



Rollno.....

End Semester Examination, December 2024

B.TECH – 3rd Semester

Java Programming (ENCS201)

Time: 2.5 hrs.

Max Marks: **50**

Note: *All Questions are compulsory. Marks are indicated against each question.*

PART-A

Q.1 *First question will consist of 7 small questions covering entire syllabus out of which student has to attempt 5 questions and understanding level will be assessed.*

(a) What will be the output of the following code snippet?

```
public class StringExample {  
    public static void main(String[] args) {  
        String str1 = "hello";  
        String str2 = new String("hello");  
        System.out.println("str1 == str2: " + (str1 == str2));  
        System.out.println("str1.equals(str2): " + str1.equals(str2));  
    }  
}
```

[CO-1] [L-2] **2**

(b) What will be the output of the following code snippet?

```
class Animal {  
    void sound() {  
        System.out.println("Animal sound");  
    }  
}  
  
class Dog extends Animal {  
    void sound() {  
        System.out.println("Bark");  
    }  
}  
  
public class Main {  
    public static void main(String[] args) {  
        Animal a = new Dog();  
        a.sound();  
    }  
}
```

[CO-1] [L-4] **2**

(c) What will be the output of the following code snippet?

```
public class ExceptionExample {  
    public static void main(String[] args) {  
        try {  
            int result = 10 / 0;  
            System.out.println("Result: " + result);  
        } catch (ArithmeticException e) {  
            System.out.println("ArithmeticException caught");  
        } finally {  
            System.out.println("Finally block executed");  
        }  
    }  
}
```

[CO-1] [L-2] **2**

(d) What will be the output of the following code snippet?

```
class MyThread extends Thread {
    public void run() {
        for (int i = 1; i <= 3; i++) {
            System.out.println(Thread.currentThread().getName() + " - " + i);
        }
    }
}

public class Main {
    public static void main(String[] args) {
        MyThread t1 = new MyThread();
        MyThread t2 = new MyThread();
        t1.start();
        t2.start();
    }
}
```

[CO-1] [L-2] 2

(e) Explain the difference between == and .equals() when comparing two strings in Java. Why is it important to understand this difference? [CO-1] [L-2] 2

(f) What is the purpose of the try-catch-finally structure in Java exception handling? Describe the role of each part with a brief example. [CO-1] [L-4] 2

(g) Define encapsulation in the context of object-oriented programming. How does encapsulation improve the reliability and security of a Java application? [CO-1] [L-] 2

2x5

PART-B

Q.2 Explain the concept of inheritance in Java with an example. How does inheritance promote code reusability? Differentiate between single, multilevel, and hierarchical inheritance with code illustrations. [CO-3] [L-3] 10

OR

Design a Java class hierarchy for a **Vehicle** with subclasses for **Car**, **Bike**, and **Truck**. Include common attributes such as **speed** and **fuel** and define specific attributes and methods for each subclass. Demonstrate polymorphism by overriding a method such as **drive()** in each subclass. [CO-3] [L-3] 10

Q.3 What are interfaces in Java, and how do they differ from abstract classes? Write a Java program demonstrating the implementation of an interface in a class and explain how interfaces support multiple inheritance. [CO-4] [L-3] 10

OR

Discuss the importance of packages in Java. Describe the steps involved in creating and using packages, along with an example that demonstrates how packages help in organizing Java classes and managing access control. [CO- 4] [L-3] 10

Q.4 Explain the concept of exception handling in Java and its significance in robust application development. Write a program to demonstrate the use of try-catch-finally blocks, explaining the role of each part. [CO-5] [L-3] 10

OR

Define custom exceptions in Java and describe the process of creating them. Provide a code example where a custom exception is created to handle a specific error case and explain how it differs from Java's built-in exceptions. [CO- 5] [L-3] 10

Q.5 Describe the concept of multithreading in Java and its importance in developing responsive applications. Explain thread synchronization with an example, highlighting the issues it solves and how it is implemented. [CO-5] [L-3] 10

OR

Write a Java program that reads text from a file, counts the occurrences of each word, and prints them in descending order of frequency. Use Array or strings for storing and sorting the word based on number of characters. [CO- 5] [L-3] 10