CST8234- C Language

Assignment 3

Due Date - December 1, 2019

Problem statment:

In this assignment, you will build a registration system that will read the students' and courses' information from local files. It will also give the user the ability to store new information.

This assignment is intended to get you to practise using file IO functions and user inputs.

Background Information:

This assignment is designed to help understand how to read from and write to text files in C. You are given two files, `Students.txt` and `Courses.txt`, which contain some information for existing students and courses in a school system.

Think of theses files as your database, you are going to read the information form it, and will be storing any new information to it.

The system will give the user the chance to either display the currently existing information, or add new information. All information will be coming from the local files mentioned above.

Requirements:

Write a program that achieves the following requirements:

1. The program should give the user a menu of action to choose from. These actions are
    1. Display current students.
    2. Display current courses.
    3. Add a new student
    4. Add a new course
2. When the user chooses to display current students, the program should read all the information from the students file and display it neatly.
3. When the user chooses to display current courses, the program should read all the information in the courses file and display it neatly.
4. If the user chooses to add a new student, the program should ask for the following information:
    1. The student ID
    2. The first name
    3. The last name.
5. The program should save the new student information to the students file, without deleting any of the previous information.
6. If the user chooses to add a new course, the program should ask for the following information:
    1. The course code.
    2. The course name, which could contain more than one word, i.e "C Programming Language".
7. The program should save the new course information to the course file, without deleting any of the previous information.
8. The program MUST maintain the formats for the text file, keeping each student or course information in a separate line and with the same order for each piece of information.

Design Requirements:

1. You must wright your code using **3 or more functions**. Having the code only in `main` will make you loose **2 points.**
2. **YOU MUST** build your code using all build flags mentioned in the slides as will as '`-w`' switch. Failing to do so will make you loose **1 point** on the assignment.
3. Your code must close any open files and free any allocated memory when appropriate before the end of the program.

Sample Output:

```
1- Display students
2- Display courses
3- Add a student
4- Add a course
5- Quite
Please choose an option: 1
Amelia Daniel 12345
Thomas Joe 23456
Harry Michael 34567
Sophia Charles 45678
James Reece 56789
Emma Richard 67891

1- Display students
2- Display courses
3- Add a student
4- Add a course
5- Quite
Please choose an option: 3
Please enter student ID: 23543
Please enter student first name: St1
Please enter student last name: St1
1- Display students
2- Display courses
3- Add a student
4- Add a course
5- Quite
Please choose an option: 1
Amelia Daniel 12345
Thomas Joe 23456
Harry Michael 34567
Sophia Charles 45678
James Reece 56789
Emma Richard 67891
St1 St1 23543

1- Display students
2- Display courses
3- Add a student
4- Add a course
5- Quite
Please choose an option: 5
Program ended with exit code: 0
```

Submission instructions:

1. No late submissions are accepted.
2. You **MUST** work in a group of maximum 2 students to complete this assignment. Individual work is **NOT** accepted.
3. You must submit the source code for the program you wrote.
4. Add all your files under a folder call “lastName-firstName-Assign2”, then Zip the folder and submit the zipped folder only.
5. Make sure to submit all files required to compile and run the program on the instructor machine without any errors.
6. **DON’T** submit any extra file. For example, the binary file (AKA object or output) files like .exe or .o.
7. Add a Readme.txt file that contain each student name and student number.
8. Brightspace is configured to keep the last submission only. Please make sure your last submission in the one you want to get marked.
9. All submission must be done on the main Brightspace shell, 19F_CST8234_010_ALL or 19F_CST8234_020_ALL, not the lab section one.