

Data Structures Project

The main objective of this project is to create a Student Information System (SIS) for the students of the Antonine University. This SIS will allow already registered students to login and check their course schedule. They can add new courses, drop already enrolled courses, and swap two courses. Do not check for time conflicts between enrolled courses, however when listing the courses enrolled by a given student, you may display them in alphabetical or chronological order. This SIS will also allow new students to sign up for a new account. In this case, the newly students is not enrolled yet in any course.

The SIS allows the students to see the list of other students that are already enrolled in a particular course.

For this purpose, we will define the University Courses as a doubly linked list of all the courses that are offered at the Antonine University. This is a static list that can be read from a file and will not be changed by the program. Furthermore, we will define the University Students as a singly linked list of all the students that are attending the Antonine University. This list may vary over time as new students can join this university. Thus, you need to read the list of students from a file when your program starts and write it back at the end of your program in order to save the changes that you made to the list of students and to the list of courses that are enrolled by each student.

A Student is identified by a Student ID, a First Name, a Last Name, an Email Address, and a doubly linked list containing the courses that he/she already enrolled. A Course is identified by a Course ID, a Course Name, an Instructor Name, a Time Schedule when the course is taught, a Course Capacity designating the maximum number of students that are allowed to enroll in this course, and a singly linked list of students that are currently enrolled in this course.

A particular attention should be paid to exclude the possibility to enroll in the same course twice or in two courses with the same name.

It is strictly forbidden to use arrays in this project except for the arrays of characters defined in Table 1. You can use singly or doubly linked lists instead. You may also use the data structures defined in Table 1.

Data Structures Project

Table 1 - Data Structures to be used in the project

<pre> struct Student { int studentID; char firstName[30], lastName[30], emailAddress[50], password[20]; EnrolledCourseList enrolledCourses; Student *next; }; </pre>	
<pre> struct Course { int courseID, courseCapacity; char courseName[50], instructorName[50]; defineYourTimeScheduleType timeSchedule; EnrolledStudent *attendees; Course *next; Course *previous; }; </pre>	
<pre> struct UniversityCourseList { Course *head, *tail; }; </pre>	<pre> struct EnrolledCourseList { EnrolledCourse *head, *tail; }; </pre>
<pre> struct EnrolledStudent { Student *studentReference; EnrolledStudent *next; }; </pre>	<pre> struct EnrolledCourse { Course *courseReference; EnrolledCourse *next; EnrolledCourse *previous; }; </pre>

- The structure **UniversityCourseList** is a doubly linked list of the **Course** offered by the university.
- The structure **Course** is a data structure representing a university course. It contains two pointers 'next' and 'previous' that are required for the proper operation of the doubly linked list **UniversityCourseList**. It also contains a singly linked list 'attendees' that lists all the students that are enrolled in this course.
- The structure **Student** is a data structure representing a university student. It contains the pointer 'next' that is required for the proper operation of the singly linked list **UniversityStudentList**. It also contains a doubly linked list 'enrolledCourses' that lists all the courses where this student is already enrolled.

Your task is to write a program and test it on an input text file(s) containing already some Students already enrolled in some course. You can choose the format of the input text file(s) according to your needs.

Data Structures Project

Instructions:

- a) This project can be done by a group of one or two students. When working in a group of two students, the efficiency of the team work will also be evaluated.
- b) You need to send an email to your instructor, by the 11th of December 2022, specifying whether you will be doing the project alone or in a group. In the latter case, you will need to provide the name of your teammate.
- c) The deadline for submitting your project through Microsoft Teams (source code and input file(s)) is by the 20th of December 2022 (midnight).
- d) The dates for defending your projects will be specified later on by your instructors. During the project defense, you will briefly present your project and you will be assigned a task that can be related to any part of the course.

Extra Tasks:

If your project is working perfectly, you can improve your project by adding extra functionalities to your project. This will allow you to get a higher grade on your project only when the basic requirements of your project are already fulfilled. These extra tasks could be:

- Adding the possibility for an instructor to log in and see the list of courses that he/she teaches.
- Adding the possibility for an instructor to do the class attendance and the students can view it.
- Adding the possibility for an instructor to add grades for the course and the students can view them.
- Any other functionality that you deem as useful.