CS 251
Lab Exercise 03

Main topics: Writing Classes
Declaring / Using Instance Variables
Writing Instance Methods
Accessors and Mutators
public vs private
Constructors
static vs non-static

Exercise

This week we will be practicing writing a "complete" class, starting from where you left off last week, which includes all of the standard mutators as well as an instance count. In addition careful attention to the `public`, `private` and `static` specifiers will be made.

Getting Started To start this exercise, you should:

1. Open eclipse and start a new Java project named `Lab03`
2. Add a Class (named `Circle`) to this project, and copy the contents of the `Circle` file provided into it.
3. Add a Class (named `CircleDriver`) to this project, and copy the contents of the `CircleDriver` file provided into it

Requirements

Circle.java A simple class which models a circle by its one defining characteristic, which is its radius. This class is not complete and must be modified as such:

1. All class variables must be private
2. The initial instance count must be 0
3. Include a public accessor for the instance count
4. Include a public mutator for the instance count
5. By default all instances have a radius of 1.0
6. Include the standard default constructor
   All access of instance data by this constructor must be made via the the specifying constructor.
7. Include the standard specifying constructor
   All access of class and instance data by this constructor must be made via the accessors and mutators.
8. Include the standard copy constructor All access of instance data by this constructor must be made via the the specifying constructor.
9. Modify the `Circle clone()` method so that it makes use of the copy constructor, instead of using the default constructor, followed by the resize() method.

CircleDriver.java A simple *driver* class to test the Circle class. This class is not complete and must be modified as such:

1. Add / modify lines of code and "test" that each of your constructors are functioning properly.
2. Add / modify lines of code and "test" that your instance count is being maintained properly.

Once you have completed the requirements:

1. Make sure that your program runs without errors or warnings.
2. Run your program enough times to verify its correctness.
3. If it runs correctly, then see your TA for a check-off.