



Rollno.....

End Semester Examination, December 2024
BCA(AI/DS)- III Semester
ENCA203 (Fundamentals of Java Programming)

Time: 2.5 hrs.

Max Marks: **50**

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

PART-A

Q.1 *Attempt any five questions:*

(a) Fill in the blanks

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

- i. The _____ converts bytecode into machine code.
- ii. To read an integer input using Scanner, we use the method _____.
- iii. Runtime polymorphism in Java is achieved through _____ methods.
- iv. The _____ keyword in Java is used to refer to the parent class object.

(b) Multiple choice question

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

i. Which of the following blocks must be present in a try-catch-finally structure?

- a) try
- b) catch
- c) finally
- d) all of the above

ii. Which of the following is not a runtime exception?

- a) ClassCastException
- b) NoSuchMethodException
- c) NullPointerException
- d) IllegalArgumentException

iii. Which method is used to start a thread execution?

- a) run()
- b) start()
- c) execute()
- d) init()

iv. Which of the following is not a type of polymorphism in Java?

- a) Compile-time polymorphism
- b) Runtime polymorphism
- c) Static polymorphism
- d) Dynamic polymorphism

(c) Find Output

[CO-1] [L-3] **2**

```
i. class Example {  
    static int count = 0;  
    Example() {  
        count++;  
    }  
}  
Example e1 = new Example();  
Example e2 = new Example();  
System.out.println(Example.count);
```

```

ii. class Shape {
    void draw() {
        System.out.println("Drawing Shape");
    }
}

class Circle extends Shape { void draw() {
    System.out.println("Drawing Circle");
}

}

class Square extends Shape {
    void draw() {
        System.out.println("Drawing Square");
    }
}

public class Test {

    public static void main(String[] args) {
        Shape s;
        s = new Circle();
        s.draw();
        s = new Square();
        s.draw();
    }
}

```

(d) Find error

[CO-1] [L-3] 2

```

i. class MyClass {
    private int myVar;

    public void myMethod() {
        System.out.println(myVar);
    }
}

class AnotherClass {

    public void anotherMethod() { MyClass
        obj = new MyClass();
        System.out.println(obj.myVar);
    } }

ii. class Animal { void
    sound() {
        System.out.println("Animal makes a sound");
    }
}

class Dog extends Animal {

```

```

        void Sound() {
            System.out.println("Dog barks");
        }
    }

    public class Main {

        public static void main(String[] args) {
            Animal myDog = new Dog();
            myDog.sound();
        }
    }

```

- | | |
|---|-----------------------|
| (e) Explain the concept of polymorphism in Java and its benefits. | [CO-1] [L-2] 2 |
| (f) Explain how to read data from a file using <i>BufferedReader</i> in Java. | [CO-1] [L-2] 2 |
| (g) Explain the lifecycle of a thread in Java. | [CO-1] [L-2] 2 |

2x5

PART-B

Q.2 Describe the four main principles of Object-Oriented Programming (OOP). Discuss the importance of constructors in object-oriented programming. Develop a Java program to handle student grades, including methods to calculate the average, highest, and lowest grades. Use method overloading to support different input formats (array, list) and constructors to initialize student data. [CO-1] [L-4] **10**

OR

What is a nested if statement, and when is it used? What is switch case and how its different from if else statement. Create a menu-driven Java program that offers various options to the user. The program should display a menu with options such as performing arithmetic operations, calculating area and perimeter of geometric shapes, solving equations, etc. The user can choose an option from the menu, provide necessary inputs, and the program should execute the selected operation accordingly. [CO-1] [L-3] **10**

Q.3 What is an abstract class in Java, and how is it different from an interface? Explain the concept of interface inheritance in Java with an example. Create an interface Playable with a method play(). Create two classes Guitar and Piano that implement Playable and demonstrate polymorphism. [CO-2] [L-4] **10**

OR

How can an interface be used to achieve polymorphism? Explain how you can create a hierarchy of classes using inheritance and abstract classes. Create an abstract class Employee with an abstract method calculateSalary(). Create two subclasses Manager and Developer that extend Employee and provide specific implementations for the calculateSalary() method. [CO-2] [L-3] **10**

Q.4 . Explain the concept of exception handling in Java. Describe the various types of exceptions, including checked and unchecked exceptions, and explain how they are declared and handled. Additionally, elaborate on the use of the finally clause and how custom exceptions can be created. Provide code examples to illustrate your points. [CO-3] [L-4] **10**

OR

Discuss the fundamental concepts of multithreaded programming in Java. Explain how the Java thread model works, including thread priorities. Describe how threads can be created and managed using the Thread class and the Runnable interface. [CO-3] [L-3] **10**

- Q.5 Explain the use of byte streams in Java with examples of FileInputStream and FileOutputStream. Write a code snippet to read a binary file and write its contents to another binary file using these classes. [CO-4] [L-3] **10**

OR

Discuss the concept of file permissions and attributes in Java. Write a code snippet to display the read, write, and execute permissions of a file. [CO- 5] [L-4] **10**