CSE/IT 213: Homework 1

Due: Wednesday, Feb. 7 @ 11:59pm
This homework in individual.

Problem

For this homework, you need to write 3 classes: Test, Rectangle, and a Sphere class. Remember to use a package for your source code. Name the package following the rules outlined in homework 0. For this homework, you will have one package which contains 3 classes.

UML

The classes are described using the UML modeling language. UML uses a simple schema to model classes.

Following the name of the method or constructor, parameters are included in the (). The method's return type follows a colon. For example

`Rectangle(width : double, height : double)`

is a public constructor that takes two double parameters. You know its a constructor as it has the same name as the class followed by parentheses.

`area() : double`

is a method that takes no parameters and returns a double.

The organization of the UML model is:

```
Class Name
----------------
Instance fields
----------------
Constructors
----------------
Methods
```

Often is the case that the instance fields and constructors are left off the UML diagram as well as setters and getters for the instance fields.

Some authors precede the names with a + or − to distinguish public (+) from private (−) instance fields and methods. However, often that detail is omitted as it is here.

Test class

The Test class contains the `main()` method whose purpose is to test the other classes. Make sure you test all constructors and methods of the Rectangle and Sphere class. Test with what you think is appropriate data.

Rectangle class

```
Rectangle
---------------------
Rectangle()
Rectangle(width : double, height : double)
Rectangle(x : double, y : double, width : double, height : double)
---------------------
area() : double
perimeter() : double
diagonalLength() : double
distanceFromOrigin() : double
```

The Rectangle class has three constructors: 1) a default constructor that defaults to a unit square of width one and a height of one with the lower left hand corner located at the origin and the upper right hand corner at point (1,1); 2) a constructor that sets the width and height of the rectangle, but keeps the default location of the lower left hand corner of the rectangle; and 3) a constructor that can set the width, height and, position of the lower left hand corner of the rectangle.

Methods of the Rectangle class are: find the area of the rectangle, its perimeter, the length of the diagonal of the rectangle, and the distance the lower left hand corner of the rectangle is from the origin.

Make sure you have setters and getters for all instance fields.

Sphere Class

The Sphere class is similar to the Rectangle class.

```
Sphere
```

```
---------------------
Sphere()
Sphere(radius : double)
Sphere(x : double, y : double, z : double)
Sphere(x : double, y : double, z : double, radius : double)
---------------------
volume() : double
surfaceArea() : double
distanceFromOrigin() : double
```

The sphere class has four constructors: 1) the default constructor defaults to a sphere of radius 1 centered at the origin (0, 0, 0); 2) a constructor that sets the radius, but keeps the sphere centered at the origin; 3) a constructor that sets the center position of the sphere but keeps the radius one; and 4) a constructor that sets the position and the radius.

Methods of the Sphere class are: find the volume of the sphere, find the surface area of the sphere, and find the distance the center of the sphere is to the origin. Use `Math.PI` for the value of `PI`.

Make sure you have getters and setters for all instance fields.

Comments

Make sure all classes and methods are commented. Follow these simple rules.

In general, the rule for comments is that comments explain *why*, not the *how*.

Every class begins with the class comment that was described in homework 0.

In general, test classes need a class comment and, if needed, one liners about the main method's logic.

Instance fields should have a comment after its declaration, stating what it is used for if its purpose is not clear from its name.

```
int foo; //a brief description of foo
```

Every method, except `main()`, has the following comment structure:

```
/**
* A brief description of what fooBar does.
*
* @param foo a description of foo
* @param bar a description of bar
* @return a description what is returned.
```

```
*
* Remarks: Any remarks, assumptions, etc goes here.
* The @return tag is not used if the method returns void
*/
public int fooBar(int bar, String bar) {
 //statements

}
```

NB that *all* parameters are commented for methods using the `@param` tag.

Java Coding Style

Except for 4 space indentation, follow Google's Java Style guide, which is available at `http://google-styleguide.googlecode.com/svn/trunk/javaguide.html`

Submission

Create a jar file of your 3 classes named

`cse213_firstname_lastname_hw1.jar`

Upload the jar file to Canvas before the due date.