Points: [100/100] Due Date: Oct. 20th, 2021

Project #1: Florida Poly Simple Student Record System

Submission Notes:

1. In addition to the code, please upload your report submission in PDF format. The report should have a clean and clear run of the system including screenshots.

2. Do not submit any zip or compressed files, binary files, or any other generated files. Do not put the code inside the PDF file. Submission of unnecessary files or binary files in addition to source files will make you lose the points of the assignment. The code must have the following:
    a. Use of descriptive names for the identifiers and active names for functions.
    b. Use consistent indentation through your code.
    c. Use consistent braces style through your code.
    d. Efficient utilization of functions. Make sure to reduce (*keep functions as simple as you can make them*) and reuse (*create functions for repeated parts of the code*) your code.
    e. File-level comments, these comments will include the file name, your name, section number, and date of last modification. In addition, these comments must have instructions on how we run the program and test it.
    f. Function level comments include the function's description, the function's parameters, and the function's return value. These comments will appear before each of the functions, not the prototypes.
    g. In function comments, these comments will include a description of the atomic functionalities within the bodies of the functions.

Project #1: Florida Poly Simple Student Record System

Write a C++ program to implement a simple student record system. The following points describe the requirements that the program should have:

1. A `class` called `Person` has two data members:
   - i. `name` of type `string`.
   - ii. `age` of type `uint8_t`.
2. A `class` called `Course` that has:
   - i. A data member called `code` of type `string`.
   - ii. A data member called `title` of type `string`.
   - iii. A data member called `grade` of type `double`.
   - iv. A data member called `creditHour` of type `uint8_t`.
   - v. A member function `setAll` that will be able to set all the values for the data members of this class.
3. A `class` called `Student` that inherits all the properties of the Person class. The `Student` class has:
   - i. A data member called `totalCreditHours` of type `uint32_t`.
   - ii. A data member called `numberOfCourses` of type `uint32_t`.
   - iii. A data member called `uid` of type `uint32_t` used for university ID.
   - iv. A data member called `gpa` of type `double`.
   - v. A singly linked list of `Course` objects called `courses`.
   - vi. A member function called `addCourse` that will add a new course to the courses singly linked list and set the values of the data members for the course.
   - vii. A member function called `deleteCourse` that deletes a course from the courses singly linked list.
4. A class called Records that has:
   - i. A data member called `numberOfStudents` of type `uint32_t`.
   - ii. A data member called `averageGpa` of type `double` and holds the *average* GPA of all students.
   - iii. A doubly linked list of `Student` objects that is called `students`.
5. Your `classes` should have a setter and a getter member functions for every data member in the class.

6. An operator overloading for << in each class to handle printing the content of its object. For example, this is a print out of the `Student` class:

```
Name: Jane Doe
Age: 38
Credit Hours: 30
GPA: 3.5
Number of Courses: 12
COP3530: Data Structures & Algorithms ( GPA: 3.5 ) 3.5 Credit Hour(s)
CDA2108: Introduction to Computer System ( GPA: 3.9 ) 3.9 Credit Hour(s)
```

7. When the program starts running:

   i. Loads the student records from a file called `default.csv`. This file is a Comma Separated Values (CSV) file. The file has a row (record) for every student. Each row in the CSV file has the following format:

   ```
   uid,"Name",age, Total Credit Hours, Number of Courses, gpa
   ```

   ii. Loads the course records of each student from a file called coursesDefaults.csv. This is a single file that contains the course records for all students. Each row in this CSV file has the following format:

   ```
   uid,"Course Code", "Course Title", Credit Hours, Grade
   ```

8. The program will show a menu that offer the following functionalities:

```
        Welcome to FlPoly Simple Student Record System
        ----------------------------------------------
Please select one of the following:
 [1] Print all records
 [2] Print the record for a single student
 [3] Add a student
 [4] Delete a student
 [5] Add a course to a student
 [6] Delete a course from a student.

 Enter your selection:
```

9. Make sure to implement all functionalities in the above menu and adding all the other necessary data members and member functions.

Guidelines

1. Make sure to include sufficient documentation for your code. That includes comments and choosing the appropriate names for your data members, functions, classes, and objects.
2. Proper formatting of the code.
3. Separate the implementation and interface in all classes.