

L07

Shape.java

```
public abstract class Shape {
    private String shapeName;

    public Shape() {
        shapeName = "Generic Shape";
    }

    public Shape(String shapeName) {
        this.shapeName = shapeName;
    }

    public String getShapeName() {
        return shapeName;
    }

    public abstract double findPerimeter();

    public abstract double findArea();

    public boolean equals(Object other) {
        if (other instanceof Shape) {
            Shape temp = (Shape) other;
            return this.shapeName.equals(temp.shapeName);
        }
        return false;
    }

    public String toString() {
        return "Shape Name: " + getShapeName();
    }
}
```

Circle.java

```
public class Circle extends Shape {  
    private double radius;  
  
    public Circle(double radius) {  
        super("Circle");  
        this.radius = radius;  
    }  
  
    public double findPerimeter() {  
        return 2 * Math.PI * radius;  
    }  
  
    public double findArea() {  
        return Math.PI * radius * radius;  
    }  
  
    public boolean equals(Object other) {  
        if (other instanceof Circle) {  
            Circle temp = (Circle) other;  
            return super.equals(temp) &&  
                (this.radius == temp.radius);  
        }  
        return false;  
    }  
  
    public String toString() {  
        return super.toString() + " radius: " + radius;  
    }  
}
```

Ellipse.java

```

public class Ellipse extends Shape {
    private double majorAxis, minorAxis;

    public Ellipse(double majorAxis, double minorAxis) {
        super("Ellipse");
        this.majorAxis = majorAxis;
        this.minorAxis = minorAxis;
    }

    public double findPerimeter() {
        return 2 * Math.PI * Math.sqrt(((Math.pow(majorAxis, 2) + Math.pow(minorAxis, 2)) / 2));
    }

    public double findArea() {
        return Math.PI * majorAxis * minorAxis;
    }

    public boolean equals(Object other) {
        if (other instanceof Ellipse) {
            Ellipse temp = (Ellipse) other;
            return super.equals(temp) && (this.majorAxis == temp.majorAxis) && (this.minorAxis ==
temp.minorAxis);
        }
        return false;
    }

    public String toString() {
        return super.toString() + " major axis: " + majorAxis + " minor axis: " + minorAxis;
    }
}

```

Rectangle.java

```

public class Rectangle extends Shape {
    private double length, width;

    public Rectangle(double length, double width) {
        super("Rectangle");
    }
}

```

```

        this.length = length;
        this.width = width;
    }

    public double findPerimeter() {
        return 2 * length + 2 * width;
    }

    public double findArea() {
        return length * width;
    }

    public boolean equals(Object other) {
        if (other instanceof Rectangle) {
            Rectangle temp = (Rectangle) other;
            return super.equals(temp) && (this.length == temp.length) && (this.width == temp.width);
        }
        return false;
    }

    public String toString() {
        return super.toString() + " length: " + length + " width: " + width;
    }
}

```

ShapeTest.java

```

public class ShapeTest {
    public static void main(String[] args) {
        Shape[] myShapes = new Shape[10];
        Circle myCircle = new Circle(10.0);
        double expected = 314.16;
        System.out.println("Circle Area: " + myCircle.findArea());

        myShapes[0] = new Circle(10.0);
        expected = 314.16;
        System.out.println("Circle in Shapes Array Area: " + myShapes[0].findArea());

        myShapes[1] = new Ellipse(5.0, 10.0);
        expected = 157.08;
    }
}

```

```
System.out.println("Ellipse in Shapes Array Area: " + myShapes[1].findArea());

myShapes[2] = new Rectangle(5.0, 10.0);
expected = 50.0;
System.out.println("Rectangle in Shapes Array Area: " + myShapes[2].findArea());
}
}
```

Revision #1

Created 24 April 2025 22:20:01 by Brandon Duke

Updated 24 April 2025 22:20:07 by Brandon Duke