**CS 4322 – Coding Assignment 11**

See Schedule for due date. You can work individually or in groups of two.

**Overview**

You will write a *Circle* class that implements the *Shape* interface. However, much of the functionality specified by *Shape* exists in the *CircleShape* class. Thus, you will implement *Circle* as an adapter for the *CircleShape* class. You have been provided with a zip file that contains:

* A *CircleTester* class that contains test code which should produce this output:

Circle: radius=5.00, location=(3,4), area=78.54, circumference=31.4

Circle: radius=5.00, location=(6,7), area=78.54, circumference=31.4

Circle: radius=10.00, location=(1,2), area=314.16, circumference=62.8

areaDifference=-235.6

Circle: radius=5.59, location=(6,7), area=98.17, circumference=35.1

* A jar file, *shape\_stuff.jar* that contains the class files for these classes: *CircleShape, Point,* and *Shape*. You should inspect these files in Eclipse to see what members exist. *Shape* is an interface and the *Circle* class you write must implement it. *Shape* specifies these methods:

|  |  |
| --- | --- |
| **Behavior** | **Description** |
| **double** area() | Returns the area of the shape |
| **double** circumference() | Returns the circumference of the shape |
| **void** moveTo(Point location) | Moves the shape to a new location |
| Point location() | Returns the location of the shape as a *Point* object. Note that the *CircleShape* class does not utilize *Point.* Thus, this method will need to create and return it. |
| **double** areaDifference(Shape shape) | Returns the difference in area between the implementing shape and the argument. |
| **void** enlargeArea(**double** percent) | Enlarges the shape’s area by the amount of the argument specified as a percent. For example, use 25.4 to represent 25.4%. Implementing this requires a tiny bit of thought and a tiny bit of algebra. Note that the radius in *CircleShape* cannot be changed. Thus, this method will need to create a new *CircleShape*. |

Requirements for the *Shape* class

1. The constructor should have the signature below. Note that it accepts the x,y location of the circle and the diameterof the circle. Note that the *CircleShape* class accepts a radius.
2. There should be exactly one instance variable which is the adaptee. No other instance variables are allowed.
3. You should provide a *toString* that provides a string like this:

Circle: radius=5.00, location=(6,7), area=78.54, circumference=31.4

Notes about the *CircleShape* class:

1. There is one constructor and it accepts the radius of the circle. Other methods can be used to set the x,y location.

**Requirements**

1. Log your time on the Time Log
2. Write the *Circle* class and verify that your results are correct by running against *CircleTester*.

**Steps to Complete**

1. Unzip the supplied file.
2. Open Eclipse, create a project, and create a package named *ca11*.
3. Drag *shape\_stuff.jar* and *CircleTester.java* into this package.
4. Right-click *shape\_stuff.jar* and choose: Build Path, Add to Build Path. This will move *shape\_stuff.jar* to a Referenced Libraries node in the Package Explorer. Expand *shape\_stuff* and you will see the classes and available members.
5. Write the *Circle* class.

**Deliverable**

1. Zip the *ca11* package and Time Log and submit on Blazeview.

**Time Log**

If working in a group, submit a single time log ONLY if you worked face-to-face for the entire assignment; Otherwise submit one for each member (copy table below).

* **Delete empty rows, add if needed.**
* **Put the total time at bottom.**

|  |  |
| --- | --- |
| **Time (hrs)** | **Task** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  | **Total Time (hrs)** |