Test:4

1. Create a base class called Shape with virtual functions area() and perimeter(). Derive two classes Rectangle and Triangle from the base class. Implement the area() and perimeter() functions for each class.

Program:

```
// Shape interface
interface Shape {
    // Abstract method for calculating area
    double area();
    // Abstract method for calculating perimeter
    double perimeter();
}
// Rectangle class implementing Shape
class Rectangle implements Shape {
    private double length;
    private double width;
    // Constructor
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }
    // Method to calculate area of rectangle
    @Override
    public double area() {
        return length * width;
    }
```

```
    // Method to calculate perimeter of rectangle
    @Override
    public double perimeter() {
        return 2 * (length + width);
    }
}
// Triangle class implementing Shape
class Triangle implements Shape {
    private double side1;
    private double side2;
    private double side3;
    // Constructor
    public Triangle(double side1, double side2, double side3) {
        this.side1 = side1;
        this.side2 = side2;
        this.side3 = side3;
    }

    // Method to calculate area of triangle using Heron's formula
    @Override
    public double area() {
        double s = (side1 + side2 + side3) / 2;
        return Math.sqrt(s * (s - side1) * (s - side2) * (s - side3));
    }
    // Method to calculate perimeter of triangle
    @Override
```

```
    public double perimeter() {
        return side1 + side2 + side3;
    }
}
// Main class to demonstrate usage
public class Main {
    public static void main(String[] args) {
        // Create instances of Rectangle and Triangle
        Rectangle rectangle = new Rectangle(4.0, 3.0);
        Triangle triangle = new Triangle(3.0, 4.0, 5.0);
        // Calculate and print areas and perimeters
        System.out.println("Rectangle:");
        System.out.println("Area: " + rectangle.area());
        System.out.println("Perimeter: " + rectangle.perimeter());
        System.out.println("\nTriangle:");
        System.out.println("Area: " + triangle.area());
        System.out.println("Perimeter: " + triangle.perimeter());
    }
}
```

Output:

```
Rectangle:
Area: 12.0
Perimeter: 14.0
Triangle:
Area: 6.0
Perimeter: 12.0
```

2. Create a base class called Animal with a virtual function move(). Derive two classes Bird and

Fish from the base class. Implement the move() function for each class.

Program:

```
// Animal class
abstract class Animal {
    // Abstract method move (to be implemented by subclasses)
    public abstract void move();
}

// Bird subclass of Animal
class Bird extends Animal {
    // Implementing move method for Bird
    @Override
    public void move() {
        System.out.println("Bird is flying.");
    }
}

// Fish subclass of Animal
class Fish extends Animal {
    // Implementing move method for Fish
    @Override
    public void move() {
        System.out.println("Fish is swimming.");
    }
}

// Main class to demonstrate usage
```

```
public class Main {
    public static void main(String[] args) {
        // Create instances of Bird and Fish
        Bird bird = new Bird();
        Fish fish = new Fish();

        // Call move method for Bird and Fish
        bird.move();
        fish.move();
    }
}
```

Output:

Bird is flying.

Fish is swimming.