CS 162 LAB #4 – File I/O

In order to get credit for the lab, you need to be checked off by the end of lab. You can earn a maximum of 3 points for lab work completed outside of lab time, but you must finish the lab before the next lab. For extenuating circumstances, contact your lab TAs and the instructor.

This lab is worth 10 points total. Here's the breakdown:

- 4 points: Design for the problem
- 6 points: Implement your design

(4 pt) Step 1: Design

In this lab, you will practice reading and writing to a file, a.k.a. File I/O. You can copy and paste or download this example file using wget command:

wget https://classes.engr.oregonstate.edu/eecs/spring2023/cs162-010/labs/input.txt

The input file provides details for a student database in the following format:

Number_of_students

ID_Number Student_First_Name Student_Last_Name Major GPA

…<Repeats n number of times>…

ID_Number Student_First_Name Student_Last_Name Major GPA

Write a program that does the following:

1. Open the file using a file object.

2. From the file, read the number of students and each student's information into your program. (Hint, you need to store all students into a dynamic array of struct student, whose size is determined by the number of students from the file)

3. Provide the following two options to the user, and ask for the input:

- Search students by last name
- Print all students that are in the honor roll (with GPA >= 3.50)

4. Based on the user input, process the data from the array, and write the result **to an output file** named `output.txt`. For each student that meets the requirements, use the following format to store/print:

```
ID:
First Name:
Last Name:
Major:
GPA:
```

Each section of information should be labeled in the output file in all capital letters. A student struct should be used to store and manipulate the file information between reading and writing the file. You must include the follow three functions with the exact prototypes:

(Feel free to add more functions when necessary)

- `student * create_student_db(int);`
  This function will create the array of student based on the number of students in the file

- `void populate_student_db_info(student *, int, ifstream &);`
  This function should do the work of actually reading through the `ifstream` that represents your opened file and storing the information in a pre-allocated array of student that's passed in
  8
- `void delete_student_db_info(student *&);`
  This function will delete all dynamic memory created in your program.
  Note: don't forget to set the original 1D array pointer to NULL inside this function.
  Question: Why do we need to pass the pointer by reference?

Your **main function needs to check to make sure the file you open exists** before moving forward. If the file doesn't exist, then you need to provide an error message and get a file name that does exist.

- (1 pt) Write a **design** for the main function in the driver file, `driver.cpp`.
- (3 pts) Write a **design** for the `create_student_db()`, `populate_student_db_info()`, **and** `delete_student_db_info()` **as well as the functions needed to satisfy the above bulleted output** functions in the implementation file, `student_db.cpp`

Here's some documentation that will help you get going with File I/O:

- C++ Basic file I/O: http://www.learncpp.com/cpp-tutorial/186-basic-file-io/
- ifstream: http://www.cplusplus.com/reference/fstream/ifstream/
- ofstream: http://www.cplusplus.com/reference/fstream/ofstream/
- fstream: http://www.cplusplus.com/reference/fstream/fstream/

(6 pts) Step 2: Implementation

(5 pts) Now, implement the `driver.cpp, student_db.cpp`, and `student_db.h` files.
(1 pt) Create a `Makefile` to manage the compilation of all these files. You can adapt the Makefile that was posted on the Calendar page in Canvas.

(2 pts) Extra Credit: Sorting

In addition to the two options listed above (searching by last name and students in the honor roll), implement a third option to **sort the students by ID number** (in ascending order), and write the result to the output file.

Hint: you may use bubble sort algorithm: https://www.geeksforgeeks.org/bubble-sort/

Show your completed work and answers to the TAs for credit. You will not get points if you do not get checked off!

Submit your work to TEACH for our records **(Note: you will not get points if you don’t get checked off with a TA!!!)**

1. Create a **zip file** that contains all files you’ve created in this lab:

```
zip lab4.zip student_db.h student_db.cpp driver.cpp Makefile
```

2. Transfer the tar file from the ENGR server to your local laptop.
3. Go to TEACH.
4. In the menu on the right side, go to **Class Tools → Submit Assignment**.
5. Select **CS162 Lab4** from the list of assignments and click “**SUBMIT NOW**”
6. Select your files and click the Submit button.