

## Selected files

### 5 printable files

Lab 3\2A\Circle.java  
Lab 3\2A\Rectangle.java  
Lab 3\2A\Shape.java  
Lab 3\2A\Square.java  
Lab 3\2A\TestShape.java

#### Lab 3\2A\Circle.java

```
1 public class Circle extends Shape {
2     // Instance variable
3     private double radius;
4
5     // Constructors
6     public Circle() {
7         super();
8         this.radius = 1.0;
9     }
10
11     public Circle(double radius) {
12         super();
13         this.radius = radius;
14     }
15
16     public Circle(double radius, String color, boolean filled) {
17         super(color, filled);
18         this.radius = radius;
19     }
20
21     // Getter and setter for radius
22     public double getRadius() {
23         return radius;
24     }
25
26     public void setRadius(double radius) {
27         this.radius = radius;
28     }
29
30     // Methods
31     public double getArea() {
32         return Math.PI * radius * radius;
33     }
34
35     public double getPerimeter() {
36         return 2 * Math.PI * radius;
37     }
38
39     // Override toString method
40     @Override
41     public String toString() {
42         return "A Circle with radius=" + radius + ", which is a subclass of " +
43         super.toString();
44     }
45 }
```

#### Lab 3\2A\Rectangle.java

```
1 public class Rectangle extends Shape {
2     // Instance variables
3     private double width;
4     private double length;
5
6     // Constructors
7     public Rectangle() {
8         super();
9         this.width = 1.0;
10        this.length = 1.0;
11    }
12
13    public Rectangle(double width, double length) {
14        super();
15        this.width = width;
16        this.length = length;
17    }
18
19    public Rectangle(double width, double length, String color, boolean filled) {
20        super(color, filled);
21        this.width = width;
22        this.length = length;
23    }
24
25    // Getter and setter for width and length
26    public double getWidth() {
27        return width;
28    }
29
30    public void setWidth(double width) {
31        this.width = width;
32    }
33
34    public double getLength() {
35        return length;
36    }
37
38    public void setLength(double length) {
39        this.length = length;
40    }
41
42    // Methods
43    public double getArea() {
44        return width * length;
45    }
46
47    public double getPerimeter() {
48        return 2 * (width + length);
49    }
50
51    // Override toString method
52    @Override
53    public String toString() {
54        return "A Rectangle with width=" + width + " and length=" + length + ", which is a
55        subclass of "
56            + super.toString();
57    }
58 }
```

```
1 public class Shape {
2     // Instance variables
3     private String color;
4     private boolean filled;
5
6     // Constructors
7     public Shape() {
8         this.color = "green";
9         this.filled = true;
10    }
11
12    public Shape(String color, boolean filled) {
13        this.color = color;
14        this.filled = filled;
15    }
16
17    // Getters and setters
18    public String getColor() {
19        return color;
20    }
21
22    public void setColor(String color) {
23        this.color = color;
24    }
25
26    public boolean isFilled() {
27        return filled;
28    }
29
30    public void setFilled(boolean filled) {
31        this.filled = filled;
32    }
33
34    // toString method
35    @Override
36    public String toString() {
37        String filledStatus = filled ? "filled" : "Not filled";
38        return "A Shape with color of " + color + " and " + filledStatus;
39    }
40 }
```

### Lab 3\2A\Square.java

```
1 public class Square extends Rectangle {
2     // Constructors
3     public Square() {
4         super();
5     }
6
7     public Square(double side) {
8         super(side, side);
9     }
10
11    public Square(double side, String color, boolean filled) {
12        super(side, side, color, filled);
13    }
14
15    // Override toString method
```

```
16     @Override
17     public String toString() {
18         return "A Square with side=" + getWidth() + ", which is a subclass of " +
super.toString();
19     }
20
21     // Override setWidth and setLength to maintain square geometry
22     @Override
23     public void setWidth(double side) {
24         super.setWidth(side);
25         super.setLength(side);
26     }
27
28     @Override
29     public void setLength(double side) {
30         super.setWidth(side);
31         super.setLength(side);
32     }
33 }
```

### Lab 3\2A\TestShape.java

```
1 public class TestShape {
2     public static void main(String[] args) {
3         // Test Shape class
4         Shape shape1 = new Shape();
5         System.out.println(shape1);
6
7         Shape shape2 = new Shape("blue", false);
8         System.out.println(shape2);
9
10        // Test Circle class
11        Circle circle1 = new Circle();
12        System.out.println(circle1);
13        System.out.println("Area: " + circle1.getArea());
14        System.out.println("Perimeter: " + circle1.getPerimeter());
15
16        Circle circle2 = new Circle(3.5, "red", true);
17        System.out.println(circle2);
18        System.out.println("Area: " + circle2.getArea());
19        System.out.println("Perimeter: " + circle2.getPerimeter());
20
21        // Test Rectangle class
22        Rectangle rectangle1 = new Rectangle();
23        System.out.println(rectangle1);
24        System.out.println("Area: " + rectangle1.getArea());
25        System.out.println("Perimeter: " + rectangle1.getPerimeter());
26
27        Rectangle rectangle2 = new Rectangle(2.5, 4.0, "yellow", false);
28        System.out.println(rectangle2);
29        System.out.println("Area: " + rectangle2.getArea());
30        System.out.println("Perimeter: " + rectangle2.getPerimeter());
31
32        // Test Square class
33        Square square1 = new Square();
34        System.out.println(square1);
35        System.out.println("Area: " + square1.getArea());
36        System.out.println("Perimeter: " + square1.getPerimeter());
37    }
```

```
38 |         Square square2 = new Square(5.0, "purple", true);
39 |         System.out.println(square2);
40 |         System.out.println("Area: " + square2.getArea());
41 |         System.out.println("Perimeter: " + square2.getPerimeter());
42 |     }
43 | }
44 |
```