Lab #6
C++ String and I/O

Purpose:

The purpose of this lab is to write some code in C++ that uses aspects of the language that has been covered in class thus far.

Source code files for the lab are on Canvas. You will save all your source files for the lab to the GitHub account you created in Lab 0.

Preparation:

Review class presentations and chapters from the text according to the syllabus.

Procedure:

1) Start up CodeLite and open the workspace ECEGR2020 you created for Lab 0

2) In the workspace, create a new project: File -> New -> New Project. Name the project Lab_4 and make the Type a "Simple executable (g++)"

3) For each function below, program, run and debug each. In your lab report, be sure to place the output of each function to show that your program worked.

Program the Following:

You'll be updating the Student class from Lab #5 and the program from Lab #4.

A) Use your Student class from Lab #5 and update the Lab #4 program to use it. Update the file I/O operations to use ifstream and ofstream objects

    1) Now that the Student class has names with strings of various length instead of fixed lengths like in Lab #4, what are some of the things you think you need to change with your file structure in order to handle those strings? Implement your approach and document it with comments in your program

    2) A big part of software development is dealing with the content of older programs. Assuming that your new Student program needs to read files created by the program in Lab #4 and this lab, what do you need to do to ensure compatibility?

    3) Code up the ability to read files made from Lab #4 in your Lab #6 program. The general rule in software development is that version N of a program needs to be backwards compatible with versions 1 to N-1, but older versions do not need to be

forward compatible (would be nice, but very difficult to do given new features are added). Knowing this compatibility rule, what are some things you could have done in the Lab #4 version of the program and/or data file structure that would help? Write down your ideas in your lab report.

B) A late breaking change from upper management on the Student program implementation. Implement a new version of the Student class that uses the string class. Same functionality needed in that it needs to read files from Lab #4, but implement your scheme in the file to support compatibility going forward from this version of the application