ENSC 251 Lab Assignment 2

In the ENSC 251 course, you will work on a multi-lab project: you will design and implement a simplified graduate student admission system using the skills learned throughout this course (Object-Oriented Programming with C++). This project will be developed incrementally throughout a total of four labs, each weighing 10% of your final grade marks. All lab assignments will be carried out and evaluated in pairs (2 students per group) - I hope by now all of you have already found a partner. Some general grading logistics have been posted on our course website: https://coursys.sfu.ca/2019fa-ensc-251-d1/pages/labs. Please note that the detailed grading scheme for each lab will be released after your lab grading though: think about you are in a real interview, nobody will tell you what the detailed grading schemes are.

More description of the simplified graduate admission system was given in lab assignment 1. In assignment 1, you have built the four classes that you will use throughout the four lab assignments. Now we are moving to lab assignment 2, which will be based on lab assignment 1. For those student groups who didn't do well in assignment 1, the TAs will post the code of assignment 1 from those top student groups who got bonus points, and you are free to use their assignment 1 code as basis for your assignment 2.

Lab Assignment 2:

Assume there are a number of domestic and international student applicants who are applying to SFU graduate school, and their profile is stored in domestic-stu.txt and international-stu.txt, which you are already able to read in assignment 1. In lab assignment 2, assume we have a fixed number of domestic and international applicants which we know beforehand, and both numbers don't exceed 100 (i.e., no more than 100 domestic student applicants, and no more than 100 international student applicants). The updated domestic-stu.txt and international-stu.txt files are available on the course website for download. You will have to achieve the following goals:

1. Use arrays or vectors to store all the DomesticStudent and InternationalStudent objects, which are read and initialized from the input files.
2. Implement the following friend functions for Student class—**compareCGPA, compareResearchScore, compareFirstName, compareLastName**—so that you can compare the CGPA, research score, first name, last name of two DomesticStudent objects (and two InternationalStudent objects). Each of these functions should return three kinds of values: less than, equal to, or greater than. For lab assignment 2, we don't compare a DomesticStudent object against an InternationalStudent object.
3. Overload the << operator of DomesticStudent and InternationalStudent classes to print out the object information.
4. Based on the user input, sort all DomesticStudent (and all InternationalStudent) objects in the array or vector by their CGPA, research score, first name, last name using these friend functions, and print out the sorted array or vector of objects. For CGPA and research score, sort them from high score to low score; and for first name and last name, sort them using ascending order, e.g., "Alex" comes before "Mary". Note each time the sorting is based on a single field**,** e.g., based on only CGRA, or only first name. Your program should be able to take input from a user (hint: using cin). For example, if user types in 'c', you should sort all DomesticStudent (and all InternationalStudent) objects based on their CGPA, and print

out the sorted objects. A better approach is to first ask a user to select DomesticStudent or InternationalStudent, and then ask a user to select the field that they want to sort.

5. Based on the user input (if a user chooses an overall sorting), sort all DomesticStudent (and all InternationalStudent) objects in the array or vector, such that they are first sorted based on their research score. If they have the same research score, then they are further sorted based on their CGPA. If they further have the same CGPA, then they are sorted by their province (for DomesticStudent) or country (for InternationalStudent). For InternationalStudent, if one's TOEFL score doesn't meet certain conditions, we drop that student object from the sorting process. The standard is as follows: the minimum overall TOEFL score is 93 with a minimum of 20 in each category (reading, listening, speaking, and writing). In contrast to point 4 where sorting is based on a single field, this overall sorting is based on multiple fields as we described. And you may want to add some friend function or operator overloading to the DomesticStudent and InternationalStudent class. Similarly, after this overall sorting, you will print out all the sorted objects.
6. **!!IMPORTANT NOTE!!** You have to write your own code for all the above task, no library function calls (e.g., sort API) are allowed.

In addition to the actual coding implementation, you need to provide good commenting, naming, and other good coding styles (including the header file style to avoid the multiple includes problem); all these count in your lab assignment marking.

Note: For grading logistics and remote machine access, please refer to the course website. If you have any questions, please post them on Piazza.

Assignment Submission:

Your lab assignment 2 will be submitted electronically through CourSys. You will need to submit a single lab2.zip file. Failure to comply with this format could result in zero score on this assignment.

Submission Deadline:

Your lab assignment 2 is due at **11:59:59pm on Sunday, Oct 13th, 2019**. You need to meet the deadline: every 10 minutes late for submission, you lose 10% of this lab mark; that is, 100 minutes late, you will get zero for this lab.

Lab Demonstration:

You will have to demo your lab assignment 2 to your TA in the following lab sessions that you enrolled in: Tuesday (**Oct 15th, 2019**), Thursday (**Oct 17th, 2019**), and Friday (**Oct 18th, 2019**). Only code from your CourSys submission is allowed in the lab demo. Each student group has around 6-7 minutes to explain your code to the TA. If you fail to do the demo (without a medical note), or if it is determined that you do not understand the code being evaluated, you will be awarded zero on this lab assignment. Also please show up in the demo day on time (beginning time of each lab session), otherwise you will lose 0.5 mark.