

Write a Java program to create a class called "Shape" with abstract methods for calculating area and perimeter, and subclasses for "Rectangle", "Circle", and "Triangle".

Note: An abstract class is a class that is declared abstract—it may or may not include abstract methods. Abstract classes cannot be instantiated, but they can be subclassed.

An abstract method is a method that is declared without an implementation (without braces, and followed by a semicolon).

Sample Solution:

Java Code:

```
//Shape.java
// Define the Shape class as an abstract class
public abstract class Shape {

    // Abstract method to get the area of the shape
    // This method must be implemented by any subclass of Shape
    public abstract double getArea();

    // Abstract method to get the perimeter of the shape
    // This method must be implemented by any subclass of Shape
    public abstract double getPerimeter();
}
```

The above "Shape" class is an abstract class that defines the basic properties and behaviours of a geometric shape. It contains two abstract methods, "getArea()" and "getPerimeter()", which are to be implemented by its subclasses. Since the class is abstract, it cannot be instantiated on its own, but it provides a framework for other classes to extend and implement its methods.

```
//Rectangle.java
// Define the Rectangle class, which extends the Shape class
public class Rectangle extends Shape {

    // Private fields to store the length and width of the rectangle
    private double length;
    private double width;

    // Constructor to initialize the length and width of the rectangle
    public Rectangle(double length, double width) {
```

```

        this.length = length; // Set the length field to the provided length
        this.width = width; // Set the width field to the provided width
    }

    // Method to calculate and return the area of the rectangle
    public double getArea() {
        return length * width; // Calculate the area by multiplying length and width
    }

    // Method to calculate and return the perimeter of the rectangle
    public double getPerimeter() {
        return 2 * (length + width); // Calculate the perimeter using the length and width
    }
}

```

The above code represents a class called 'Rectangle' that extends the 'Shape' abstract class. The 'Rectangle' class has two instance variables, 'length' and 'width', and a constructor that initializes these variables. It also has methods to calculate the area and perimeter of a rectangle using the 'getArea()' and 'getPerimeter()' methods respectively. The 'getArea()' method returns the product of the length and width variables while the 'getPerimeter()' method returns twice the sum of the length and width variables.

```

// Circle.java

// Define the Circle class, which extends the Shape class
public class Circle extends Shape {

    // Private field to store the radius of the circle
    private double radius;

    // Constructor to initialize the radius of the circle
    public Circle(double radius) {
        this.radius = radius; // Set the radius field to the provided radius
    }

    // Method to calculate and return the area of the circle
    public double getArea() {
        return Math.PI * radius * radius; // Calculate the area using the radius
    }

    // Method to calculate and return the perimeter (circumference) of the circle
    public double getPerimeter() {
        return 2 * Math.PI * radius; // Calculate the perimeter using the radius
    }
}

```

```
        public double getPerimeter() {  
            return 2 * Math.PI * radius; // Calculate the perimeter using the t  
        }  
    }  
}
```

The "Circle" class is a subclass of the abstract "Shape" class. It has a private instance variable for the radius of the circle. It implements abstract methods for calculating the area and perimeter of the circle. The area is calculated by multiplying the square of the radius by pi. The perimeter is calculated by multiplying the radius by twice pi value.

```
// Triangle.java  
  
// Define the Triangle class, which extends the Shape class  
public class Triangle extends Shape {  
  
    // Private fields to store the sides of the triangle  
    private double side1;  
    private double side2;  
    private double side3;  
  
    // Constructor to initialize the sides of the triangle  
    public Triangle(double side1, double side2, double side3) {  
        this.side1 = side1; // Set the side1 field to the provided side1  
        this.side2 = side2; // Set the side2 field to the provided side2  
        this.side3 = side3; // Set the side3 field to the provided side3  
    }  
  
    // Method to calculate and return the area of the triangle  
    public double getArea() {  
        double s = (side1 + side2 + side3) / 2; // Calculate the semi-perim  
        return Math.sqrt(s * (s - side1) * (s - side2) * (s - side3)); // C  
    }  
  
    // Method to calculate and return the perimeter of the triangle  
    public double getPerimeter() {  
        return side1 + side2 + side3; // Calculate the perimeter by summing  
    }  
}
```

The "Triangle" class is a subclass of the "Shape" abstract class that represents a triangle shape. It has three instance variables to store the length of its three sides. It has a constructor that

takes these three sides as parameters. The class provides implementations of the abstract methods "getArea()" and "getPerimeter()" to calculate the area and perimeter of the triangle based on its three sides using standard formulas.

```
// Define the Main class
public class Main {

    // The main method - the entry point of the Java application
    public static void main(String[] args) {

        // Create a rectangle shape with length 10 and width 12
        Shape rectangle = new Rectangle(10, 12);

        // Create a circle shape with radius 5
        Shape circle = new Circle(5);

        // Create a triangle shape with sides 7, 8, and 6
        Shape triangle = new Triangle(7, 8, 6);

        // Print the header for the area and perimeter calculations
        System.out.println("\nArea and perimeter of various shapes:");

        // Print the details and calculations for the rectangle
        System.out.println("\nRectangle: Length-10, Width-12");
        System.out.println("Area: " + rectangle.getArea());
        System.out.println("Perimeter: " + rectangle.getPerimeter());

        // Print the details and calculations for the circle
        System.out.println("\nCircle: Radius-5");
        System.out.println("Area: " + circle.getArea());
        System.out.println("Perimeter: " + circle.getPerimeter());

        // Print the details and calculations for the triangle
        System.out.println("\nTriangle: Side1-7, Side2-8, Side3-6");
        System.out.println("Area: " + triangle.getArea());
        System.out.println("Perimeter: " + triangle.getPerimeter());
    }
}
```

The Main class creates instances of different shape objects including a rectangle, circle, and triangle. It then calls methods to calculate the area and perimeter of each shape and displays

the results in the console.

Sample Output:

Area and perimeter of various shapes:

Rectangle: Length-10, Width-12

Area: 120.0

Perimeter: 44.0

Circle: Radius-5

Area: 78.53981633974483

Perimeter: 31.41592653589793

Triangle: Side1-7, Side2-8, Side3-6

Area: 20.33316256758894

Perimeter: 21.0

Flowchart: