**Data Structure and Algorithms**

**Semester Final Project**

**Student Prerequisite Subject Management System**

**Group Members**

|  |  |
| --- | --- |
| **Name** | **ID** |
| **Hussain Haider** | **30185** |
| **Muhammad Tayyab Khan** | **30579** |
| **Niamat Ullah** | **30142** |
| **Syed Shahzab Ali Shah** | **29305** |

**Submitted to:**

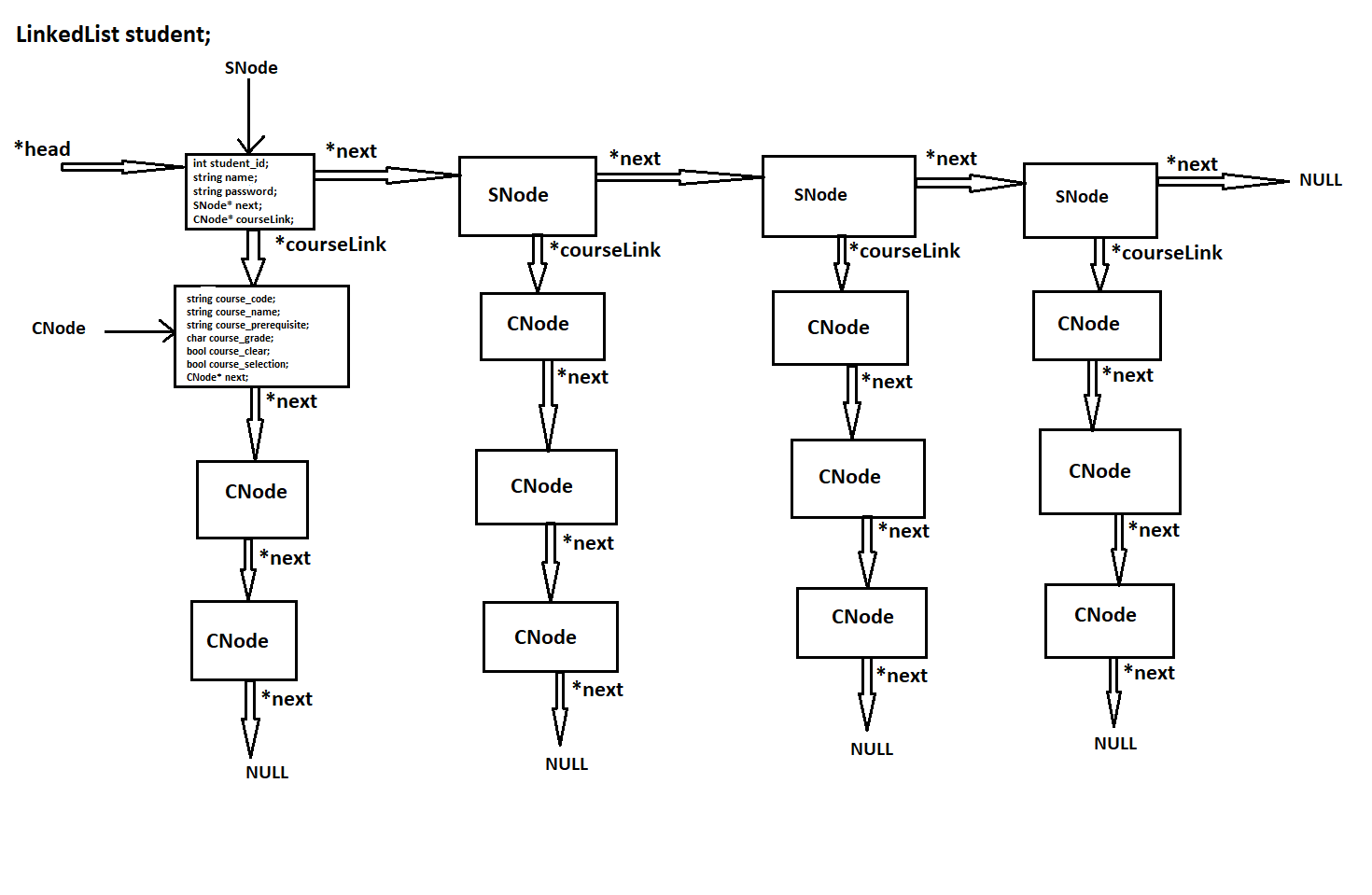
**Mr. Muhammad Awais Ali**

**Project: Student Prerequisite Subject Management System**

**Introduction:**

Our project is Student Prerequisite Subject Management System. It is a system in which student can