CS-22: Programming with Java

UNIT 1: History, Introduction, Language Basics, Classes & Objects

[5 Marks] Q: Explain Java Tokens with examples. Java tokens are the smallest individual units in a Java program. Java programs are a collection of tokens, and the compiler uses them to understand and compile the program.

Types of Tokens:

1. **Keywords** – Reserved words like `class`, `public`, `if`, `else`, etc.
2. **Identifiers** – Names used for classes, methods, variables, e.g., `Student`, `sum`, `main`.
3. **Literals** – Constant values like `100`, `3.14`, `'A'`, `"Hello"`.
4. **Operators** – Symbols that perform operations, e.g., `+`, `-`, `*`, `/`, `==`.
5. **Separators** – Special characters like `;`, `{}`, `()`, `[]`.
6. **Comments** – Used for documentation (`// single line`, `/* multiline */`).

[3 Marks] Q: Write a Java program to accept marks of 5 subjects in an array and display the total and average.

```
import java.util.Scanner;
class TotalAverage {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      int[] marks = new int[5];
      int total = 0;
      for (int i = 0; i < 5; i++) {
         System.out.print("Enter mark for subject " + (i+1) + ": ");
         marks[i] = sc.nextInt();
         total += marks[i];
      }
      double average = total / 5.0;
      System.out.println("Total: " + total);
      System.out.println("Average: " + average);
   }
}
```

[1 Mark] Q: What is the role of the `main()` method in Java? The `main()` method is the entry point of any standalone Java application. The JVM looks for `public static void main(String[] args)` to start the execution.

[1 Mark] Q: Define type casting in Java. Type casting is converting one data type into another. For example: `(int) 3.14` will convert the float into an integer.

UNIT 2: Inheritance, Java Packages

[5 Marks] Q: Explain types of inheritance with a program. Types of Inheritance in Java:

1. Single
2. Multilevel
3. Hierarchical (Java does not support multiple inheritance with classes)

Multilevel Example:

```
class A {
    void msg() { System.out.println("Hello from A"); }
}
class B extends A {
    void greet() { System.out.println("Hello from B"); }
}
class C extends B {
    void welcome() { System.out.println("Welcome from C"); }
    public static void main(String[] args) {
        C obj = new C();
        obj.msg(); obj.greet(); obj.welcome();
    }
}
```

[3 Marks] Q: What are access specifiers in Java? Access specifiers determine the visibility of classes and class members.

- **public** – Accessible from everywhere.
- **private** – Accessible only within the class.
- **protected** – Accessible within the package and subclass.
- **default** – Accessible only within the same package.

[1 Mark] Q: What is method overriding? Method overriding is redefining a superclass method in a subclass with the same signature.

[1 Mark] Q: Name any two Java API packages.

1. `java.util`
2. `java.io`

UNIT 3: Exception Handling, Threading, Streams

[5 Marks] Q: Life cycle of a thread and program using Thread class. Thread life cycle stages:

- New
- Runnable
- Running
- Blocked/Waiting
- Terminated

Thread Example:

```
class MyThread extends Thread {
    public void run() {
        System.out.println("Thread running: " + getName());
    }
    public static void main(String[] args) {
        MyThread t1 = new MyThread();
        t1.start();
    }
}
```

[3 Marks] Q: Program using try-catch-finally.

```
public class ExceptionExample {
    public static void main(String[] args) {
        try {
            int result = 10 / 0;
        } catch (ArithmeticException e) {
            System.out.println("Cannot divide by zero");
        } finally {
            System.out.println("Finally block executed");
        }
    }
}
```

[1 Mark] Q: Difference between `throw` and `throws`.

- `throw` is used to explicitly throw an exception.
- `throws` declares exceptions a method can throw.

[1 Mark] Q: Name two character stream classes.

