

TUTORIAL 1: Examining a Java class

The following code defines a Java class. Read through the code carefully and try to answer the questions overleaf:

```
public class Circle
{
    private String label; // a label to display on the circle
    private int x, y;     // coordinates of centre of circle
    private int radius;   // radius of circle

    public Circle(String label, int x, int y, int radius)
    {
        this.label = label;
        this.x = x;
        this.y = y;
        this.radius = radius;
    }

    public int getRadius()
    {
        return radius;
    }

    public void setRadius(int radius)
    {
        this.radius = radius;
    }

    public String getLabel()
    {
        return label;
    }

    public void move(int dx, int dy)
    {
        x = x + dx;
        y = y + dy;
    }

    public double area()
    {
        return Math.PI * radius * radius;
    }
}
```

1. What do you think might be the purpose of this class?
2. List the instance variables and methods of the **Circle** class
3. What is the effect of the keyword **private** associated with the instance variables?
4. What do you think is the purpose of the methods **getRadius** and **setRadius**?
5. There is a **getLabel** method, but no **setLabel** method – what is the result of this?
6. How can the values of **x** and **y** be changed?
7. What does the word **this** refer to in the constructor of the class? Why is it used here?
8. Which *one* of the following is a valid way of creating a **Circle** object?
 - a. `Circle myCircle = new Circle();`
 - b. `Circle myCircle = new Circle("MY CIRCLE", 5, 5, 3);`
 - c. `Circle myCircle = new Circle("MY CIRCLE", 5, 5);`
 - d. `Circle myCircle = new Circle(5, 5, 3, "MY CIRCLE");`
9. Describe the signature of the **move** method.
10. Assuming a **Circle** object called **myCircle** has been instantiated, which of the following are valid ways of using the object?
 - a. `myCircle.move(2,3);`
 - b. `int moved = myCircle.move(2,3);`
 - c. `myCircle.move(5);`
 - d. `int myArea = myCircle.area();`
 - e. `double myArea = myCircle.area();`
 - f. `double myArea = myCircle.area(5);`
 - g. `String myLabel = myCircle.getLabel();`
 - h. `myCircle.setLabel("MY MODIFIED CIRCLE");`
 - i. `myCircle.radius = 5;`
 - j. `myCircle.setRadius(5);`
11. You decide that you would like **Circle** objects to be created by default with centre at (0,0) and radius of 5. Write down an additional constructor which would allow this, and a statement which would instantiate a **Circle** object with default values.