

CST8234 - C Programming

LAB 4

Due Date Sunday Oct 20th, 2019 at 11:30pm

LAB OBJECTIVE

By completing this lab, you will learn to:

- Use structs and pointer to structs.
- Defining structs as types.
- Reading user input into structs.

LAB INSTRUCTIONS:

STATEMENT OF THE PROBLEM:

Continuing on what we developed in assignment 1, this lab will examine a different way to save students and their course's information using C.

The system that you will developed, holds the personal information for a number of students and the information of the course each student is registered in.

The assumption is that each student is registered in a single course only. But the system will have a number of students saved.

REQUIREMENTS:

In this lab, you will introduce a simple student records system, which will save the student information as well as a single course that the student is registered in.

Firstly, the system should ask the user to enter the number of students they wish to save. Then the system should start asking the user to enter each student and course information.

Keep in mind that the student information is to be presented in a struct, called `Student`, while the course information is presented in another struct, called `Course`.

The `Student` struct will have a pointer reference to the `Course` struct variable that has the course information. Yet, the `Course` struct doesn't hold any information about the student at all.

Once the user enters all students and course information, the system must print out all information entered by the user for verification.

The Following is a complete list of the system requirements:

1. Define a `Student` struct the holds the following information about each student:
 1. First name
 2. Last Name
 3. Student ID.
 4. A pointer to the course student registered in.
2. Define a `course` struct that holds the following information about each course:
 1. Course code, which is a 7 alphanumeric string.
 2. Course name
3. `Course` struct must be a type defined unnamed struct.
4. The system should ask the user for the number of students they wish to save.
5. For each student, the system should promote the user to enter their information.
6. For the student's course, the system should prompt the user to enter the course information.
7. Once all students and their courses are entered, the system must print all their information for the user verification.

The following is a ample output of the system.

```
How many students do you wish to enter = 2
Enter student #1 first Name: john
Enter student Last Name: Simth
Enter student ID: 12345
Enter course code: CST8234
Enter course name: C

Enter student #2 first Name: John
Enter student Last Name: Appleseed
Enter student ID: 23456
Enter course code: CST8288
Enter course name: Java

Student Name: john Simth
Student ID: 49
Course Code: CST8234
Course Name: C

Student Name: John Appleseed
Student ID: 50
Course Code: CST8288
Course Name: Java
```

SUBMISSION INSTRUCTIONS

- No late submissions are accepted.
- You can work in a group of maximum 2 students to complete this lab. Individual work is also accepted.
- If working in a group, add a Readme.txt file that contain each student name and student number.
- You are required to show your work during the lab time to the instructor.
- You must submit the source code for the program and the makefile. Add all your files under a folder call “lastName-firstName-Lab4”, then Zip the file and submit the zip file only. Make sure to submit all files required to compile and run the program on the instructor machine with any errors.
- DON'T submit any extra file please. For example, the binary file (AKA object or output) files like .exe or .o.
- Brightspace is configured to keep the last submission only. Please complete all steps required to finish the lab before your first submission.
- All submission must be done on the main Brightspace shell, 19F_CST8234_010_ALL or 19F_CST8234_020_ALL, not the lab section one.