

COEN 243 Project

Deadline: Friday August 11th, 11:59 PM EDT

Type: Group (Not Compulsory)

Weight: 7%

Submission: Moodle

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 and staff 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: Employees

At Concordia University, we have two different categories: *students* and *staff members*. Below are the details for two categories of employees:

The initial information of *staff* members is maintained in a file called ***students.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 function 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.

- bachelor: 140 credit
- Master: 16
- Ph.D: 12

➤ StudentStatus (): This function 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 prints out all the information of one student (all the attributes).

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

- employee first name
- employee last name
- employee id
- data of hiring
- bonusCode
- current salary

You need to create a class **Staff** with the above attributes and following function members:

- **Setter and Getter Functions for of all the attributes**

- **Print_Stf_Info()** : This function prints out all the information of one staff (all the attributes).

- **Calculate_Salary()**: calculate the new year salary based on the bonus code as below:

Code Bonus %

A 8%

B 6%

C 3%

D 1%

E 0%

(Note: an x% bonus indicated x% increase of the salary)

Student_Staff Class:

You need to implement this class to read the input files and create the arrays of students and staff.

Attributes:

a pointer to an array of students

a pointer to an array of staff

Constructor():

Reads *student.txt* files and creates a dynamic array of students

Reads *staff.txt* files and creates a dynamic array of staff

Member functions:

Highest_GPA(): This function accepts an array of students and 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 size and returns the total number of undergraduate students

Same_Hire_Year(): This function accepts an array of staff, and size and a hiring year and prints out the name of all staff who have been hired in that year.

Part 2: Driver:

- a) In the driver you will test your system. You first create an object of Student_Staff class to be able to read the input files and create your student and staff arrays. Then you need to test the member functions of the Student_Staff Class and show the results.
- b) To test the functions in the Student and Staff classes, you can manually create the student and staff objects and call different functions for these two classes. Another option is to use the objects that you have created in part a.

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