Assignment 9

In the lecture notes on ***Object-Oriented Programming with C++ (Part 1)*** an example is given of a Point class, and a Line class which uses this Point class.

In this Assignment you are to write a class to represent a Triangle using these Point and Line classes; and also a derived class of Triangle to represent an Equilateral Triangle.

In your Triangle class, you need **3 private data fields** to hold the 3 points which are the vertices of a triangle.

-- You should provide **only 1 constructor**, which takes the x and y coordinates of two points in the plane as arguments: each of these points is a vertex of the Triangle object. This constructor sets the third vertex of the Triangle object to the point (0.0,0.0).

-- You should provide **only 2 public methods** in your Triangle class: one which calculates the ***area*** of a Triangle, and one which calculates the ***perimeter*** of a Triangle. Both methods return a value of type double, and can use objects of the Point and/or Line classes if required. *Note: the area of a triangle can in general be found using Heron's formula.*

-- You should provide a class-wide variable called **triCount** to keep track of the number of Triangle objects that have been instantiated.

-- You should declare a friend function called **print_Tri_details** which prints out the coordinates of the vertices of a Triangle object.

An equilateral triangle is a special kind of Triangle in which all 3 sides have the same length. In your Equilateral class, you need **1 private data field** to hold the length of the side of an equilateral triangle.

-- You should provide **only 1 constructor**, which takes one argument: the x-coordinate of a point in the plane; ***you may assume that the y coordinate in the declaration of an Equilateral object is 0.0***. Since all Triangle objects have one vertex at (0.0,0.0), and you know another vertex of the Equilateral being constructed is at (x,0.0), you can compute the coordinates of the third vertex of that equilateral triangle, and use the computed coordinates in the definition of your Equilateral constructor. *Hint: you may use 1.732 as an approximation for sqrt(3)*

-- In your Equilateral class, you **must override** the 2 methods inherited from your Triangle class to calculate the area and perimeter of a triangle, instead using the length of the side of the equilateral triangle to calculate the area and perimeter of that triangle; ***you may NOT use these inherited methods as-is***.

-- You should provide a class-wide variable called **equiCount** to keep track of the number of Equilateral objects that have been instantiated.

-- You should declare a friend function called **print_Equi_details** which prints out the value of the length of the side of an Equilateral object.

Note: you must use the Point and Line classes (defined in the lecture notes) as-is, ***without any changes***. Their header and method implementation files are provided for you in CS Moodle: you must not upload them with your submission. **If your submission does not compile and run with the Point and Line files posted in CS Moodle, you will receive 0 for your approach to this Assignment.**

Write another C++ program which tests your Triangle and Equilateral classes, as follows: first, write the code for the friend functions **print_Tri_details** and **print_Equi_details** as specified above. Then in the main function:

-- Declare a Triangle object for which the x and y coordinates of two points are entered by the user from the keyboard. ***You may assume the user always enters valid values (of type double) for all coordinates.***

-- Declare another Triangle object for which the x and y coordinates of two points are read from the file **comp20080-autumn2021-Asst9-data.txt** which is also posted in CS Moodle. The x and y coordinates of the first point are on the 1st and 2nd lines of this textfile; the x and y coordinates of the second point are on the 3rd and 4th lines of this textfile.

-- For each Triangle object, print out the values of its private data fields using the appropriate friend function.

-- For each Triangle object, if the area is calculated to be 0 (which happens if all 3 vertices are collinear), your program should output a message to the screen saying that the triangle is "trivial". Otherwise, your program should output a message to the screen with the area and perimeter of the Triangle object.

-- Then the user enters a single value which your program should use as the x-coordinate of a vertex of an Equilateral object. ***You may assume the user always enters a non-zero value of type double.***

-- Print out the value of this Equilateral object's private data field using the appropriate friend function.

-- Output a message to the screen with the area and the perimeter of this Equilateral object.

-- Finally, print out to the screen how many Triangle and Equilateral objects were instantiated by your program.