COSC 220 - Computer Science II
Project 1

Dr. Joe Anderson

Due: October 13, 2019

1 Description

You will create a program that emulates an interactive database of students and the courses they have taken. The program will use the command-line and allow a user to 1) create, 2) update, and 3) delete student records (often referred to as "CRUD" operations). Each student will have the following metadata: name, date of birth, and major. In addition, each student will have a collection of courses taken; each of these courses will have a name, department, semester taken, and grade received.

2 Specifications

The executable should be called `StudentDB` and will be executed at the command line.

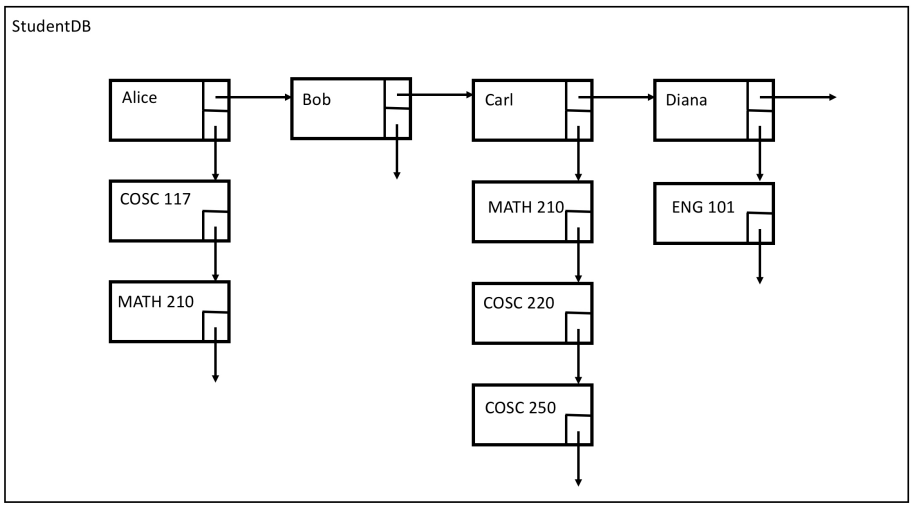
Your main database structure should be its own class, called `StudentDB` which contains a singly linked list of students, and methods to create, update, and delete student records. The database should also have member functions to access metadata, such as number of students in the database, and to print a list of the students.

Create a class called `Student` which will be the data contained in the nodes of the `StudentDB` class, along with member methods to update and delete the student. Each student object will also have as a member attribute the head of a singly linked list of courses taken by the student.

Each course recorded should be of a class called `Course` that contains the name, department, semester, and grade received. All the user to input the class names, departments, and majors.

To delete a student from the database or a course from a single student, you can use name-based search (e.g. enter the student's name, find it, and remove it from the linked list). More complicated mechanisms are also acceptable as long as the instructions presented by the program are clear.

The structure of an example database can be visualized as:



Include a `README` file that outlines the functionality of the program, providing a brief user-guide.

Any and all user input should be validated!

3 Submission

Upload your project files to the course canvas system in a single `.zip` file.

Turn in (stapled) printouts of your source code, properly commented and formatted with your name, course number, and complete description of the code.

Also turn in printouts reflecting several different runs of your program (you can copy/past from the terminal window). Be sure to test different situations, show how the program handles erroneous input and different edge cases.

4 Bonus

You may or may not add the following features for extra credit:

- (5 pts) Allow the user to select a student's major from a pre-populated list, which is read from a file when the program starts. Allow the user to add a new one, but then save that option to the file to be used again later.
- (5 pts) Same as above, but for the department field
- (5 pts) Same as above, but for the grade field. You do not have to allow new grade types to be saved.
- (10 pts) Allow the user to save the database to a file, and load that file into memory when the program starts.