register for new course or checked his/her cleared courses or check prerequisite course of any subject etc. etc. This system will allow student to register for any courses if he/she has cleared its prerequisite course.

The data structure used in this system is Linked List. The basic structure is as follows:



We have student linked list. We can add student node (student account (SNode)) and to each student we can add different course nodes (CNode) through \*courseLink link by creating a multilevel linked list.

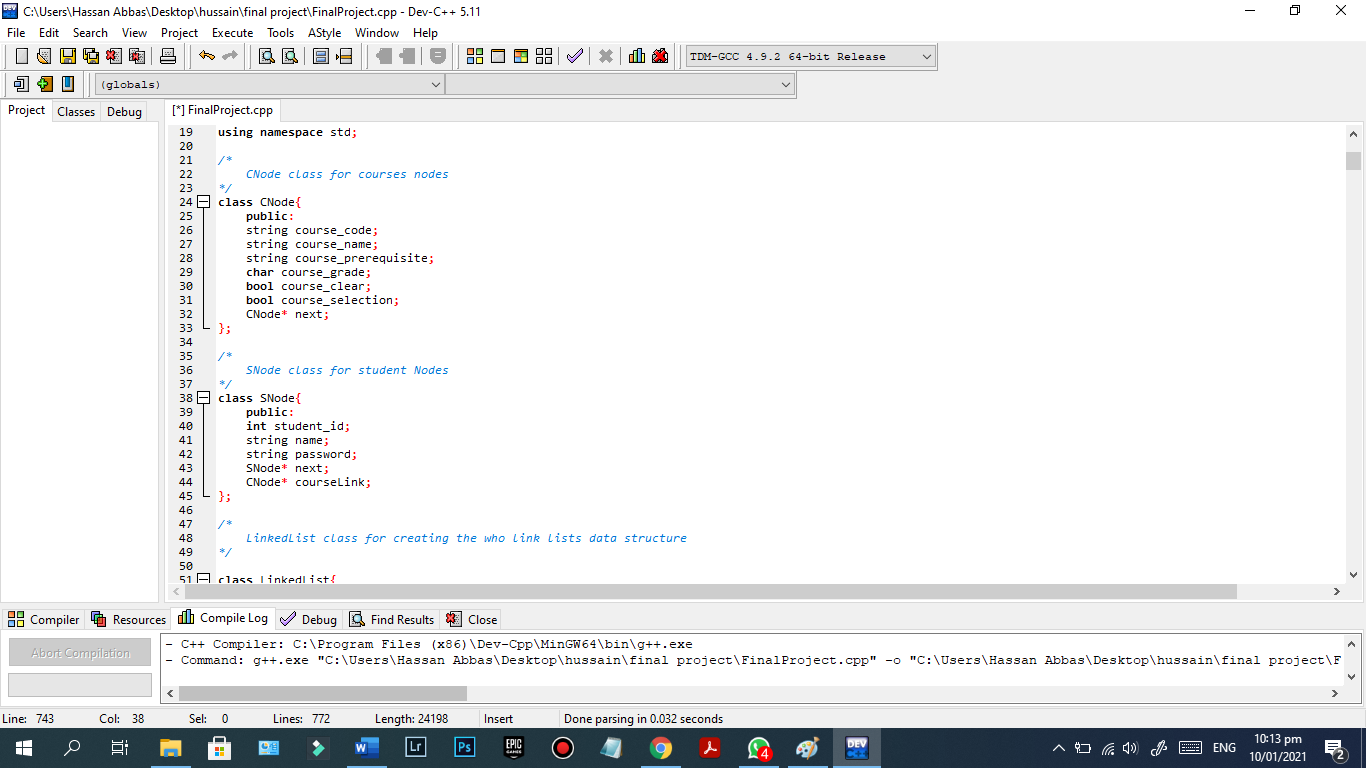
In this system we have two types of user.

