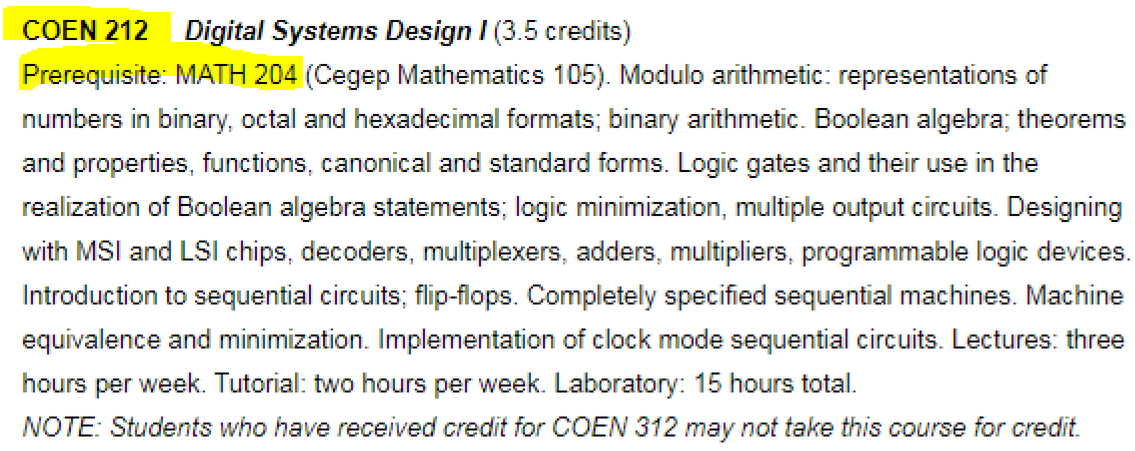
COEN 352 Fall 2022 Assignment 3

This assignment practices graph data structure and associated algorithms. We aim to develop a simple application that output the prerequisite course numbers given a course number. We take the example of the description in the document here.

https://www.concordia.ca/academics/undergraduate/calendar/current/sec71/71-60.html#coen

Under the tag of COEN program, we can see the description of each course and the prerequisite of the course. For example,



COEN 212 ***Digital Systems Design I*** (3.5 credits)

Prerequisite: MATH 204 (Cegep Mathematics 105). Modulo arithmetic: representations of numbers in binary, octal and hexadecimal formats; binary arithmetic. Boolean algebra; theorems and properties, functions, canonical and standard forms. Logic gates and their use in the realization of Boolean algebra statements; logic minimization, multiple output circuits. Designing with MSI and LSI chips, decoders, multiplexers, adders, multipliers, programmable logic devices. Introduction to sequential circuits; flip-flops. Completely specified sequential machines. Machine equivalence and minimization. Implementation of clock mode sequential circuits. Lectures: three hours per week. Tutorial: two hours per week. Laboratory: 15 hours total.

NOTE: Students who have received credit for COEN 312 may not take this course for credit.

If we trace from a given course, such as coen 352, we will find its prerequisite course is coen 244 whose prerequisite has coen 243, and so on. Therefore, the course sequences form a graph and the prerequisite is a path in the graph.

Problem 1(5 marks): Define a graph file named coen_course.gph for all the courses from COEN 212 to COEN 498, following the example source code released on Moodle site under the week of lecture on Graph. The node is a course. The prerequisite is the edge between two courses.

Problem 2 (30 marks): Program a graph class that creates a graph that stores all the courses and their relations on prerequisite based on the file coen_course.gph. Define two functions in the graph class

(1) return a sequence of prerequisites. The courses in a sequence are separated by tokens such as , or space. [hint: if needed you can also change the return values to other types such as string[]]

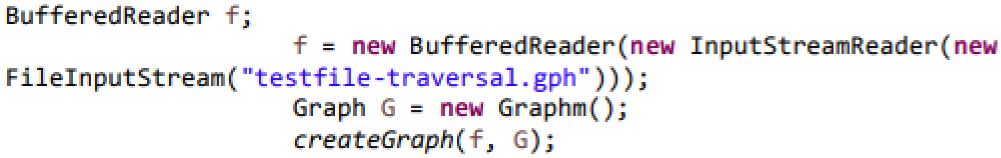
    public string getPrerequisistePath(string courseCode)

(2) return boolean value if a source course is a prerequisite for the given destination course.

    public boolean isPrerequisite(string sourceCourse, string destinationCourse)

Problem 3 (20 marks): Define cases (1) test case on getPrerequisistePath(); (2) test case on isPrerequisite();

Please do not give any absolute path or other relative path. The reason is, the grading will replace your file with a test file under the same name. Therefore, changing the file path rather than the current directory may cause your submission code not working.



```
BufferedReader f;
                    f = new BufferedReader(new InputStreamReader(new
FileInputStream("testfile-traversal.gph")));
                    Graph G = new Graphm();
                    createGraph(f, G);
```

Submission Specification:

This is a group of two or individual assignment. The submission should contain

(1) all the source code of problem 1,2 and 3;
(2) the graph file;
(3) Please have screenshots of running the test cases and the status of the unit testing result from your IDE. A pdf file of the screenshots of running unit test cases. (10 marks, 5 marks for each test case)

in a single archive as .z, .gz. .tar, .zip. NO RAR file will be graded, following the naming convention
[SID_1]_[SID_2]_A3.zip or
[SID_1]_[SID_2]_A3.gz   or

[SID_1]_[SID_2]_A3.tar