Course Title: Introduction to problem solving & programming Course Code: CS143

Semester: 2nd 2019/2020

Issue Date: 31/5/2020 Submission Date & Time: 7/6/2020 at 11:59 PM

Lecturer name: ---------------------------------------------------------------------------------------

Student Name: ---------------------------------------------------------------------------------------------

Registration #: ---------------------------------------------------------------------------------------------

Department: ------------------------------------------------------------------------------------------------

**INSTUCTIONS**

* Fill in all information required above in the file and turn it in
* You are required to submit (.c file) of the program code
* Attach a copy of your student/National ID with your submitted file.
* File naming protocol when submitting to lecturer:
  + This word file: your\_registration\_no\_your\_name CS143 Early Pass.doc. Replace “your\_registration\_no\_your\_name” in the file naming with your personal data
  + ID image file: your\_registration\_no\_your\_name ID.jpg. Replace “your\_registration\_no\_your\_name” in the file naming with your personal data.
  + Program file: your\_registration\_no\_your\_name Program.c. Replace “your\_registration\_no\_your\_name” in the file naming with your personal data.
* Attempt all requirements
* Turn-in solved Early-Pass Assignment onto Google classroom of your lecturer.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| **For lecturer use only** | | | |
| **Assignment** | **Marks** | | |
| Satisfactory | | Unsatisfactory |
| **Project** |  | |  |
| **Lecturer** | Name: | | |
| Date: | Sign: | |

**University Registration System**

It is required to write a C program to perform some functions of the registration system of an educational institute (e.g. AASTMT). The program should include a data structure for each one of the possible students available in the institution. The data structure should include at least the following:

1. Student name, which is to be defined as a structure containing the following:
   1. First name
   2. Middle name
   3. Last name
2. Student registration number, entered by the user.
3. Student nationality.
4. Student ID.
5. Student home address which is a structure containing the following:
   1. Flat number
   2. House number
   3. Street name
   4. Area/District name
   5. City name
   6. Governorate name
   7. Country
   8. Postal code
6. Student date of birth, which is a structure containing the following:
   1. Day of birth (restricted between 1 and 31)
   2. Month of birth (restricted between 1 and 12)
   3. Year of birth (restricted between 1980 and present date)
7. Student gender.
8. First enrolment term and year.
9. Student department of enrolment.
10. Current term, which is calculated according to the credit hours finished.
11. Student contact, which is a structure containing the following:
    1. Contact e-mail of the student
    2. Student home phone number
    3. Student mobile number
12. Credit hours, which is a structure containing the following:
    1. Maximum required for the degree
    2. Finished
    3. Current
    4. Remaining (The remaining hours = Maximum -Finished – Current)
13. Student GPA, which is a structure containing the following:
    1. Overall GPA attained so far by the student
    2. Number of semesters finished so far.

**For the above data structure, it is required to perform the following:**

1. Define and Implement the above data structure in C.
2. A dynamic array containing the data of all the students should be defined as a local variable inside the “main” function. This array should be sized so as to exactly contain the data of the students defined in the program according to the new type implemented above. Dynamic allocation is used to ensure no waste of space occurs in the memory.
3. Write functions to perform the required tasks from the system:
   1. A function to set the data of one student at a time from the user.
   2. A function to get the data of one student and display on the screen.
   3. A function to calculate the age of the student.
   4. A function to sort in ascending order, students of a certain department and enrolment year based on their GPA.
   5. A function to retrieve students coming from a certain family to be able to apply discounts on paid fees.