Sindh Madressatul Islam University, Karachi

Department of Computer Science

Spring 2020

CSC 104 – OBJECT-ORIENTED PROGRAMMING

ASSIGNMENT 02

Due Date: July 25th, 2020

Plagiarism policy is minus-100%.

PROBLEM SET:

Create C++ Object-Oriented Source codes with the following classes, data members and member functions.

1. Class "***Person***" that stores the following characteristics of a person:
    a. Name
    b. Date of Birth
    c. Address

    The following public functionalities should also be designed in the "***Person***" class.
    d. Constructor(s)
    e. Getters, Setters for every data member (if necessary).
    f. Change of address

2. Another class "***Student***" that resides the following characteristics in it:
    a. Student Name
    b. Student ID
    c. Date of Birth
    d. Course(s) he/she is enrolled in
    e. Phone Number
    f. Mailing address
    g. Email address

    The following public functionalities should also be designed in the "***Student***" class.
    a. Constructor(s)
    b. Getters, Setters for every data member (if necessary).
    c. Change of email address
    d. Change of mailing address
    e. Change of Phone Number
    f. Enroll in a course

3. Another class "***Teacher***" that resides the following characteristics in it:
    a. Teacher Name
    b. Teacher ID
    c. Address
    d. Phone Number
    e. Email Address
    f. Date of Birth
    g. Course(s) he/she is teaching

    The following public functionalities should also be designed in the "***Teacher***" class.
    g. Constructor(s)
    h. Getters, Setters for every data member (if necessary).
    i. Addition of course(s)
    j. Removal of course(s)

4. Another class "***Course***" that resides the following characteristics in it:
    a. Course Name
    b. Course Code
    c. Number of students registered
    d. Teacher of the course

    The following public functionalities should also be designed in the "***Course***" class.
    k. Constructor(s)
    l. Getters, Setters for every data member (if necessary).
    m. Enroll a student
    n. Change of teacher
    o. De-enroll a student
    p. Plus(+) operator to be overloaded when two courses are to be merged as per following rules:

	Course 1	**Course 2**	**Resultant**
Course Name	Accounting	Finance	Accounting & Finance
Course Code	104	105	1045
Number of Students	17	10	27
Teacher	Mr. Akram	Ms. Ayesha	Mr. Akram & Ms. Ayesha

    In addition, all students registered in these different courses should be then enrolled in the resultant course with the invoking of this operator.

NOTE: Every class should have a function named printData() that should be overridden in derived classes.

The following point should be addressed in this system:

1. A student cannot be enrolled in more than 3 courses.
2. A teacher should not teach more than two courses at a time.
3. Each course should have a separate course code.
4. Two courses may have similar names but not the same course codes.
5. Multiple students and multiple teachers may have similar names but never a similar ID.
6. Inheritance should be implemented in this system (You will decide which class(es) need to be inherited from which).
7. There should not be any redundancies in objects and their data. i.e. if a class modifies the data of another class's object then the real data of that object should be updated. For Example: if a student enrolls himself/herself in a course, the number of students (4c) should be updated.
8. The data members and member functions are not given to be strictly followed as they are i.e. you may modify the placement of the data members and member functions to avoid redundant code where needed.

*********END OF PROBLEM SET*********

SUBMISSION INSTRUCTIONS:

For LMS

1. Code all files in a single project (Name you project as your roll number without dashes and your section after underscore e.g. ***csc19f123_AB***)
2. Make headers and implementations for all classes separate e.g. Student.h for class structure and Student.cpp for implementations, etc.
3. Compress your project folder containing all the files as a single ".zip" file having the same name as your project e.g. "***csc19f123_AB.zip***". (**Video Link**)
4. Upload the file on your LMS Assignment titled as "Assignment 02 – OOP".
5. You can update your assignment submission before the due date ends.
6. No late submissions are allowed on LMS.

For Email

7. In addition to submitting on LMS, you are also required to submit your assignment on email as well.
8. Attach the file (as mentioned in point 3) in email.
9. Write subject as "Assignment 02 – OOP – AB – csc19f123" (or CD instead of AB if your section in CD).
10. Write your full name and roll number in email body.
11. Send the email to oopspring2020smiu@gmail.com.
12. Email Submission's due date is the same as the due date mentioned in your assignment as well as on LMS.
13. Late submissions will be allowed on Email. However, late submission's acceptance is subject to instructor's approval and it can be accepted with penalized marks.

*********THE END*********