

## COEN 243 Project- Winter 2021

**Deadline:** April 21<sup>st</sup>, 11:59 PM EDT

**Type:** Group (Not Compulsory)

**Weight:** This project is worth 15% of your final

**Submission:** Must be on Moodle (**both must submit the file and include the partner's name**)

---

**Purpose:** The purpose of this project is to allow you practice Object Oriented Design, Pointers, File I/O, Exception Handling, Functions and Arrays.

**Problem Objective:** The objective of this problem is to create a fully functioning system that handles the records of students at Concordia University. Based on the following narrative, you need to come up with an Object-Oriented design to represent the different entities in the system.

### Part 1: Student class

The initial information of *student* members is maintained in a file called ***student.txt*** (**first line shows the number of lines**). Each record in this file is composed of:

- Student first name
- Student last name
- Student id
- Date of birth
- GPA to the date (Range: 0 - 4.4)
- Start year
- Completed credit
- Program (B: bachelor, M: master's, P: Ph.D.)

You need to create a class **Student** with the above attributes and following member functions:

- **Setter and Getter Functions for of all the attributes.**
- **CompleteProgram:** This member function takes the program type and the student id as input arguments and returns **true** if the student has finished all the courses by comparing the completed credits with the required credit for the program that the student is enrolled in. The function returns false if the program is not completed.
  - Bachelor: 140 credit
  - Master: 16
  - Ph.D: 12
- **StudentStatus ():** This function takes the GPA as an input argument and returns the status as below:
  - if GPA ≥ 3.5 returns A+
  - if GPA in the range of 3 (inclusive) - 3.5: A

- if GPA in the range of 2.5(inclusive) - 3: B
  - if GPA in the range of 2(inclusive) - 2.5: C
  - Less than 2: D
- **Print\_Std\_Info()** : This function accepts the id of one student and prints out all the information of that student (all the attributes).
- **Comapre\_GPA()**: This function compares the GPA of two students and returns the id of the student who has the higher GPA.
- 

## Part 2: Faculty Class:

Assume the faculty of engineering has three departments: electrical and computer engineering, mechanical engineering and civil engineering.  
You need to implement this class to read the input files and create the arrays of students for different departments of the faculty of engineering.

### Attributes:

a pointer to an array of students of electrical and computer engineering  
a pointer to an array of students of mechanical engineering  
a pointer to an array of students of civil engineering

### Constructor():

Reads **student\_elec\_comp.txt** file and creates a dynamic array of electrical and computer engineering students  
Reads **student\_mech.txt** file and creates a dynamic array for mechanical engineering students  
Reads **student\_civil.txt** file and creates a dynamic array for civil engineering students

### Member functions:

**Highest\_GPA**: This function accepts an array of students and its size, then prints out the information of the one who has the highest GPA

**N\_of\_UnderGrad**: This function accept an array of students, and its size and returns the total number of undergraduate students.

**N\_of\_Grad**: This function accept an array of students, and its size and returns the total number of graduate students.

**Avg\_UnderGrad**: This function accept an array of students and its size, and returns the average of GPA of undergraduate students.

**Avg\_Grad**: This function accept an array of students, and its size and returns the average of GPA of graduate students.

---

## Part 3: Driver:

- a) In the driver you will test your system. You first create an object of **Faculty** class

to be able to read the input files and create your student arrays. Then you need to test the member functions of the **faculty** Class and show the results.

- b) To test the functions in the **Student** class, you can manually create the student objects and call different functions for this class.

**Note:** You need to store the information of at least **five** students in your input files (*studentst.txt*)