Lab 12 Class and Objects

Please test the correctness of your program in **Q1, Q2** and **Q3** on **PASS**.

Q1. [For Practice]

Design and implement a class to compute the area of a triangle using the values of their sides. The design of class (named `Triangle`) should include:

1) **Three private members** `side1`, `side2`, and `side3`, which present sides of a triangle.
2) **One private member** `area` to preserve the computed area of triangle.
3) **A default constructor** (i.e., `Triangle()`) to initialize sides of triangle and area of triangle as zero.
4) **A parameter constructor** (i.e., `Triangle(double, double, double)`) to initialize the values of sides for corresponding triangle and initialize area of triangle as zero.
5) **A set method** (i.e. `setSides(double, double, double)`) to enter the new values for the sides.
6) **A method** (i.e. `computeArea()`) to compute the area of triangle.
7) **A get method** (i.e. `getArea()`) to display the area.

Hint-1. Area of Triangle can be computed through the following Heron's Formula:
Area = sqrt(s(s-a)*(s-b)*(s-c)), where s=(a+b+c)/2.*

Hint-2. You need to include <cmath> library to use the ***sqrt()*** *function and include <iomanip> library to use* ***setprecision*** *and* ***fixed manipulators*** *to change the precision value and printing format.*

According to the requirements above, write a program to:

1) Create first Triangle object (i.e., ***`triangle1`***) with default constructor and display its area.
2) Create second Triangle object (i.e., ***`triangle2`***) with parameter constructor and display its area. The values of sides are 3, 4, and 5.
3) Call the set method using first Triangle object (i.e., ***`triangle1`***) and display its area. Assume that all inputs are valid, i.e., new sides can form a triangle.

Expected Outputs:

Example 1.	Example 2.
Area of triangle1: 0.00 Area of triangle2: 6.00 Enter new sides for triangle1: 4 2 3 Area of triangle1: 2.90	Area of triangle1: 0.00 Area of triangle2: 6.00 Enter new sides for triangle1: 3 3 3 Area of triangle1: 3.90

Q2. [will be marked]

You must click submit to submit your solution on PASS. Deadline is 11:59pm, April 19, 2023.

Modify the program in **Q1** such that the program defines an array of triangles `Triangle arr[10]` and sort them based on their areas.

The program should let the user input the sides of ***n*** triangles (where ***n*** is an integer input by the user and $1 \le n \le 10$) and store them in the array. The program should sort and print all triangles in ascending order based on their areas (if there are two or more triangles with the same area, print them in the original order that they are defined).

Assume that all inputs are valid, i.e., *n* is within the range specified, and those three sides can form a triangle.

Hint-1. You may need to firstly design a function largertriangle(), which accepts two Triangle objects (i.e., traingle1 and triangle2) as input. The function should print whether the first Triangle (i.e., triangle1) has a larger area than the second one (i.e., triangle2) or not.

Hint-2. You may need to check how to use the bubble sort we have introduced before.

Expected Outputs:

Example 1.

```
Enter a number between 1 and 10:
4
Enter the sides of triangle 1:
3 3 3
Enter the sides of triangle 2:
3 4 5
Enter the sides of triangle 3:
3 4 5
Enter the sides of triangle 4:
3 2 3
Area of triangle 4: 2.83
Area of triangle 1: 3.90
Area of triangle 2: 6.00
Area of triangle 3: 6.00
```

Example 2.

```
Enter a number between 1 and 10:
5
Enter the sides of triangle 1:
6 6 6
Enter the sides of triangle 2:
4 5 6
Enter the sides of triangle 3:
7 8 9
Enter the sides of triangle 4:
3 4 5
Enter the sides of triangle 5:
5 7 5
Area of triangle 4: 6.00
Area of triangle 2: 9.92
Area of triangle 5: 12.50
Area of triangle 1: 15.59
Area of triangle 3: 26.83
```

Q3. [Will be marked]

You must click submit to submit your solution on PASS. Deadline is 11:59pm, April 19, 2023.

Design and implement a class, which helps calculates the distance from the origin of a Euclidean space, for a given point. The design of class (named **Point3D**) should include. Check and complete the skeleton code in **Point3D.cpp**.

In particular, complete the following:

1) Member fields, constructors, mutators, and accessors have already been provided. Can you replace the two constructors in the skeleton code with a default constructor realized with initializer list?
2) Complete the function declaration for the **friend** function, which overloads the operator << such that we can use cout to print out the object. The function has two parameters, namely a variable of type **std::ostream&**, and a variable of type **Point3D&**.
3) Complete the function definition for **calDistance()** as well as for the above friend function overloading the operator << such that the following outputs are printed out calling this function when the user inputs a **3D** point.

*Hint: You need to include <cmath> library to use the **sqrt()** function and include <iomanip> library to use **setprecision** and **fixed manipulators** to change the precision value and printing format.*

Expected Outputs:

Example 1.

```
Enter a 3D point's x, y, and z coordinates in order:
4 2 3
This point is: (4.00, 2.00, 3.00)
The distance from (0, 0, 0) is 5.39
```

Example 2.

```
Enter a 3D point's x, y, and z coordinates in order:
3.1 4.2 5.7
This point is: (3.10, 4.20, 5.70)
The distance from (0, 0, 0) is 7.73
```