- FileReader
- BufferedReader

UNIT 4: JavaFX Basics and Event-driven Programming

[5 Marks] Q: JavaFX program using TranslateTransition.

```
import javafx.animation.TranslateTransition;
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.shape.Rectangle;
import javafx.scene.layout.Pane;
import javafx.stage.Stage;
import javafx.util.Duration;

public class MoveRectangle extends Application {
    public void start(Stage stage) {
        Rectangle rect = new Rectangle(50, 50, 100, 50);
        TranslateTransition tt = new TranslateTransition(Duration.seconds(2), rect);
        tt.setFromX(0);
        tt.setToX(200);
        tt.setCycleCount(TranslateTransition.INDEFINITE);
        tt.setAutoReverse(true);
        tt.play();

        Pane root = new Pane(rect);
        Scene scene = new Scene(root, 400, 200);
        stage.setScene(scene);
        stage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```

[3 Marks] Q: What is property binding in JavaFX? Property binding allows two UI elements to stay in sync. If one property changes, the other automatically updates.

```
textField.textProperty().bind(slider.valueProperty().asString());
```

[1 Mark] Q: Name two layout panes.

- BorderPane
- GridPane

[1 Mark] Q: What is an event source in JavaFX? An event source is the UI control that generates the event (e.g., a button click).

UNIT 5: JavaFX UI Controls and Multimedia

[5 Marks] Q: JavaFX form with ComboBox, Label, Button, TextField.

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;

public class FormApp extends Application {
    public void start(Stage stage) {
        Label label = new Label("Enter Name:");
        TextField name = new TextField();
        ComboBox<String> gender = new ComboBox<>();
        gender.getItems().addAll("Male", "Female");
        Button submit = new Button("Submit");
        Label output = new Label();

        submit.setOnAction(e -> output.setText("Hello " + name.getText() + ", Gender: " + gender.getValue()));

        VBox vbox = new VBox(10, label, name, gender, submit, output);
        Scene scene = new Scene(vbox, 300, 200);
        stage.setScene(scene);
        stage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```

[3 Marks] Q: Explain any 3 UI controls in JavaFX.

1. **Button** – A push button which triggers an action.
2. **TextField** – A control to input single-line text.
3. **ComboBox** – A dropdown that lets the user choose from a list.

[1 Mark] Q: Two multimedia features in JavaFX.

- Audio playback (Media & MediaPlayer)
- Video playback (MediaView)

[1 Mark] Q: Use of TextArea in JavaFX. `TextArea` allows for multi-line text input, useful for entering long messages or descriptions.

End

Paper 2

UNIT 1: History, Introduction, Language Basics, Classes & Objects

[5 Marks] Q: Explain the different types of arrays in Java with examples. Java supports three types of arrays:

1. **One-dimensional Array:**

```
int[] arr = {1, 2, 3};
```

2. **Multidimensional Array (Rectangular):**

```
int[][] matrix = {{1, 2}, {3, 4}};
```

3. **Jagged Array:**

```
int[][] jagged = new int[2][];
jagged[0] = new int[2];
jagged[1] = new int[3];
```

Arrays store similar type data and are accessed using indices.

[3 Marks] Q: Write a program to demonstrate the use of command-line arguments.

```
public class CmdArgs {
    public static void main(String[] args) {
        for(int i = 0; i < args.length; i++) {
            System.out.println("Argument " + i + ": " + args[i]);
        }
    }
}
```

Compile and run: `java CmdArgs Hello World`

[1 Mark] Q: What are Java keywords? Keywords are reserved words in Java used for specific purposes, e.g., `class`**,** `if`**,** `else`**,** `while`**.**

[1 Mark] Q: Name two data types in Java.

1. **int**
2. **boolean**

UNIT 2: Inheritance, Java Packages

[5 Marks] Q: Explain method overriding with a Java program. Method overriding allows a subclass to provide a specific implementation of a method in the parent class.

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

[3 Marks] Q: Explain inner classes and anonymous inner classes.

- **Inner Class: Defined inside another class.**
- **Anonymous Inner Class: No name, used to override methods on the fly.**

```
abstract class Hello {
    abstract void greet();
}
class Test {
    public static void main(String[] args) {
        Hello h = new Hello() {
            void greet() {
                System.out.println("Hi!");
            }
        };
        h.greet();
    }
}
```

[1 Mark] Q: What is a package in Java? A package is a namespace that organizes classes and interfaces, like `java.util`, `java.io`.

[1 Mark] Q: Name one import type in Java. Static import: `import static java.lang.Math.*;`

UNIT 3: Exception Handling, Threading, Streams

[5 Marks] Q: Explain thread synchronization with example. Thread synchronization ensures only one thread accesses a resource at a time.

