Roll No. 25420 MCA 020

M.C.A./M.Sc. Semester I Examination 2025-26

Computer Science

CS 204 : Objective Oriented Programming

Time : Three hours **Max. Marks : 70**

(Write your Roll No. at the top immediately on the receipt of this question paper)

Note: Question 1 is compulsory; attempt any four questions from the remaining six questions.

1. (a) What will be the output of the following program? Explain briefly. [2]

```
class Test extends Thread {
    public void run() { System.out.print("Running ");
    }
    public static void main(String[] args) { Test t = new Test(); t.start();   System.out.print("Main ");
    }
}
```

(b) What is the output of the following code snippet? [1]

```
int x = 10; int y = ++x * 10 + x--;
System.out.println(y);
```

(c) Complete the code to create a thread using the Runnable interface: [1]

```
class MyRunnable _______________ {
    public void run() {
        System.out.println("Thread executing");
    }
}
```

(d) What will be the result of the following code? [2]

```
String s1 = "Code"; String s2 = "Code";String s3 = s1;
System.out.println(s1 == s3); System.out.println(s1 == "Code");
```

(e) What will be the result of the following code? [2]

```
class Animal {
    void sound() { System.out.print("Animal "); }
}
class Dog extends Animal {
    void sound() { System.out.print("Dog "); }
}
Animal a = new Dog();
a.sound();
```

(f) What will be the result of the following code? [2]

```
int[] arr = new int[3]; arr[0] = 5; arr[1] = arr[0] * 2;
System.out.println(arr[1]);
```

(g) In Java, arrays are _____ objects stored in heap memory. [1]

(h) In Java, _____ is the mechanism where a subclass provides a specific implementation of a method already provided by its parent class. [1]

(i) The _____ method of the Object class is called by the garbage collector before an object is garbage collected. [1]

(j) The _____ access modifier restricts access to within the same package only. [1]

2. (a) Design a Java class named `Person` with three attributes: a `private name`, a `protected age`, and a `public address`. Write a driver program to illustrate how these access modifiers control the visibility and accessibility of each field from within the main method. [6]

(b) What are the Buzzwords in Java? Explain each buzzword with a suitable example. [8]

3. (a) Compare and contrast primitive data types with reference types in Java, discussing their memory allocation, default values, and behavior when passed to methods. Provide examples to illustrate the differences. [5]

P.T.O.

(b) Compare and contrast the four types of nested classes in Java: static nested class, inner class, local class, and anonymous class. Explain the access rules, memory implications, and typical use cases for each type. [5]

(c) What are the key features of the Object-Oriented concept? Explain each feature with an appropriate example. [4]

4. (a) Create a Java program that implements compile-time polymorphism using method overloading. Explain how it differs from runtime polymorphism and how the compiler resolves method calls. [5]

(b) Implement a Java program with both default and parameterized constructors. Explain what happens when no constructor is explicitly defined, and how constructors differ from regular methods. [4]

(c) Write a Java program to demonstrate the difference between string literals and string objects. Create two string variables: one using a string literal and the other using the new keyword. Compare them using the = = operator and explain the result. [5]

5. (a) With the help of a Java program, illustrate the sequence in which different initialization components of a class are executed. Your program must demonstrate:

(I) A static initializer block that displays a message upon class loading.

(II) An instance initializer block that displays a message during each object instantiation.

(III) A method named `initialize()` that displays a message when invoked.

(IV) A constructor that invokes the `initialize()` method and displays a message upon object creation.

Instantiate two objects of this class and explain the precise order in which the static block, instance block, method, and constructor are executed—first during class loading and then during the creation of each object. [9]

(b) Write a Java program that uses an abstract class with both abstract and concrete methods. Explain why abstract classes are used and how they differ from interfaces. [5]

6. (a) Demonstrate exception handling using `try-catch-finally` blocks with a Java program that throws an `ArithmeticException`. Explain the flow of execution when an exception occurs. [6]

(b) Write a Java program to implement a `Runnable` interface and start a thread. Explain the thread life cycle and the role of the `start()` method. [4]

(c) Create a Java program to read data from a file using `FileReader`. Explain the difference between byte streams and character streams, and when to use each. [4]

7. (a) Explain the architecture and core interfaces of the Java Collections Framework. Discuss the purpose and characteristics of the primary collection types—lists, sets, maps, and queues—and provide examples of their typical usage scenarios. [7]

(b) Define event-driven programming in the context of Java applications. Illustrate how user interactions like button clicks or keystrokes are managed in a Graphical User Interface (GUI) by describing the event delegation model and the role of event listeners. [7]