

CSC 2110--Computer Science I

Project 01

Project Description:

Goal: demonstrate everything you have learned during this course in your first project. It should be a fully-fledged C++ program, written entirely from the scratch by you, satisfying the requirements specified below.

Develop an employee information management system in C++ with object oriented concept and to introduce current technical issues in the field of object oriented programming (OOP). The objective is to manage courses', students', and instructors' records using operations like adding, modifying, deleting, searching, and listing records in a certain manner.

Basic features make up most of this project, but you can write your own code to add more features, and make this project more effective and better overall.

Here's a brief overview of the features:

1) **Add Records**

This feature allows you to add general information records. The student class with data members like student ID (required field), student name (required field), student age (required field), phone number, address, gender (required field), GPA (required field), hobbies (required field), and email address, is required to be declared and used in the system. All the added records in this student information management system are stored in a file.

2) **Sort Records and list them in ascending or descending order**

It sorts all the records in the system in terms of the feature specified by a user such as GPA, ID, or name in ascending or descending order. The system should be able to differentiate features specified by the user, sort all the records based on the specified feature, and then cout the results. For example, if the user chooses GPA (or other required fields), the results should be ordered by GPA (or other required fields). There are some improvements you can make to this feature to make the list look more attractive.

3) **Modify Records**

This feature allows you to modify the added records based on user's input. The modified records are then stored in file.

4) Calculate Salary

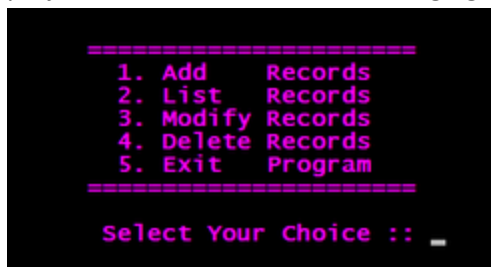
A full-time employee is usually fixed for a year. In addition, a full-time employee may receive a bonus. The salary of a part-time employee is usually calculated according to the pay rate per hour and the number of hours worked.

5) Delete Records

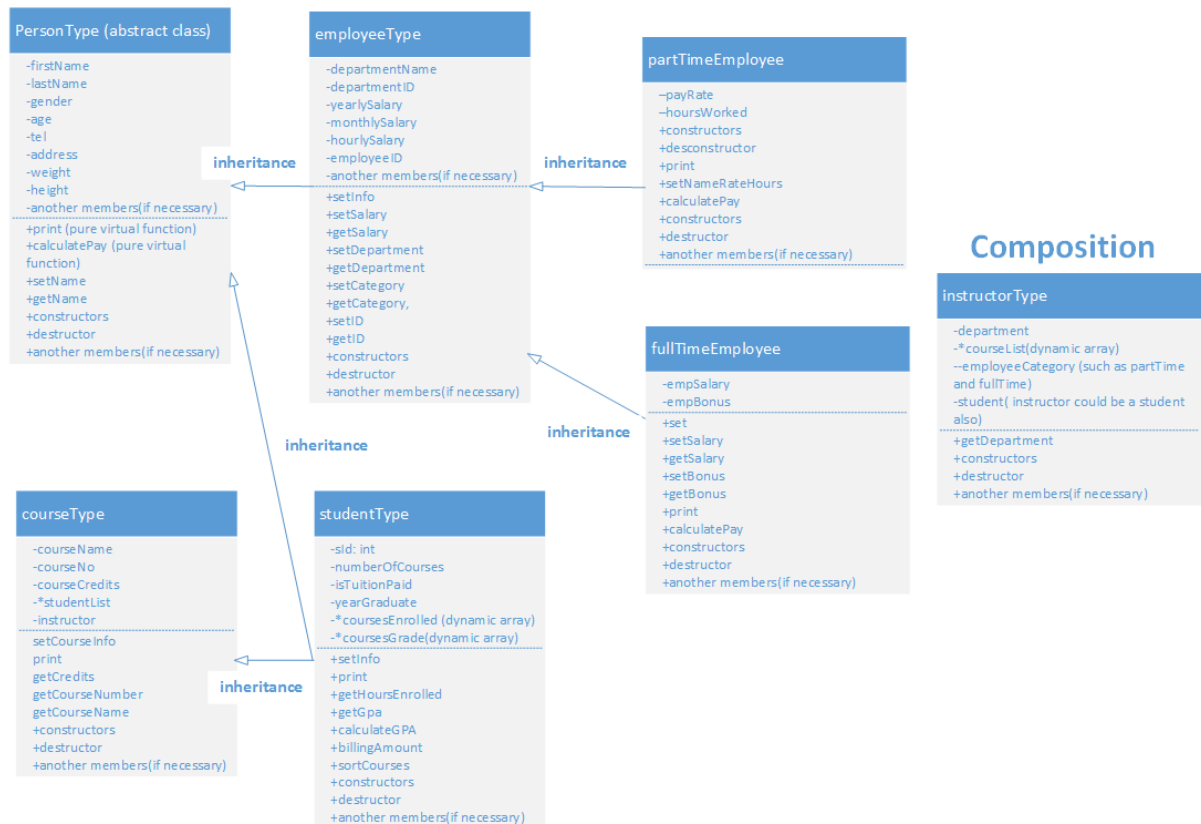
This feature is for deleting the added students' information from the file.

Requirement:

- 1) You may not use any other external libraries (e.g. for graphical interfaces). Generally a good rule of thumb is that if something was not covered in this course, you should probably not use it (if in doubt, email me!). Stick to a text interface. That is, you need to design a text menu that includes various operating functions. After the application is running, the first display is showing in the menu. Then select the desired operation performed by the user via the menu item project. Please refer to the following figure.



- 2) Employees' Information can be defined as classes s (preferably using advanced class features such as inheritance wisely), use the file to save the data. More precisely, classes that are needed to be defined in this project are personType, courseType, employeeType, partTimeEmployee, fullTimeEmployee, studentType, and instructor Type, as show in the figure below. Note that an instructor could also be a student taking courses; you should also consider this case in this project.
- 3) To help you have a better understanding of the key concepts mentioned in the classes, both simple data type and structured data type are suggested to be used in this project. Besides, pointers, array, user-defined functions, typedef, function overloading, abstract class, class and inheritance, composition, and virtual functions are also strongly recommend to be utilized to facilitate the implementation of this project. **Note vectors are not allowed in this project.**
- 4) Read students and instructors' data from a file and create objects accordingly. When there are any modifications, the file is needed to be updated also.



- 5) It is very important that you write easily readable, well-designed code.
- 6) Include a README file (.txt), with some basic documentation and instructions on how to use your program. Also include in this README what problems you had with your project, what the challenges were, and what would you have done differently if you could do it again.
- 7) Include a project description file (.doc). In the file, you should introduce your project, explain the design of your project (e.g., the architecture of project), and talk about challenges that you faced when you were working on your project and listed your solutions. (Optional) highlight unique features of your project.
- 8) Upload your project to Blackboard, satisfying the requirements. Pack everything into a single .zip file (i.e. your code, the README file, and the project description file). You will get bonus points for the use of more advanced C++ features.