

CIS 212 Assignment 3: 100 points

The goal of this assignment is to gain basic Object-Oriented Programming experience working with Java classes and interfaces.

For this assignment, you'll implement a hierarchy of Java classes that share a common interface. You'll then be able to create a data structure using this interface type and populate it with various instances of classes inheriting from the interface.

1. [10] Implement a new interface named `Measurable` with a single method named `getArea()` which takes no arguments and returns a double.
2. [10] Implement a new class named `Rectangle` that implements `Measurable`. Provide a class constructor which takes arguments appropriate for creating a 2D rectangle and implement the `getArea()` method to return the area of the rectangle.
3. [10] Implement a new class named `Box` that extends `Rectangle`. Provide a class constructor which takes arguments appropriate for creating a 3D box and implement the `getArea()` method to return the total surface area of the box.
4. [10] Implement a new class named `Circle` that implements `Measurable`. Provide a class constructor which takes arguments appropriate for creating a 2D circle and implement the `getArea()` method to return the area of the circle.
5. [10] Implement a new class named `Sphere` that extends `Circle`. Provide a class constructor which takes arguments appropriate for creating a 3D sphere and implement the `getArea()` method to return the surface area of the sphere.
6. [30] Implement a new class named `Main` with a public static `main()` method and private static `nextDouble()` and `calculateSum()` methods:
 - (10) The `nextDouble ()` method should simply return a double on the range (0.0, 1.0] (i.e., 0.0 exclusive to 1.0 inclusive) so that there are no 0.0 areas. Hint: see `java.util.Random.nextDouble` and `java.lang.Double.MIN_VALUE`.
 - (10) The `calculateSum ()` method should take an `ArrayList` of type `Measurable` as an argument and return the sum of all areas in the list.
 - (10) The `main()` should create an `ArrayList` of type `Measurable` and populate that list with 1000 random instances of your `Measurable` classes from parts 2-5 (i.e., 25% chance of that each instance is one of the four classes). Use the `nextDouble()` method described above to generate random dimensions to pass into the `Measurable` constructors. Track the number of instances of each class created and print the results. Finally call the `calculateSum()` method and print the result.

7. [20] Write code that is clear and efficient. Specifically, your code should be indented with respect to code blocks, avoid unnecessarily repetitive code, avoid code that is commented out or otherwise unused, use descriptive and consistent class/method/variable names, etc.

Your output should look something like:

```
rects: 244 boxes: 233 circles: 256 spheres: 267
```

```
sum: 1801.4553056918676
```

Please zip your Java source file(s) into a zipped file, rename that file as <Your Full Name>Assignment3.zip, e.g., SteveJobsAssignment3.zip, and then upload that file to Canvas. You are free to use whatever IDEs you prefer to complete your assignment; but no other files, except .java files, should be included in your zipped file. No “package” keyword should exist in your codes.