1. Admin
2. Student

The admin has the most control of the system. Admin can add new student accounts, add new course for the student to register, update data of already added courses or view all the different data in the data structure. The admin has his own id and password through which he can access the system.

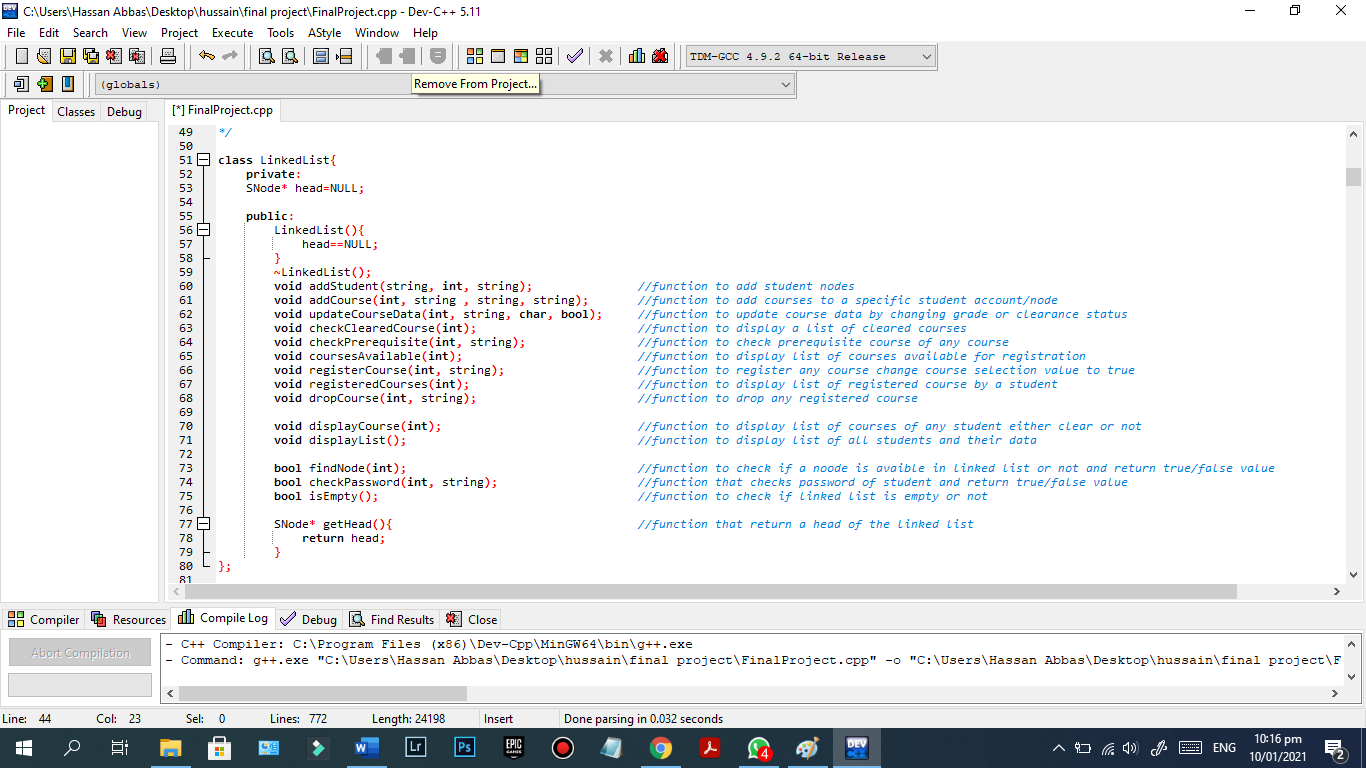
The student on the other don’t have the capability to change any data in the data structure but can view all data related to him/her like cleared courses, courses available for registration, register for any courses if its prerequisite is cleared or not or drop any registered course.