```
class Counter {
    int count = 0;
    synchronized void increment() {
        count++;
    }
}
class TestThread extends Thread {
    Counter c;
    TestThread(Counter c) { this.c = c; }
    public void run() {
        for (int i = 0; i < 1000; i++) c.increment();
    }
    public static void main(String[] args) throws Exception {
        Counter c = new Counter();
        TestThread t1 = new TestThread(c);
        TestThread t2 = new TestThread(c);
        t1.start(); t2.start();
        t1.join(); t2.join();
        System.out.println("Count: " + c.count);
    }
}
```

[3 Marks] Q: Explain the use of BufferedReader with a program.

```
import java.io.*;
class ReadFile {
    public static void main(String[] args) throws IOException {
        BufferedReader br = new BufferedReader(new FileReader("test.txt"));
        String line;
        while ((line = br.readLine()) != null) {
            System.out.println(line);
        }
        br.close();
    }
}
```

[1 Mark] Q: Name any one stream class. `FileInputStream`

[1 Mark] Q: What is multithreading? Executing multiple threads simultaneously is called multithreading.

UNIT 4: JavaFX Basics and Event-driven Programming

[5 Marks] Q: Write a JavaFX program to display a circle and set fill color.

```
import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.paint.Color;

import javafx.scene.shape.Circle;

import javafx.stage.Stage;

import javafx.scene.layout.Pane;


public class CircleApp extends Application {

    public void start(Stage stage) {

        Circle circle = new Circle(150, 150, 100, Color.BLUE);

        Pane pane = new Pane(circle);

        Scene scene = new Scene(pane, 300, 300);

        stage.setScene(scene);

        stage.setTitle("Circle Example");

        stage.show();

    }

    public static void main(String[] args) {

        launch(args);

    }

}
```

[3 Marks] Q: What are event handlers in JavaFX? Event handlers handle user-generated events like button clicks. They are registered with nodes using `setOnAction` or similar methods.

[1 Mark] Q: Name one event in JavaFX. `ActionEvent`

[1 Mark] Q: What is a pane in JavaFX? A pane is a container for layout and positioning of UI components.

UNIT 5: JavaFX UI Controls and Multimedia

[5 Marks] Q: Design a form in JavaFX with TextField, Button, and Label.

```
import javafx.application.Application;

import javafx.scene.Scene;

import javafx.scene.control.*;

import javafx.scene.layout.VBox;

import javafx.stage.Stage;


public class Form extends Application {

    public void start(Stage stage) {

        Label label = new Label("Enter your name:");

        TextField tf = new TextField();

        Button btn = new Button("Submit");

        Label output = new Label();

        btn.setOnAction(e -> output.setText("Hello, " + tf.getText()));

        VBox vbox = new VBox(10, label, tf, btn, output);

        Scene scene = new Scene(vbox, 300, 200);

        stage.setScene(scene);

        stage.setTitle("Form Example");

        stage.show();

    }

    public static void main(String[] args) {

        launch(args);

    }

}
```

[3 Marks] Q: Explain the use of ComboBox, Slider, and TextArea.

- **ComboBox: A dropdown menu to select one item.**
- **Slider: Allows selecting a numeric value from a range.**
- **TextArea: Used for multi-line text input.**

[1 Mark] Q: What is a ListView? It displays a scrollable list of items.

[1 Mark] Q: Name one multimedia class in JavaFX. `MediaPlayer`

End

UNIT 1: History, Introduction, Basics, Classes & Objects

[5 Marks] Q: Explain the features of Java that make it platform-independent and secure. Answer: Java is platform-independent due to the concept of bytecode and Java Virtual Machine (JVM). When a Java program is compiled, it is converted into bytecode, which can run on any system with a JVM. This eliminates platform dependency. Java is secure because:

- **No use of explicit pointers.**
- **Bytecode verification.**
- **Automatic memory management and garbage collection.**
- **Built-in security APIs and sandboxing.**

[3 Marks] Q: Write a Java program to demonstrate the use of a for-each loop.

