

**III Semester**  
**Course 5: Object Oriented Programming using Java**  
Credits -3

---

**Course Objectives**

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

**Course Outcomes**

Upon successful completion of the course, a student will be able to:

1. Understand the basic concepts of Object-Oriented Programming and Java Program Constructs
2. Implement classes and objects and analyze Inheritance and Dynamic Method Dispatch
3. Demonstrate various classes in different packages and can design own packages
4. Manage Exceptions and Apply Threads
5. Create GUI screens along with event handling

**UNIT-I**

**OOPs Concepts and Java Programming:** Introduction to Object-Oriented concepts, procedural and object-oriented programming paradigm

**Java programming:** An Overview of Java, Java Environment, Data types, Variables, constants, scope and life time of variables, operators, type conversion and casting, Accepting Input from the Keyboard, Reading Input with Java.util.Scanner Class, Displaying Output with System.out.printf(), Displaying Formatted Output with String.format(), Control Statements

**UNIT-II**

Arrays, Command Line Arguments, Strings-String Class Methods

**Classes & Objects:** Creating Classes, declaring objects, Methods, parameter passing, static fields and methods, Constructors, and 'this' keyword, overloading methods and access

**Inheritance:** Inheritance hierarchies, super and subclasses, member access rules, 'super' keyword, preventing inheritance: final classes and methods, the object class and its methods; **Polymorphism:** Dynamic binding, method overriding, abstract classes and methods;

**UNIT-III**

**Interface:** Interfaces VS Abstract classes, defining an interface, implement interfaces, accessing implementations through interface references, extending interface;

**Packages:** Defining, creating and accessing a package, understanding CLASSPATH, importing packages.

**Exception Handling:** Benefits of exception handling, the classification of exceptions, exception hierarchy, checked exceptions and unchecked exceptions, usage of try, catch, throw, throws and finally, rethrowing exceptions, exception specification, built in exceptions, creating own exceptions sub classes.

**UNIT-IV**

---

**Multithreading:** Differences between multiple processes and multiple threads, thread states, thread life cycle, creating threads, interrupting threads, thread priorities, synchronizing threads, inter thread communication.

**Stream based I/O (java.io)** – The Stream classes-Byte streams and Character streams, Reading console Input and Writing Console Output, File class, Reading and writing Files, The Console class, Serialization

#### **UNIT-V**

**GUI Programming with Swing-** Introduction, MVC architecture, components, containers. Understanding Layout Managers - Flow Layout, Border Layout, Grid Layout, Card Layout, GridBag Layout.

**Event Handling-** The Delegation event model- Events, Event sources, Event Listeners, Event classes, Handling mouse and keyboard events, Adapter classes, Inner classes, Anonymous Inner classes.

#### **Text Books:**

1. Java The complete reference, 9th edition, Herbert Schildt, McGraw Hill.
2. Understanding Object-Oriented Programming with Java, updated edition, T. Budd, Pearson Education.

#### **Reference Books**

1. Cay S. Horstmann, “Core Java Fundamentals”, Volume 1, 11 th Edition, Prentice Hall, 2018.
2. Paul Deitel, Harvey Deitel, “Java SE 8 for programmers”, 3rd Edition, Pearson, 2015.
3. S. Malhotra, S. Chudhary, Programming in Java, 2nd edition, Oxford Univ. Press.

#### **SUGGESTED CO-CURRICULAR ACTIVITIES & EVALUATION METHODS:**

**Unit 1: Activity:** Quiz on Object-Oriented Programming Concepts and Java Constructs

**Evaluation Method:** Quiz Performance and Knowledge Retention

**Unit 2: Activity:** Object-Oriented Programming Assignment: Class Implementation

**Evaluation Method:** Assignment Completion and Correctness

**Unit 3: Activity:** Hands-on Lab Activity: Creating and Using Custom Java Packages

**Evaluation Method:** Lab Performance and Correctness of Code Implementation

**Unit 4: Activity:** Case Study Discussion on where multi-threading is crucial

**Evaluation Method:** Critical thinking, problem-solving, and presentation skills.

**Unit 5: Activity:** GUI design contest using Java Swings

**Evaluation Method:** GUI design, Visual appearance and user friendliness, usability, and adherence to event handling principles.

**III Semester**  
**Course 5: Object Oriented Programming using Java Lab**  
Credits -1

---

**List of Experiments**

1. Write a Java program to print Fibonacci series using for loop.
2. Write a Java program to calculate multiplication of 2 matrices.
3. Create a class Rectangle. The class has attributes length and width. It should have methods that calculate the perimeter and area of the rectangle. It should have read Attributes method to read length and width from user.
4. Write a Java program that implements method overloading.
5. Write a Java program for sorting a given list of names in ascending order.
6. Write a Java program that displays the number of characters, lines and words in a text file.
7. Write a Java program to implement various types of inheritance
  - i. Single
  - ii. Multi-Level
  - iii. Hierarchical
  - iv. Hybrid
8. Write a java program to implement runtime polymorphism.
9. Write a Java program which accepts withdraw amount from the user and throws an exception “In Sufficient Funds” when withdraw amount more than available amount.
10. Write a Java program to create three threads and that displays “good morning”, for every one second, “hello” for every 2 seconds and “welcome” for every 3 seconds by using extending Thread class.
11. Write a Java program that creates three threads. First thread displays “OOPS”, the second thread displays “Through” and the third thread Displays “JAVA” by using Runnable interface.
12. Implement a Java program for handling mouse events when the mouse entered, exited, clicked, pressed, released, dragged and moved in the client area.
13. Implement a Java program for handling key events when the key board is pressed, released, typed.
14. Write a Java swing program that reads two numbers from two separate text fields and display sum of two numbers in third text field when button “add” is pressed.
15. Write a Java program to design student registration form using Swing Controls. The form which having the following fields and button SAVE

Form Fields are: Name, RNO, Mailid, Gender, Branch, Address.

---