**Classes:**



**CNode** class is for creating Course nodes. This class has all the instance variable to save data related to courses like course code, name, grade, prerequisite courses etc. and a pointer to save address or link to the next course node

**SNode** class is for creating students’ nodes. This class has all the instance variable related to student to save its data like name, id etc and two pointers. next pointer used to link to the next SNode and coursLink is used to link to the CNode for courses.

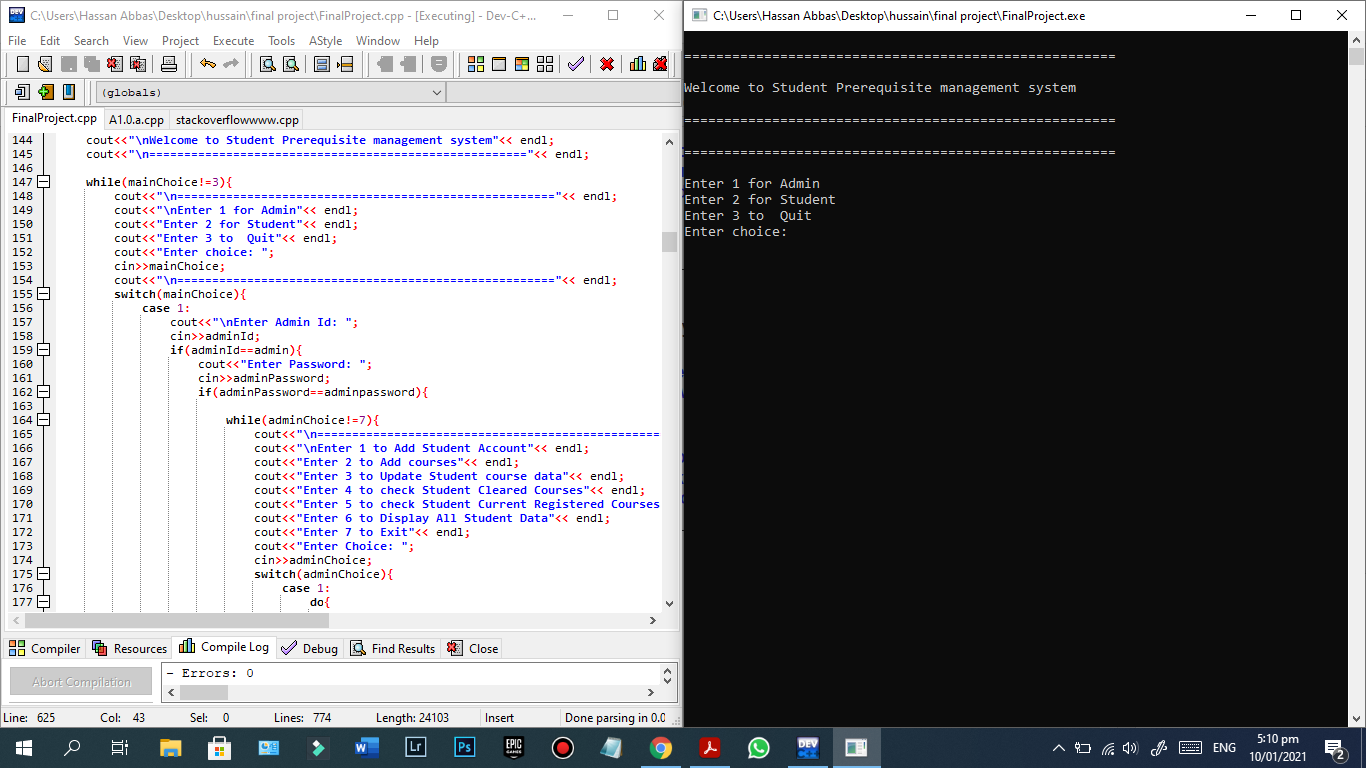


**LinkeList** class is for creating a linked list data with constructor and destructor and all the different functions to perform different operation and the head pointer.

