CS 158 Lab 2 January 19, 2010
Shapes

1. Start a new project in Netbeans called Shapes. Open a browser, go to the class webpage: http://faculty.valpo.edu/jcaristi/cs158 and follow the link to "Shapes". Create each file as a class in your project. I suggest you create each by opening the file in the browser, copying everything except the "package" line, and pasting that into a class in your project with that same name. To avoid compilation errors, you should download them in this order: Shape, Rectangle, Circle, ThreeD, RectangularPrism. Get everything to compile cleanly. Read each class and try to understand what each part does. Ask if you have any questions.

2. Add a test class for **Rectangle**. Test the **getPerimeter**, **getSurfaceArea**, **toString**, and **equals** methods using nontrivial tests, and get them all to pass (since most of these methods use **double**s, use the version of **assertEquals** that has three arguments, the last one being the "tolerance". A tolerance of 0.01 should be good enough).

3. It seems reasonable to me that a **Rectangle** that is 3 by 4 is the same as one that is 4 by 3. Add a test (that will fail) that asserts this, then change the **equals** method in **Rectangle** so that the test passes.

4. Create a new class called **Square** that is a subclass of **Rectangle**. This is a ***bad idea***, and the purpose of the next few steps is to show you why. Notice how the **Rectangle** class constructors work. For the **Square** class, create appropriate constructors (one with no arguments and one with one argument) that create squares in the obvious way. Override the **setHeight** and **setLength** methods so that each of them sets both variables. This is necessary so that if someone creates a **Square** object, and then someone else sends it a **setHeight** message, the object will still remain a square.

5. In the **Rectangle** class, add a new method called **setUpABox** that returns nothing, but takes a **Rectangle** argument. All this method should do is to set the height of the Rectangle passed in equal to 2 and the length of the **Rectangle** passed in equal to 4.

6. Add a test for **setUpABox** that creates a **Rectangle** object that is 5 by 5. Assert that that rectangle's surface area is 25. Pass that object to the **setUpABox** method, and assert that the surface area of the **Rectangle** passed in is now 8. Your tests should all pass.

7. Now create a **Square** object in that same test that has a side length of 5. Assert that its surface area is 25. The tests should pass. Now pass that **Square** object to **setUpABox**, and assert that its surface area is 8. That should now fail. You should also observe that the surface area is not 25 either. This is a problem. We must always be able to substitute a subclass instance for any instance of a parent class in a method, and here we cannot. This is an important principle in object oriented software design called the Liskov Substitution Principle: if S is a subtype of T, then objects of type T in a program may be replaced with objects of type S without altering any of the desirable properties of that program (e.g., correctness).

8. Create a **Cylinder** class that follows the pattern used in **RectangularPrism**. Notice that both a **Cylinder** and a **RectnagularPrism** implement an interface called **ThreeD**. We will be talking about interfaces a lot this semester. For now, all you have to know is that **Cylinder** must state that it **implements ThreeD**. Of course, a cylinder only needs two parameters: a radius and a height. Add a test class with a test for the **getVolume** method. The volume of a cylinder is $\pi * r^2 * h$ where r is the radius and h is the height. You do NOT have to implement the **getPerimeter** method.

There is nothing to turn in for this lab. However, make sure you finish everything and that you understand everything and have no unanswered questions before you go on to subsequent labs.