Methods

UNIT – III

Inheritance: Inheritance, The keyword 'super', The Protected Specifier, Types of Inheritance

Polymorphism: Polymorphism with Variables, Polymorphism using Methods, Polymorphism with Static Methods, Polymorphism with Private Methods, Polymorphism with Final Methods, final Class

Type Casting: Types of Data Types, Casting Primitive Data Types, Casting Referenced Data Types, The Object Class

Abstract Classes: Abstract Method and Abstract Class

Interfaces: Interface, Multiple Inheritance using Interfaces

Packages: Package, Different Types of Packages, The JAR Files, Interfaces in a Package, Creating Sub Package in a Package, Access Specifiers in Java, Creating APIDocument

UNIT – IV

Exception Handling: Errors in Java Program, Exceptions, throws Clause, throw Clause, Types of Exceptions, Re – throwing an Exception

Streams: Stream, Creating a File using File Output Stream, Reading Data from a File using File Input Stream, Creating a File using File Writer, Reading a File using File Reader, Zipping and Unzipping Files, Serialization of Objects, Counting Number of Characters in a File, File Copy, File Class

UNIT – V

Threads: Single Tasking, Multi Tasking, Uses of Threads, Creating a Thread and Running it, Terminating the Thread, Single Tasking Using a Thread, Multi Tasking Using Threads, Multiple Threads Acting on Single Object, Thread Class Methods, Deadlock of Threads, Thread Communication, Thread Priorities, thread Group, Daemon Threads, Applications of Threads, Thread Life Cycle.

Applets: Creating an Applet, Uses of Applets, <APPLET> tag, A Simple Applet, An Applet with Swing Components, Animation in Applets, A Simple Game with an Applet, Applet Parameters

Practicals:

1. Write a program to read *Student Name, Reg.No, Marks[5]* and calculate *Total, Percentage, Result*. Display all the details of students
2. Write a program to perform the following String Operations
 - a. Read a string
 - b. Find out whether there is a given substring or not
 - c. Compare existing string by another string and display status
 - d. Replace existing string character with another character
 - e. Count number of words in a string
3. Java program to implements Addition and Multiplication of two N X N matrices.
4. Java program to demonstrate the use of Constructor.
5. Calculate area of the following shapes using method overloading.
 - a. Triangle Rectangle Circle Square
6. Implement inheritance between *Person (Aadhar, Surname, Name, DOB, and Age)* and *Student (Admission Number, College, Course, Year)* classes where *ReadData()*, *DisplayData()* are overriding methods.
7. Java program for implementing Interfaces
8. Java program on Multiple Inheritance.
9. Java program for to display *Serial Number from 1 to N* by creating two Threads
10. Java program to demonstrate the following exception handlings
 - a. Divided by Zero Array Index Out of Bound File Not Found
11. Create an Applet to display different shapes such as Circle, Oval, Rectangle, Square and Triangle.