/ Define a class representing a geometric shape

class Shape {

// Attributes

private String color;

// Constructor

public Shape(String color) {

this.color = color;

}

// Getter method for color

public String getColor() {

return color;

}

// Method to calculate area (to be overridden by subclasses)

public double calculateArea() {

return 0.0;

}

// Method to display information about the shape

public void displayInfo() {

System.out.println("Color: " + color);

System.out.println("Area: " + calculateArea());

}

}

// Define a subclass representing a circle

class Circle extends Shape {

// Additional attribute

private double radius;

// Constructor

public Circle(String color, double radius) {

super(color);

this.radius = radius;

}

// Override calculateArea method for circles

@Override

public double calculateArea() {

return Math.PI \* radius \* radius;

}

}

// Define a subclass representing a rectangle

class Rectangle extends Shape {

// Additional attributes

private double length;

private double width;

// Constructor

public Rectangle(String color, double length, double width) {

super(color);

this.length = length;

this.width = width;

}

// Override calculateArea method for rectangles

@Override

public double calculateArea() {

return length \* width;

}

}

public class ObjectOrientedDemo {

public static void main(String[] args) {

// Create objects of Circle and Rectangle classes

Circle myCircle = new Circle("Red", 5.0);

Rectangle myRectangle = new Rectangle("Blue", 4.0, 6.0);

// Demonstrate polymorphism

Shape shape1 = myCircle;

Shape shape2 = myRectangle;

// Display information about the shapes

System.out.println("Information about Circle:");

myCircle.displayInfo();

System.out.println("\nInformation about Rectangle:");

myRectangle.displayInfo();

// Demonstrate encapsulation

System.out.println("\nColor of Circle: " + myCircle.getColor());

System.out.println("Color of Rectangle: " + myRectangle.getColor());

// Demonstrate inheritance

System.out.println("\nShape 1 Information (Polymorphism):");

shape1.displayInfo();

System.out.println("\nShape 2 Information (Polymorphism):");

shape2.displayInfo();

}

}