**CS 2163 Java ---**Bonus Homework Circle class: create OOP class from scratch

**General description for bonus homework “Circle class”: this is an Eclipse project, NOT a Greenfoot project.**

To finish this bonus homework, follow the steps below:

1, create an Eclipse project named ***JohnDoeBonusCircle***, where JohnDoe should be replaced by your name.

2, unzip file “bonusHomework-CircleClass.zip”, and then locate these two java source code files:

* ***Circle.java***
* ***TestCircle.java***

There are other java files, but in this step, you need to locate ***Circle.java*** and ***TestCircle.java*** in the unzip folder.

3, add these two files ***Circle.java*** and ***TestCircle.java*** into Eclipse project ***JohnDoeBonusCircle*** that you created in step 1 above. You need to follow the instructions in file “CompileAndRunTheFirstJavaProgram.docx”, page 10~12, with section title “**How to add an existing java source code file into an Eclipse project**”.

Attention: the instructions in the section teaches you how to use “drag-and-drop” to add these two java files into the Eclipse project, and you should not type in the original source code for these two files.

4, once these two files are added into the Eclipse project after you drag-and-drop them into Eclipse’s “src” folder, you will see that the files do not compile. This is normal, because file ***Circle.java*** is not completed yet, and your job in this bonus homework is to follow the instructions in file ***Circle.java*** and fulfill all requirements. Each requirement is written within each comment block, such as :

/\*\* each requirement is in a comment block \*/

. These comment blocks are inside the definition of class **Circle**.

Notice that in file ***Circle.java***, the program header from line 1to 11 is also using a comment block, but there is nothing you need to implement for the program header. All requirements for you to implement are placed in **line number greater than 14**, inside the body of class Circle.

For file ***TestCircle,java***, it has been implemented already, and you cannot modify this class.

**Do not modify “TestestCircle.java”**. **Only modify “Circle.java”**. If you have to modify “TestestCircle.java” to make the project runnable, then you are not on the right track.

5, when you finish implementing all requirements in file “Circle.java”, then you need to debug your program and clear all compilation errors, so that the program is runnable. Then run this program and make sure the program generates correct output. If the program is compilable and runnable but the output is still incorrect, then it means that there is still logic errors in your code. You need to go back and fix the logic error.

Run your program multiple times, and input different values to verify the correctness of your code in ***Circle.java***.

When you finish debugging, and the program runs correctly, it is time to submit the project.

**What and where to submit:** you need to ***zip*** then entire ***JohnDoeBonusCircle*** Eclipse project folder, and generate zip file ***JohnDoeBonusCircle.zip***, where Johndoe should be replaced by your name.

Submit this zip file to Moodle “bonus homework Circle class drop box” in Moodle folder “chap 4”.

After submission, be sure to follow the **VERIFICATION** process:

download your .zip file to a local folder in your computer, unzip the zip file and then import this Eclipse project folder, then run the project in Eclipse. If the verification process works correctly, then you have submitted the zip file successfully in Moodle; if not, you need to delete the error submission, and then fixed the problem and resubmit. **If the submitted java files program cannot be downloaded, or does not compile or does not run, your score will be** zero.

**Question**: what sample java files to look at when you work on **Circle.java**?

**Answer**: you may want to refer to **Rectangle.java** when working on **Circle.java**.

Even though you don’t need to modify file TestCircle.java, I still provide its counterpart: TestRectangle.java.

Click the Moodle homework drop box to see the due day of each homework assignment.

This bonus assignment has 10 **points. Grading components is listed in the table below:**

|  |  |
| --- | --- |
| **item** | **points for correct implementation** |
| private data | **1** |
| default constructor | **1** |
| 2nd constructor with parameter | **2** |
| getter | **1** |
| setter | **1** |
| method calculateArea | **1** |
| method calculatePerimeter | **1** |
| method calculateTotalCost | **2** |
| **total points** | **10** |

Other grading components include, and not limited to: code alignment and indentation, variable/method/class naming conventions, programmer header, suitable comments, submitted file format, overall program logic.

For Eclipse java file, if there is still any red circle check mark in java source code, you will receive zero point, because red circle check mark in Eclipse IDE means the java source code still has compilation error. You have to fix this compilation error first, before you proceed to the next phase of running java program. Even in some rare scenarios, you can run the program in Eclipse with compilation error, but **a compilation error in java source code will result in zero point for the homework.** **A runtime exception will also result in zero point.**

When coding in Eclipse and Greenfoot, please read document “RulesForIndentAndAlignCode.docx” in Moodle folder “chap 1”, and follow all the rules in code alignment and indentation