Week 8 - S8 - Core OOP - Abstract Class And Interface - Lab Problem

Name: Ramesh Harisabapathi Chettiar

Date of Submission:07/10/25

QNO1→

Create an abstract class Fruit with protected fields color and taste. Add an abstract method showDetails().

Create an interface Edible with method nutrientsInfo().

Create a class Apple that extends Fruit and implements Edible, adding a variety field.

Hints:

- Use abstract for parent class.
- Use interface for common behavior.
- Implement both abstract and interface methods.

Fruit.java

```java
public abstract class Fruit {
    protected String color;
    protected String taste;

    public Fruit(String color, String taste) {
        this.color = color;
        this.taste = taste;
    }

    public abstract void showDetails();
}
```

Edible.java

```java
public interface Edible {
    void nutrientsInfo();
}
```

Apple.java

```java
public class Apple extends Fruit implements Edible {
    private String variety;

    public Apple(String color, String taste, String variety) {
        super(color, taste);
        this.variety = variety;
    }

    @Override
    public void showDetails() {
        System.out.println("This is a " + variety + " apple.");
        System.out.println("It is " + color + " and tastes " + taste + ".");
    }

    @Override
    public void nutrientsInfo() {
        System.out.println("Apples are a good source of fiber and Vitamin C.");
    }
}
```

Main.java

```java
public class Main {

    public static void main(String[] args) {
        Apple myApple = new Apple("Red", "Sweet", "Fuji");
        myApple.showDetails();
        myApple.nutrientsInfo();
    }
}
```

OUTPUT→

```
PS C:\Users\Ramesh\Personal Folders\MISCELLANEOUS\ENTRANCE EXAMS\SRM\SEMESTERS\SEMESTER-3\JAVA-STEP\Weeks\Week 8\Lab Problems\Program1> cd "c:\Users\Ramesh\Personal Folders\MISCELLANEOUS\ENTRANCE EXAMS\SRM\SEMESTERS\SEMESTER-3\JAVA-STEP\Weeks\Week 8\Lab Problems\Program1\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
This is a Fuji apple.
It is Red and tastes Sweet.
Apples are a good source of fiber and Vitamin C.
```

QNO2→

Create an abstract class Shape with fields area and perimeter. Add abstract methods calculateArea() and calculatePerimeter().

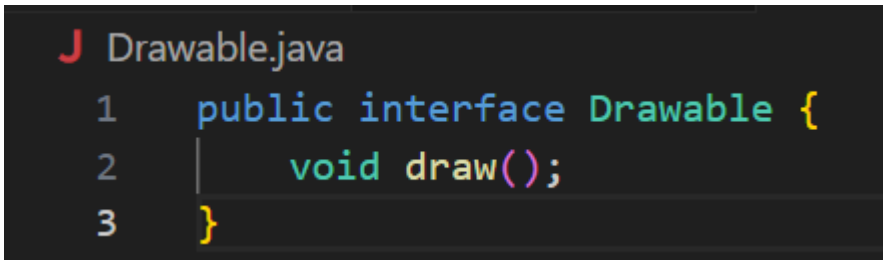
Create an interface Drawable with method draw().

Create a class Circle extending Shape and implementing Drawable.

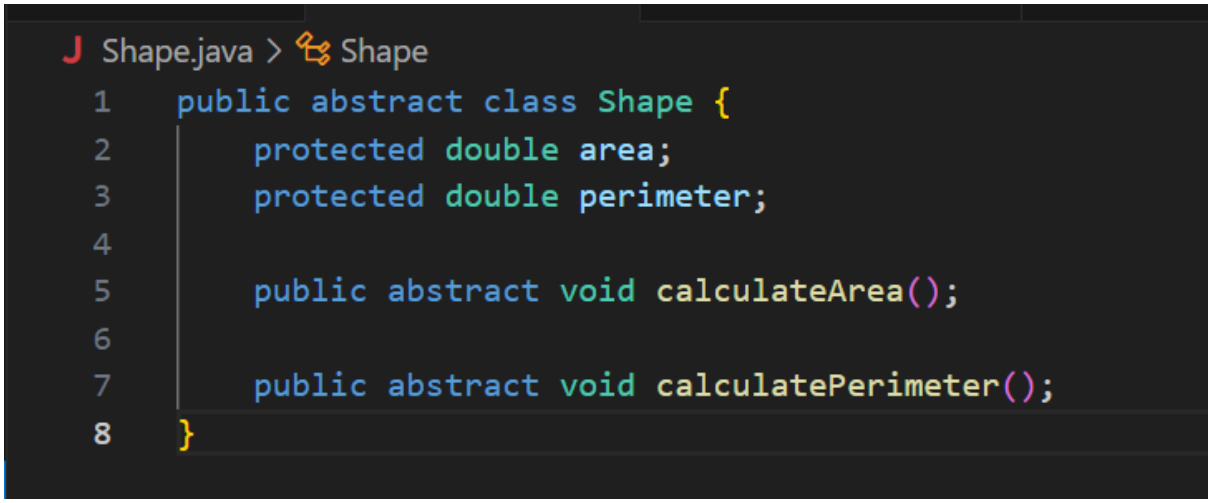
Hints:

- Abstract methods must be overridden in child class.
- Use interface to add extra behavior.

