CS212: Paradigms
Dr. Gross, Spring 2016

Lab 3 - A Round Peg Into a Square Hole

1. Open your lab project's src folder cs212-lab-sp16
2. Create a new UML file
    a. Name it shapes-lastname-lastname.uml
    b. Save it in your lab projects cs212-lab-sp16\Documents\Lab03
3. Click on that file in NetBeans and go to Team->Add
4. Go to Team->Commit… and enter "created UML file"
5. Make your design
6. Create a new package called edu.blackburn.cs.cs212.shapes.lastnamelastname

We're going to make shapes. No need to have this be interactive; it's fine to just create the various classes & objects. You are going to go back and forth between design (class diagram) and code. This is normal. You'll get better at design as you go along.

First we need measurement class to hold each measurement we make. Why? What does it need to know and/or do?

Second, make a Rectangle class and a Square class. They should both have public constructors (in which the parameters of the shape are set) and the methods:

```
public Measurement getArea()
public Measurement getPerimeter()
```

Which is the superclass, and which is the subclass? How do their constructors differ? Have the constructor for Rectangle print out "New Rectangle: X by Y" (where X and Y are base and height), and have the constructor for Square print out "New Square: X" (where X is the length of one side). Create some of each and call the two methods above, printing out the results.

Third, now for some fun. We're going to make an Ellipse class and a Circle class. Again, which is superclass and which subclass? How do their constructors differ? An ellipse has two focal points (we'll call them A and B), each the same distance from the center. How many focal points does a circle have? Don't worry about anything else yet. Have the constructor for Ellipse print out "New Ellipse" (where X and Y are base and height), and have the constructor for Square print out "New Circle" (where X is the length of one side). Create some of each.

Fourth, add in the methods:

```
public Measurement getPerimeter()
public Measurement getArea()
```

These methods are *much, much* easier to implement for one of these classes than for the other. You probably remember how. Remember you can stub out methods you don't know how to complete just yet. Properly implement the methods for the easier version, but you can leave the harder methods stubbed out. Call the two methods on these objects you created earlier, and print the results.

Fifth, one of the purposes of inheritance is that it allows us to reuse code and eliminate repetition, and you should see that above. However, there's no repeated code between Rectangle/Square and Ellipse/Circle, right? There's no way to have code that will work for both. Let's introduce a new method, though. We'll call this method:

```
public double getEfficiency()
```

Efficiency is determined by a ratio of the area to the perimeter. The more square meters per linear meters leads to a higher efficiency. How do you redesign? Where would the getEfficiency() code go? You won't be able to implement this yet, but you can design it.

Finally, still have time? Try to make Triangle, Isosceles Triangle, and Equilateral Triangle. Which one is easiest to implement the area and perimeter methods?

Some hints:

1. Don't worry about a command interface; I know you can use Scanner
2. Write test code and comment what it should do and what it proves if it works
3. Design a bit, code a bit, test a bit
4. If a method is longer than 10 lines, use Refactor->Extract Method or otherwise create a new submethod
5. GOTO 2

To save your work:

1. When you add new files (or think you might have) go to Team->Add
    a. THIS OFTEN FAILS!
2. Every few minutes, when the code compiles, go to Team->Commit
3. When you are ready to upload your code (at least every 30 mins), go to Team->Remote->Push to Upstream
4. DON'T BREAK THE BUILD (do not push code that does not compile)
5. You can always check the Github website to verify that everything is there and current