Object Oriented Programming Using Java Sem-3

Unit 1: Introduction to Java

1. What are the basic features of Java?

2. Explain the Java program structure with an example.

3. What is the JVM? How is it different from JRE and JDK?

4. Explain data types and type casting in Java.

5. Write a Java program to demonstrate command-line arguments.

Unit 2: Control Structures and Arrays

1. Differentiate between `while`, `do-while`, and `for` loops in Java with examples.

2. Write a program to find whether a given number is prime or not.

3. Explain different types of arrays in Java with examples.

4. Write a Java program to sort an array in ascending order.

Unit 3: Classes, Objects, and Methods

1. Explain how to create classes and objects in Java with a program.

2. What is constructor overloading? Write a program to demonstrate it.

3. What is the difference between static and non-static members?

4. Explain the concept of `this` keyword with an example.

5. What is garbage collection in Java?

Unit 4: Inheritance and Polymorphism

1. What is inheritance? Explain types of inheritance supported in Java.

2. Explain method overriding and dynamic method dispatch with example.

3. Define `final`, `super`, and `abstract` keywords with examples.

4. What is polymorphism? Explain compile-time and run-time polymorphism.

5. Differentiate between overloading and overriding.

Unit 5: Interfaces, Packages, and Exception Handling

1. Explain interfaces in Java with a suitable example.

2. Differentiate between abstract class and interface.

3. What are packages? How are they used in Java?

4. What is exception handling? Explain `try`, `catch`, `finally` with an example.

5. Write a program to handle `ArithmeticException` and `ArrayIndexOutOfBoundsException`.

◆ Frequently Asked 10 Marks Questions

* Explain the object-oriented principles in Java with examples.

* Write a Java program using inheritance that calculates the salary of different types of employees.

* Discuss exception handling with user-defined exceptions in Java.

* Write a program to demonstrate the concept of interfaces and multiple inheritance.

* Describe the lifecycle of a Java program with execution flow.