

**Homework 2:** Void/Value-returning functions**Due Date:** 10/16/24

You have to write a program that computes the area of a triangle. Input consists of the three points that represent the vertices of the triangle. Points are represented as Cartesian units i.e. X,Y coordinates as in (3,5). Output must be the three points, the three distances between vertices, and the area of the triangle formed by these points. The program must read the coordinates of each point, compute the distances between each pair of them and print these values. Next, it must calculate the area and print its value.

All input must be read from input file **inpuhw2.txt**. All output must be sent to the output file **outpuhw2.txt**.

**Solution requirements:**

1) Your program must solve the problem using the following functions:

**getPoint()**: Receives the input file. It reads from the file the coordinates (x, y) corresponding to a **single** vertex of the triangle and returns both of them rounded to a single decimal place to the caller.

**calcLength()**: Receives the coordinates of two points and returns the distance between them (calculated using the equation provided below) rounded to ten decimal places.

$$\text{Length between points } (x1,y1) \text{ and } (x2,y2) = \sqrt{(x1-x2)^2 + (y1-y2)^2}$$

That is, the length is the square root of the square of the distance between the **x coordinates** plus the square of the distance between the **y coordinates**.

**semiPerimeter()**: Receives three lengths and returns the value of the semi perimeter (calculated using the equation provided below) rounded to ten decimal places.

$$\text{Semi perimeter} = \frac{1}{2} * (ab + bc + ca)$$

Where **ab**, **bc** and **ca** are the lengths between points **a** and **b**, **b** and **c**, and **c** and **a** respectively.

**calcArea()**: Receives three lengths and returns the area of the triangle (calculated using the equation provided below) rounded to two decimal places. **This function must call semiPerimeter()** to calculate the value for **s** before calculating the area of the triangle.

$$\text{Area} = \sqrt{s*(s-ab)*(s-bc)*(s-ca)}$$

Where **s** is the semi perimeter and **ab**, **bc**, and **ca** are the three distances between the three points.

**printDistance()**: Receives the output file, the coordinates of two points, and the distance between them and prints a message similar to the following one:

The distance between (1.0,1.2) and (6.0,6.0) is 6.93

**square()**: Receives a value and returns its square rounded to two decimal places. You must use it to calculate the squares of the distances of the x and y coordinates in calcLength(). **Do NOT use pow() to implement this function.**

**round\_off(value, places)**: Receives a value (double precision real number) and a number indicating a quantity of decimal places (whole number) and returns the value rounded to the specified number of places.

Your **main()** function must:

- Open (check they are opened) and close the input and the output files

- Call `getPoint()`, `calcLength()`, and `printDistance()` as many times as needed
- Call `calcArea()`
- Format the output so real numbers are printed in fixed format
- Print the area of the triangle to the output file

**Note:**

The **coordinates of vertices** must be printed with a **single decimal** digit while the **distances** and the **area** must be printed with **2 decimal** digits.

Unless specified otherwise all the values are double precision real numbers.

You can declare all the variables that you need.

All the functions must be declared below `main()`.

2) You **can** use **ONLY** the material learned and used in the **first 10** lab assignments. The only if statements that you can use are those that check whether the files were opened.

3) Your program must pass all my tests.

4) You must choose the most appropriate type of function and type of parameters for each of the functions described above.

The program must compile without errors or warnings.

Open `hw2.cpp` in your IDE and implement your solution.

Enter your algorithm (as comments) and implement it in C++.

**Note:**

- Do NOT remove or modify the statements that I use to test certain things in your program.
- Pay attention to the input and the output formats. Your solution must behave exactly like mine.
- Carefully analyze the sample run shown below and use it as a reference to ensure you do the right things.
- Try the values in the provided input file and check if you get the right result (compare with my solution). If you get an error message on the output, read the comment on the line specified in the message to find out what is wrong. Next, test your program using different values in the input file.

Sample run of the program

Given the following content of the input file

1.03	1.19
6.0	5.96
3.04	6.5

The content of the output file should be

The distance between (1.0,1.2) and (6.0,6.0) is 6.93
The distance between (6.0,6.0) and (3.0,6.5) is 3.04
The distance between (1.0,1.2) and (3.0,6.5) is 5.66
 The area of this triangle is: 8.45

Review the examples discussed in class, the lab assignments done so far, and the textbook to get an idea of what you need to do. The **algorithm** must be written in **pseudocode** and should look like my lab handouts. **Include your algorithm in the source code as comments.**

Do not hesitate to use the corresponding topic in Discussions to post your questions/doubts about this assignment. I will reply as soon as I can.

### IMPORTANT:

**You must submit ONLY ONE solution per team.**

Your program must be well commented, use meaningful identifiers, use named constants, and use indentation as shown in the textbook.

Your program must have the following comments at the top:

```
//*****
// Team #                CSCI 1470                Fall 2024                Homework # 2
// First and Last Name
// First and Last Name
// Using your own words, write here a description of what the program does.
//
//*****
```

**Include the names of both teammates only if both participated in the solution, otherwise just enter your name.**

When done, submit your solution through Blackboard using the "Assignments" tool. Do Not email it.

Paste the link to your final solution along with your source code in the textbox opened when you click on [Create Submission](#) before you click on [Submit](#).

### Grading criteria

You start with 100 points and then lose points as you don't do something that is required.

#### **-30: Used material not corresponding to what was used in the first 10 lab assignments**

-10: Too few/no comment(no description of what the program does).

-10: Data type is not correct

-10: Mixed data types in expressions

-10: Incorrect output format.

-10: Didn't check if files were opened

-20: Didn't round off

-10: Incorrect rounding

**-10: Did not pass test. (each)**

**-30: Missing/incorrect algorithm**

-15: Poor quality algorithm

-25: Didn't implement function (each)

-10: Incorrect type (void/value-returning) of function (each)

-10: Incorrect type of parameters (value or reference)

-10: Incorrect number of parameters(more than those required)

-10: incorrect implementation of the function's body

-10: Incorrect function call (each)

-50: Incomplete program.

-50: Program does not compile

**-20: Incorrect/missing source code or incorrect/missing link to solution**

**-100: The code submitted is not your creation (you got it from a web site or another person)**

-100: No submission.

**-100: No team contribution**

-10: Late submission

**Important:** more points may be lost for other reasons not specified here.