CS 2 - Lab 2 Possible Points: 110

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**All About Triangles**

Write a C++ program using object-oriented design that reads an input file with information about the sides of some triangles. The program will then find various properties of each triangle and create an output report as a file with the following requirements.

1. Define a class called Triangle in Triangle.h header file. Class definition should contain only the private/protected member variables and public function prototypes to handle each triangle’s properties. Specifically:
   1. Define private data member(s) to store triangle’s 3 sides’ length
   2. Define public constructors and functions to set and get the values of private member(s)
   3. Define public member function to calculate and return perimeter of triangle
   4. Define public member function to calculate and return area of triangle
   5. Define public member function to find and return type of triangle based on sides
2. Implement Triangle class details – mostly the functions – in a separate file called Triangle.cpp.
3. Create a main program file called lab2.cpp that includes main function and tests Triangle class.
4. Your program will prompt user to enter the input file name and read the length of 3 sides of each triangle from the file. Each line in input file represents 3 sides of a triangle separated by a space.   
   Your program should read arbitrary number of triangle's data but no more than 100.
5. Program will implement and use function to sort the triangles data in ascending order based on area.
6. Your program defines and uses function(s) to generate a report file with proper formatting using *iomanip* library rounding decimal numbers to 2 decimal points, separating headers from the main contents, etc. as shown in the provided sample output file.
7. Program must prompt user for the output file name to save the report to.
8. **Bonus (10 Points):** Your program will define and use two functions to find and print the largest and smallest side of each triangle in the output report.   
   Hint: You can update Triangle.h and Triangle.cpp as necessary.

**Lab instructions and other requirements:**

1. Create cpp files as stated above and use the provided partial code.
2. Fix all the FIXMEs.
3. Create triangles.txt resource file and copy the given data items to test your program.
4. Follow the best programming practices.
   1. Must write program and programmer information at the top.
   2. Write adequate comments.
   3. Use whitespaces and indentations to format source code and output data.
   4. Write algorithm steps as comments when necessary.
   5. Use proper self-describing identifiers – variable, function names, etc.
   6. Do NOT use global variables (global constants/preprocessor directives are ok) – pass data to functions using parameters.
5. No late submission will be accepted unless prior approval.
6. **Upload the source files (\*.cpp, \*.h) into D2L Lab2 dropbox before due date.**