

Object-Oriented Programming: Basics

1. Make a new Eclipse project called “shapes1”. Create a Circle class that contains a radius field. Give it a constructor where you pass in the radius. Have your test routine create a few circles, assign a value to the radius, then print out some information about the circles. Put your Circle class and your test routine in two separate classes, like this:

- **Circle.java**

```
public class Circle {  
    public double radius;  
    ...  
}
```

- **CircleTest.java**

```
public class CircleTest {  
    public static void main(String[] args) {  
        Circle c = new Circle(...);  
        ...  
    }  
}
```

Notes

- The Circle class does not have a main, so you cannot execute it directly. You only directly run the CircleTest class (R-click, Run As, Java Application).
 - This problem requires several different capabilities. Unless you have previous Java experience, I strongly recommend you build up to the solution in a piecemeal fashion. First make a Circle class with a radius field only, and test it out from the main in CircleTest. Then add in a constructor. Then test it out again. And so on.
2. Give your Circle a getArea method that calculates its area, and a printInfo method that prints out the radius and area. Make a test case that tries these capabilities out.
 3. Make a program that creates an array of 100 circles, each with a random radius. Print out the sum of the areas of the 100 circles. Also print the biggest and smallest areas.

Object-Oriented Programming: Basics (Continued)

4. Create a Rectangle class that contains width and height fields. Also give it a `getArea` method. Again, make a few test cases.
5. Create a Square class with width and `getArea`. Then, give both Square and Circle `setArea` methods that let you specify a desired area. Make a few test cases.
6. Questions to ponder:
 - Suppose you create a method that takes a Rectangle as an argument. Now suppose you want to pass a Square to it (after all, squares are rectangles, aren't they?). Why won't it work? From what we know so far, how could you fix this problem?
 - Since there is no particular relationship among Circle, Square, and Rectangle, what would you do if you wanted to make an array of mixed shapes, then loop down the array and sum up the areas?