

APIT - Laboratory Session 2

Dr. Simon Rogers

Introduction and Aims

The purpose of this lab is to give you some time to familiarise yourself with some of the topics we've looked at so far. Also, if you haven't compiled and run Java from the command line, this would be a good time to try.

Tasks

1. Design and implement a class hierarchy that describes different shapes (squares, rectangles, circles, triangles etc). Each shape should store its position (in 2D, i.e. x and y co-ordinates), its size and its color and should have methods to move it (i.e. change its position) and compute its area and describe itself. You can use inheritance, abstract classes, interfaces. **Note:** you don not have to draw the shapes, just create the classes to represent their features.
2. Compile and run the `CallExamples.java` class (it's in the `APIIT/recap/CallByReference` folder). Are any of the outputs unexpected? If so, refer to the notes and try and work out why it does what it does!
3. `CallExamples.java` uses *inner classes* (also known as *nested* classes). These are classes defined inside another class. In this example, they are declared **static**. Why? What happens if you remove the **static** keyword? Why?
4. Assertions: Write a class defining a circle that takes as its input a radius (of type `Double`) and has a method that computes the area of the circle. Use an assertion to check that the radius provided is positive. Create a class with a main method uses `Circle` objects and try creating them with both positive and negative radius, with assertions switched on and switched off.
5. Comment your `Circle` class in the manner accepted by JavaDoc and compile a JavaDoc page for the class (instructions in slides)

Remember that the aim here is not to produce a perfect design, just to force you to try some things that you might not have tried before (inheritance, abstract classes, interfaces, protected methods/fields etc) and to give you a chance to grapple with the command line.