```
public class ForEachDemo {
    public static void main(String[] args) {
        int[] numbers = {10, 20, 30, 40};
        for (int num : numbers) {
            System.out.println(num);
        }
    }
}
```

[1 Mark] Q: What is a class in Java? Answer: A class is a blueprint or template from which objects are created.

[1 Mark] Q: Name any two loop structures in Java. Answer: for loop, while loop

UNIT 2: Inheritance and Packages

[5 Marks] Q: Describe different types of inheritance in Java with a diagram. Answer: Java supports:

1. **Single Inheritance – One class inherits from another.**
2. **Multilevel Inheritance – A class inherits from a derived class.**
3. **Hierarchical Inheritance – Multiple classes inherit from one base class. Note: Java does not support multiple inheritance through classes to avoid ambiguity (Diamond problem). Diagram:**

A

**/ **

B C (Hierarchical)

[3 Marks] Q: What is the use of the `super` keyword? Give an example. Answer: `super` is used to refer to the immediate parent class object. It is used to call superclass constructors and methods.

```
class Parent {
    void display() {
        System.out.println("Parent method");
    }
}
class Child extends Parent {
    void display() {
        super.display();
        System.out.println("Child method");
    }
}
```

[1 Mark] Q: What is the purpose of import statements in Java? Answer: To use classes from predefined or user-defined packages.

[1 Mark] Q: Give an example of a predefined package. Answer: `java.util`

UNIT 3: Exception Handling, Threads & Streams

[5 Marks] Q: Explain exception handling with try-catch-finally block. Answer: Exception handling provides a way to handle runtime errors to maintain normal program flow. Java provides:

- `try`: **Code that might throw an exception.**
- `catch`: **Handles the exception.**
- `finally`: **Executes regardless of exception occurrence.**

```
try {
    int a = 10 / 0;
} catch (ArithmeticException e) {
    System.out.println("Cannot divide by zero");
} finally {
    System.out.println("Finally block");
}
```

[3 Marks] Q: Write a Java program to create a thread using the Runnable interface.

```
class MyRunnable implements Runnable {
    public void run() {
        System.out.println("Thread running");
    }
}
public class RunnableDemo {
    public static void main(String[] args) {
        Thread t = new Thread(new MyRunnable());
        t.start();
    }
}
```

[1 Mark] Q: What is a stream in Java? Answer: A stream is a sequence of data used for input and output.

[1 Mark] Q: Give one example of a checked exception. Answer: `IOException`

UNIT 4: JavaFX Basics and Events

[5 Marks] Q: Explain the structure of a basic JavaFX application. Answer: A basic JavaFX app includes:

1. `start(Stage primaryStage)` **method to create GUI.**
2. **A** `Scene` **object added to the stage.**
3. **Controls like** `Label`, `Button` **inside layout containers like** `VBox`, `StackPane`.
4. `launch(args)` **to start the application.**

Example:

```
public class MyApp extends Application {
    public void start(Stage stage) {
        Label label = new Label("Hello JavaFX");
        StackPane root = new StackPane(label);
        Scene scene = new Scene(root, 300, 200);
        stage.setScene(scene);
        stage.setTitle("Demo");
        stage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```

[3 Marks] Q: What is an event in JavaFX? How do you handle it? Answer: An event is a signal that something has occurred (e.g., button click). It is handled by registering an event handler:

```
button.setOnAction(e -> System.out.println("Clicked!"));
```

[1 Mark] Q: What is a Scene in JavaFX? Answer: It represents the container for all content in a window.

[1 Mark] Q: Name one layout pane in JavaFX. Answer: VBox

UNIT 5: JavaFX UI Controls and Multimedia

[5 Marks] Q: Write a JavaFX program to take user input and display it using Label.

```
public class UserInputApp extends Application {
    public void start(Stage stage) {
        TextField tf = new TextField();
        Button btn = new Button("Submit");
        Label label = new Label();
        btn.setOnAction(e -> label.setText("Hello, " + tf.getText()));
        VBox root = new VBox(10, tf, btn, label);
        Scene scene = new Scene(root, 300, 150);
        stage.setScene(scene);
        stage.setTitle("User Input");
        stage.show();
    }
    public static void main(String[] args) {
        launch(args);
    }
}
```

[3 Marks] Q: Explain any three UI controls in JavaFX. Answer:

1. **TextField – Input field for text.**
2. **Button – Executes action on click.**
3. **Label – Displays static or dynamic text.**

[1 Mark] Q: What is the use of MediaPlayer in JavaFX? Answer: It is used to play audio and video files.

[1 Mark] Q: Name one control used for multi-line input. Answer: TextArea

---

End