Drawable.java



Shape.java



Circle.java

```java
public class Circle extends Shape implements Drawable {
    private double radius;

    public Circle(double radius) {
        this.radius = radius;
    }

    @Override
    public void calculateArea() {
        this.area = Math.PI * radius * radius;
    }

    @Override
    public void calculatePerimeter() {
        this.perimeter = 2 * Math.PI * radius;
    }

    @Override
    public void draw() {
        System.out.println("Drawing a circle with radius " + radius);
    }

    // Optional: Add methods to display calculated values
    public double getArea() {
        return area;
    }

    public double getPerimeter() {
        return perimeter;
    }
}
```

Main.java

```java
public class Main {

    public static void main(String[] args) {
        // Create a Circle object
        Circle myCircle = new Circle(5.0);

        // Call methods from the abstract class (Shape)
        myCircle.calculateArea();
        myCircle.calculatePerimeter();
        System.out.println("Circle Area: " + myCircle.getArea());
        System.out.println("Circle Perimeter: " + myCircle.getPerimeter());

        System.out.println(); // Add a newline for separation

        // Call the method from the interface (Drawable)
        myCircle.draw();
    }
}
```

OUTPUT→

```
PS C:\Users\Ramesh\Personal Folders\MISCELLANEOUS\ENTRANCE EXAMS\SRM\SEMESTERS\SEMESTER-3\JAVA-STEP\Weeks\Week 8\Lab
 Problems\Program2> cd "c:\Users\Ramesh\Personal Folders\MISCELLANEOUS\ENTRANCE EXAMS\SRM\SEMESTERS\SEMESTER-3\JAVA-
STEP\Weeks\Week 8\Lab Problems\Program2\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
Circle Area: 78.53981633974483
Circle Perimeter: 31.41592653589793

Drawing a circle with radius 5.0
```

QNO3→

Create an abstract class Vehicle with protected fields speed and fuelType. Add an abstract method startEngine().

Create an interface Maintainable with method serviceInfo().

Create a class Car that extends Vehicle and implements Maintainable.

Hints:

- Use extends and implements together.
- Provide concrete implementations for abstract and interface methods.

Maintainable.java

```java
public interface Maintainable {
    void serviceInfo();
}
```

Vehicle.java

```java
public abstract class Vehicle {
    protected int speed;
    protected String fuelType;

    public Vehicle(int speed, String fuelType) {
        this.speed = speed;
        this.fuelType = fuelType;
    }

    public abstract void startEngine();
}
```

Car.java

```java
public class Car extends Vehicle implements Maintainable {
    public Car(int speed, String fuelType) {
        super(speed, fuelType);
    }

    @Override
    public void startEngine() {
        System.out.println("The " + this.fuelType + " car's engine has started.");
    }

    @Override
    public void serviceInfo() {
        System.out.println("Car service is recommended every 15,000 kilometers or 1 year.");
    }

    // Optional method to show current speed
    public void showSpeed() {
        System.out.println("Current speed: " + this.speed + " km/h");
    }
}
```

Main.java

```java
public class Main {
    Run main | Debug main
    public static void main(String[] args) {
        // Create an instance of the Car class
        Car myCar = new Car(100, "Petrol");

        // Call methods from the abstract class (Vehicle)
        myCar.startEngine();
        myCar.showSpeed();

        System.out.println(); // A blank line for readability.

        // Call the method from the interface (Maintainable)
        myCar.serviceInfo();
    }
}
```

OUTPUT→

```
PS C:\Users\Ramesh\Personal Folders\MISCELLANEOUS\ENTRANCE EXAMS\SRM\SEMESTERS\SEMESTER-3\JAVA-STEP\Weeks\Week 8\Lab Problems\Program3> cd "c:\Users\Ramesh\Personal Folders\MISCELLANEOUS\ENTRANCE EXAMS\SRM\SEMESTERS\SEMESTER-3\JAVA-STEP\Weeks\Week 8\Lab Problems\Program3\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
The Petrol car's engine has started.
Current speed: 100 km/h

Car service is recommended every 15,000 kilometers or 1 year.
```

QNO4→

Create an abstract class Employee with fields name and salary. Add abstract method calculateBonus().

Create an interface Payable with method generatePaySlip().

Create a class Manager that extends Employee and implements Payable.

Hints:

- Use abstract method for bonus calculation.
- Interface method should handle pay slip generation.

Payable.java

```java
public interface Payable {
    void generatePaySlip();
}
```

Employee.java

```java
public abstract class Employee {
    protected String name;
    protected double salary;

    public Employee(String name, double salary) {
        this.name = name;
        this.salary = salary;
    }

    public abstract void calculateBonus();
}
```

Manager.java

```java
public class Manager extends Employee implements Payable {
    public Manager(String name, double salary) {
        super(name, salary);
    }

    @Override
    public void calculateBonus() {
        double bonus = salary * 0.15;
        System.out.println(name + "'s bonus is INR " + bonus);
    }

    @Override
    public void generatePaySlip() {
        System.out.println("Pay slip for " + name + " generated successfully.");
        System.out.println("Gross Salary: INR " + salary);
    }
}
```

Main.java

```java
public class Main {
    public static void main(String[] args) {
        // Create an instance of the Manager class
        Manager myManager = new Manager("Ramesh", 85000.00);

        // Call methods from the abstract class (Employee) and interface (Payab
        myManager.calculateBonus();
        System.out.println(); // Add a newline for separation
        myManager.generatePaySlip();
    }
}
```

OUTPUT→

```
PS C:\Users\Ramesh\Personal Folders\MISCELLANEOUS\ENTRANCE EXAMS\SRM\SEMESTERS\SEMESTER-3\JAVA-STEP\Weeks\Week 8\Lab
 Problems\Program4> cd "c:\Users\Ramesh\Personal Folders\MISCELLANEOUS\ENTRANCE EXAMS\SRM\SEMESTERS\SEMESTER-3\JAVA-
STEP\Weeks\Week 8\Lab Problems\Program4\" ; if ($?) { javac Main.java } ; if ($?) { java Main }
Ramesh's bonus is INR 12750.0

Pay slip for Ramesh generated successfully.
Gross Salary: INR 85000.0
```