COEN 244 (Winter 2025) - Assignment 1

Assignment 1:	Classes and Objects
Deadline:	Monday, February 3 by 11:59pm
Type:	You can work individually or in groups of two students.
Weight:	5%

Submission instructions:

- Create a cpp file for each question
- Add name and student ID of both group members at the top of each cpp file
- Compress the files using zip or other tools
- Submit the zip file on Moodle; assignments sent by email will not be corrected
- Do not submit executable files
- Late submissions will not be accepted

Questions:

Q1. (15 points) Write a class called Student that represents the students of a university. A student is identified using a student id (string), first name (string), last name (string), address (string), and email (string). The class should have one or more constructors, accessing member functions, a function that prints information about a student. Write a driver (i.e., a program with a main function) to test the Student class.

Deliverable: A zip file that contains three files: Student.h, Student.cpp, TestStudent.cpp

Q2. (15 points) Write a class called CourseSection that represents the courses offered by the university. A course section is identified using a course section id (string), a course section title (string), a course description (string), and a capacity (int). Similar to class Student, the class CourseSection should have one or more constructors, accessing member functions, a function that prints information about a course section. Write a driver to test the CourseSection class.

Deliverable: A zip file that contains three files: CourseSection.h, CourseSection.cpp, TestCourseSection.cpp

Q3. (15 points) Write a class called Professor that represents professors of the university. A prof has an id (string), first name (string), last name (string), address (string), and an email (string). The class should have one or more constructors, accessing member functions, a function that prints information about a prof. Write a driver to test the Professor class.

Deliverable: A zip file that contains three files: Professor.h, Professor.cpp, TestProfessor.cpp

Q4. (15 points) Write another class called Room, which represents class rooms. A room has an id (int), a capacity (int). Similar to Q1, Q2 and Q3, provide all the components of the class Room. In addition, you need to write a drive to test the class Room.

Deliverable: A zip file that contains three files: Room.h, Room.cpp, TestRoom.cpp

Q5. (40 points) Write a program called University.cpp that tests how the above classes are used together. The program should register students in course sections, assign professors to course sections, assign course sections to rooms. Note that we cannot go beyond the capacity of course sections or rooms.

The program should have three 2-dimension arrays:

- An array called registration to keep track of course sections and students
- An array courseAssignment to keep track of courses and rooms
- An array professorAssignment to keep track of professors and the courses they are assigned to teach.

The program should support the following functions:

Make sure we test all the functions in our cpptest files

- Return the list of students taking a particular course
- Return the list of course taught by a give prof
- Return the room where the course section takes place
- Search if a given students is taking a given course
- Search if a given prof is teaching a given courses,
- Etc.

Do we need to implmenet these assumptions even if we dont use arrays

You can make the following assumptions:

- The maximum capacity of a course section is 100 students
- The maximum capacity of a room is 120
- The maximum number of courses a prof can teach is 3.

Deliverable: A zip file that contains Student.h, Student.cpp, CourseSection.h, CourseSection.cpp, Room.h, Room.cpp, Professor.h, Professor.cpp, University.cpp.

Assignment Marking Scheme for each question:

- Program correctness (80%)
- Program clarity, completeness, and accuracy and readability (5%)
- Comments - description of variables and constants (5%)
- Test cases should be comprehensive enough to cover your program to test if it is bug free (10%)

Note on using GenAI tools such as ChatGPT:

You are allowed to use GenAI as an educational resource. If you decide to do that, **you must provide the GenAI tool you have used, a list of all the prompts** you **used, and a detailed explanation of your contribution.** Note that the TAs reserve the right to meet students and ask them questions about their assignments and they suspect plagiarism due to the use of GenAI.