



Final Project: Student Course Registration Prerequisite Checking System

Description: This project requires the use of a **2-child tree** data structure to store different courses and their prerequisites. Based on this list, a student may or may not take a specific course.

*** A 2-child tree is a tree that each of its nodes have a maximum of two children meaning that each node can have (0, 1, or 2 children).**

Basic Idea: This program will input the university course list with their prerequisites ONCE at the start of program by reading this information from a file (**this info is given in the below table**). Then, whenever a student is willing to register a number of courses, he/she will enter the courses they studied and the ones they want to register in. The system will allow or disallow the student from registering the course based on stored course prerequisites. The project will be implemented in C++. This project is console based. Graphical representation of each student's course structure is a **BONUS**.

Course list and prerequisites are shown in the table below. Each course is defined by: (1) **Course Code** which is 3-digit number (e.g. 102, 402, ..) and (2) **Course Name** which is a string with max length of 40 characters.

Course Code and Name	Prerequisites
101Introduction to Programming	None
207Fundamental of Data Structures & Algorithms	101Introduction to Programming
311Computer Architecture	101Introduction to Programming
217Advanced Computer Programming & Concepts	207Fundamental of Data Structures & Algorithms
313Software Engineering	311Computer Architecture
415Compiler Design	311Computer Architecture
304Analysis and Design of Algorithms	217Advanced Computer Programming & Concepts
419Theory of Computing	217Advanced Computer Programming & Concepts
315Operating Systems	313Software Engineering
404Database Systems	313Software Engineering

NOTE:

Sample Run:

Enter the courses which you already studied:

101Introduction to Programming
207Fundamental of Data Structures & Algorithms
311Computer Architecture

Enter the courses that you want to study:

217Advanced Computer Programming & Concepts
404Database Systems
313Software Engineering

Processing...

The following courses are OK to take:

217Advanced Computer Programming & Concepts
313Software Engineering

The following courses are CANNOT be taken:

404Database Systems

Implementation Idea:

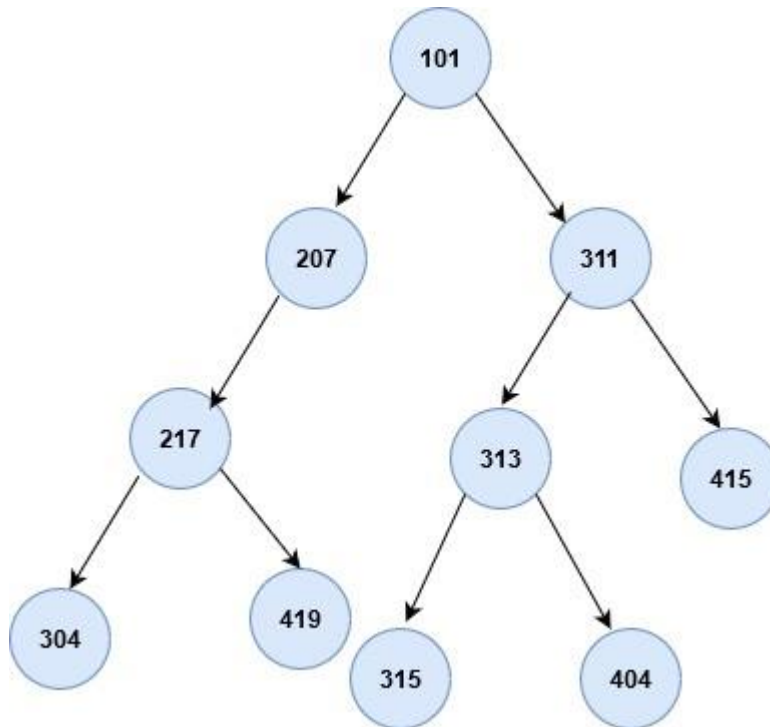
1. **First** you will build your main tree from the table given above. This can be done by storing the information in a (.txt, .csv,) file and building your tree from it. If you chose to use a .csv file you can store the info in the table in this form :

CourseCodeAndName, prerequisiteCourseCodeAndName

For example:

101Introduction to Programming, None
207Fundamental of Data Structures & Algorithms, 101Introduction to Programming
311Computer Architecture, 101Introduction to Programming
217Advanced Computer Programming & Concepts, 207Fundamental of Data Structures & Algorithms
313Software Engineering, 311Computer Architecture
415Compiler Design, 311Computer Architecture

2. **Second** you will need to extract the course code from each string to build your tree. Each tree node will have the course code **only** as its data. **When** a course node has two children you will always assume that the child entered is the left child and the child entered second is the right child. For example, the sample in the table above will be represented in the form of the following tree:



3. The input will be the courses that the student has finished and the courses that he would like to take. You will write your own top-down searching algorithm to find out if the student can take this course or not.

Detailed Stages and Deadlines:

1. Form Groups

- 3 – 4 students per group
- Deadline: 15th/JAN/2022 at 11:00 PM.

2. Delivery Details

- You will deliver a C++ program that performs each function required in the project. Please, write clean code.

NOTE: Plagiarism is **NOT ALLOWED**. And if **DETECTED** or **SUSPECTED**, tough actions will be applied.