

AP Computer Science Java

Mr. Clausen

Program 7A

Program 7A LastNameFirstNameP7A (Triangles: 50 points)

This program will use terminal input and output, no GUI for this program. This program will also require two java files. You will create a class to work with the area of Triangles, and a client program to test your class file. We will be able to calculate the area of a triangle using three different formulas in the server file, Triangles.

To see a model for this type of program, double click on My Computer, then these folders, APCSFileJava, 3rd Edition Textbook Source Codes, Chapter05, Case Study, and open the files: Student.java and StudentApp.java.

DO NOT USE a break statement to escape from any type of loop in this class regardless of what the book says or models.

- 1) Begin with the class file (server file) Triangles.
- 2) Type comments at the beginning of the program to display your name and other information just like those used for program 2A. **Don't forget to change the information from program 2A to include the new information for this program.**
- 3) There are no packages to import for this file.
- 4) Declare the class name in the format:
`public class LastNameFirstNameTriangles.`
Don't forget that the filename needs to be the same when you save your program. This would be a good time to save your program if you haven't done so already, **LastNameFirstNameTriangles.java**.
- 5) The class file does not have a main method
- 6) There are no constants to declare for the class file.
- 7) You do not instantiate an object of the Scanner class or of any other classes in the Triangles class.
- 8) Declare all of the private variables necessary for this class. You will need variables of type **double** for base, height, side1, side2, side3, area, and angleDegrees. Don't forget to declare these variables as **private**. You may need one or more other variables, but try to make them local to each method as necessary. For example, you will need a string in the toString method. Make sure that you use descriptive variable names to create readable, self-documenting code.

9) Type the following comment:

```
//-----Constructors-----
```

Don't forget that constructors are **public** methods, do NOT have a return type, and **MUST have the same name as the name of the class**. You will need only one constructor, a default constructor that initializes **all** of the private class variables to 0.

10) Type the following comment:

```
//-----Mutators-----
```

Don't forget that Mutator methods usually are **void** methods and therefore do NOT use a return statement. Most will need at least one parameter in order to change the values of the private variables. In this section, you will create the mutator methods setBase, setHeight, setSide1, setSide2, setSide3, setAngleDegrees, setAreaBaseHeight, setAreaHero, and setAreaSines. Here are the formulas for calculating the area of a triangle:

$$Area = \frac{1}{2}(base)(height)$$

Hero's Formula:

$$Area = \sqrt{s(s - side1)(s - side2)(s - side3)}$$

where s =

$$s = \frac{(side1 + side2 + side3)}{2}$$

Law Of Sines, Area Formula:

$$Area = \frac{1}{2}(side1)(side2)(\sin \theta)$$

where θ is the included angle between the 2 sides

Allow the user to enter the measure of the angle in degrees for the Law of Sines.

Have the Triangle Class method that uses the Law of Sines convert the measure of the angle into radian measure, using the toRadians method included in the Math Class Methods (Page B-6 in the appendix).

11) Type the following comment:

```
//-----Accessors-----
```

Remember that Accessor methods usually do not have parameters, and only "access" the private variables. Therefore, they **do need a return type** and must include at least one return statement in order to return the value of the private variables to the client program.

In this section, you will create the accessor methods, `getBase`, `getHeight`, `getSide1`, `getSide2`, `getSide3`, `getAngleDegrees`, `getArea`, which should return the values of the private class variables and the results of the calculations. Also include the accessor method `toString`, which will return a string representing the area. Be sure to format the numbers right justified, and display 4 decimal places in your answer (use `String.format` on page 256 in our textbook).

- 12) If you haven't done so already, compile this file, and debug all errors so that your class file is ready for the client program.

Use blank lines to separate each of the program sections listed in all of the steps above. Now it's time to create the client program, **LastNameFirstNameP7A.java**.

- 13) Type comments at the beginning of the program to display your name and other information just like those used for program 2A. **Don't forget to change the information from program 2A to include the new information for this program.**

- 14) Import the package:
`import java.util.Scanner;`

You do not need to import your Triangles class file, but both of the .java files and their corresponding .class files need to be in the same folder.

- 15) Declare the class name in the format **LastNameFirstNameP7A**. Don't forget that the filename needs to be the same when you save your program. This would be a good time to save your program if you haven't done so already, **LastNameFirstNameP7A.java**.

- 16) Declare the main method:
`public static void main(String [] args)`

- 17) There are no constants to declare for this program.

- 18) Instantiate an object of the Scanner class in order to use input from the keyboard. In addition, instantiate one object of the Triangles class, call the object `triangle`.

- 19) Declare all of the variables necessary for this program. You will need variables of type **double** for `base`, `height`, `side1`, `side2`, `side3`, `area`, and `angleDegrees`. Use different variations of these names from the private variables in the Triangles Class file, but they can be the same name as the parameters in the Triangles Class file if you used a parameterized constructor. You will also need a few string variables for the menu, and to get the user choice for the menu. The menu choice will determine which method will be used for calculating the area and will repeat the program over and over again until the user enters choice "Q" or "q" to quit (see step #21). Make sure that you use descriptive variable names to create readable, self-documenting code.

20) Type the following comment:

```
//-----Display My Information-----
```

Follow this comment with println statements to display your name and period output just like those used for program 2A. **Don't forget to change the information from program 2A to include the new information for this program.**

21) This client should be a menu driven program as described in Section 7.3, with the following menu choices:

- 1) Calculate the Area of Triangle using the base and height
- 2) Calculate the Area of Triangle using the lengths of all three sides
- 3) Calculate the Area of Triangle using two sides and the measure of the angle between them
- Q) Quit

Of course, the program will need to be surrounded by a do...while loop to see if the user wants to repeat the program like we did in program 6A. **The program should continue repeating until the user enters a "Q" or a "q".**

22) Instead of sections for input, calculations, and output, we will need a section for each of the menu choices listed above. You will need extended if statements to cover each of the menu choices #1 – 3 listed above; you don't need a choice for "Q". Menu choice "Q" is taken care of in the "do...while" loop, which causes the program to quit. Begin each section with a line of comments according to the menu choice.

23) Each section should ask for the appropriate input, error trap the user's input, and echo back the number to put our accessor methods to work. It should then perform the calculations and display the answer. Remember, the client program should call the appropriate methods in the Triangles Class to set and get the triangles information, calculate the Area, and display the results. Allow the user to enter the measure of the angle in degrees for the Law of Sines. Have the Triangle Class method that uses the Law of Sines convert the measure of the angle into radian measure, using the toRadians method included in the Math Class Methods (Page B-6 in the appendix).

Use blank lines to separate each of the program sections listed in all of the steps above. When you are finished with your program, have tested it thoroughly to make sure that your calculations are correct, and are sure that you don't need to make any changes, then save your program in the "T" network mapping, in the Program 7A folder.

After you are finished, you will need to turn in (copy and paste) the two data files:

- 1) **LastNameFirstNameTriangles.java**
- 2) **LastNameFirstNameP7A.java**