

Exercises:

Object-Oriented Programming: More Capabilities

Copy your first shapes Eclipse project and give it a new name. (Or, make a new project, and then copy any needed files from your first shapes project or from my “oop-basics-exercises” project). After you do the copying, you might want to close the first shapes project (R-click on the project in the Package Explorer on the left side and choose “Close project”) so that you don’t accidentally edit classes from the wrong project.

- 1.** Change your Circle class so that the fields are private and you have getXxx and setXxx methods to lookup and change the values of the fields. If you have Rectangle and Square classes, change them as well. Remember that Eclipse can insert the getter and setter methods for you, if you want it to (see the Source menu). Run some test cases where you create a few shapes and then access their areas.
- 2.** Add some simple JavaDoc comments above each class, constructor, and public method. Run JavaDoc from Eclipse (select the project, then go to Project menu and choose “Generate Javadoc...”). Once the documentation is created, look at it by opening the doc folder and dragging index.html onto your browser.
- 3.** If you haven’t already made a Square class, do so. Make your Square inherit from Rectangle, but still enforce the restriction that the width and the height are the same. Hint: override some method(s). You will find this problem to be a bit ugly, because you have two competing interests. On the one hand, you want squares to *be* rectangles because they are in real life. But on the other hand, the Rectangle class has separate width and height accessor methods that you can’t totally get rid of in Square. Whatever solution you come up with, be sure it is never possible for the width of a Square to be different from its height.

Run some test cases where you create squares, change their width/height, and then print out the various parameters.

- 4.** Add a toString method to each of Circle, Rectangle, and Square. Have the toString method show the relevant parameters of the object. Print out some instances of your classes, but do *not* explicitly call toString in your code.