CS136: Computer Science II – Fall 2019

Homework	Points	Announced	Due
#9	30[1]	Nov-4	Nov-8

Introduction

The purpose of this homework is to practice class composition.

General Guidelines

Read the following guidelines carefully before working on this assignment.

1. This is an *individual* homework assignment. You may discuss ideas, ask questions or explain things to your colleagues. Nevertheless, you should solve the problem(s) independently.
2. You should submit your *own work*. Material brought from elsewhere (e.g. the Internet[2], a classmate, submission at a previous offering…) is not acceptable.
3. A program with syntax errors (aka compilation errors) will receive *zero* points.

Submission Instructions

1. Submissions via email will not be accepted. The homework should be submitted via BBLearn by the due date.
2. For question 1, submit Java files (i.e. with .java extension) with the names specified in the problem description. Other file types (e.g. .class, .zip, .jar, .doc, .pdf…) are not acceptable and will receive *zero* points.
3. For question 2, submit one PDF file named q2.pdf. Most word processors can generate PDF files. Other file types (e.g. .zip, .rtf, .doc, .docx…) are not acceptable and *will receive zero points*.
4. Make sure that your code compiles and runs without errors when the supplied compilation and execution commands are used.
5. When you use an IDE (e.g., NetBeans, Eclipse…) for writing Java programs, the IDE will automatically use packages and add package statements to your code files. Java files with the package statement will compile but will not run when the below commands are used. So, make sure to remove the package statements from the code you are submitting.
6. Your code must have Javadoc-style comments for all classes, methods, and fields that you write.
7. Make sure that your code compiles without errors when the following command is used:

[1] The homework will be graded out of 30 points, but it is worth 3% (i.e. 3 points) of the overall course score.

[2] Unless explicitly asked to do so.

```
javac *.java
```

8. Make sure that your code runs without errors when the following command is used:

```
java TriApp
```

Penalties	
Item	**Points Deducted**
The program doesn’t compile using the supplied command(s)	All
The program doesn’t run using the supplied command(s)	All
Improper file format	All

[26 points] Question #1

1. Create the class `Line`. A line is defined by two points: a start point and an end point. Use the Java’s built-in `Point` class (`java.awt.Point`) to represent a line’s points. The `Line` class should have three constructors: a no-arg constructor, a constructor that receives two `Point` objects and a copy constructor. Implement the method `getLength` that returns the distance between the two points using the Euclidean distance equation.
2. Create the class `Triangle` that extends the class `Shape`. Since `Shape` is an abstract class, you will need to override its abstract methods when implementing `Triangle`. The `Triangle` class should have three constructors: a no-arg constructor, a constructor that receives three `Line` objects and a copy constructor. Implement the method `getArea` using Heron’s formula.
3. Create the class TriApp that has a main method. Create Triangle 1 with these coordinates: (-7,5), (-4,4), and (-7,0) and Triangle 2 with coordinates (0,3), (6, -3), and (-2,-5). Use the getArea method to determine which triangle has a larger surface area.

Grading Rubric	
Item	**Points**
Class Line fields	2.5
Class Line constructors	2.5
Class Line getLength	2.5
Class Line getters and setters	2.5
Class Triangle fields	2
Class Triangle constructors	2
Class Triangle getArea	2
Class Triangle getPerimeter	2
Class Triangle getters and setters	2
Class TriApp	6
Missing JavaDoc comment (per occurrence)	-1

[4 points] Question #2

The class `Triangle` design induces data redundancy (i.e. unnecessary data duplication). How would you change the class `Triangle` design to eliminate unnecessary data storage?

With best wishes
Dr. Mohamed Elwakil