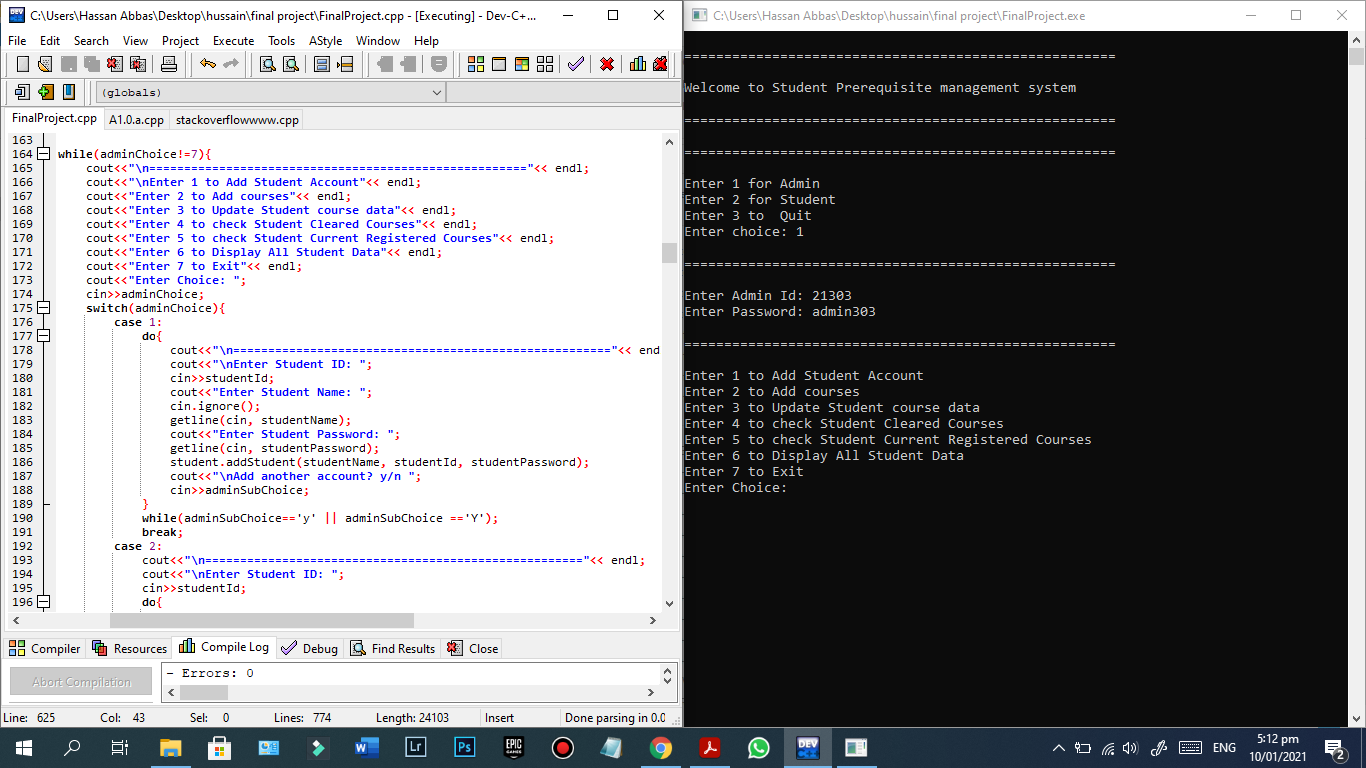
**Output and its explanation:**

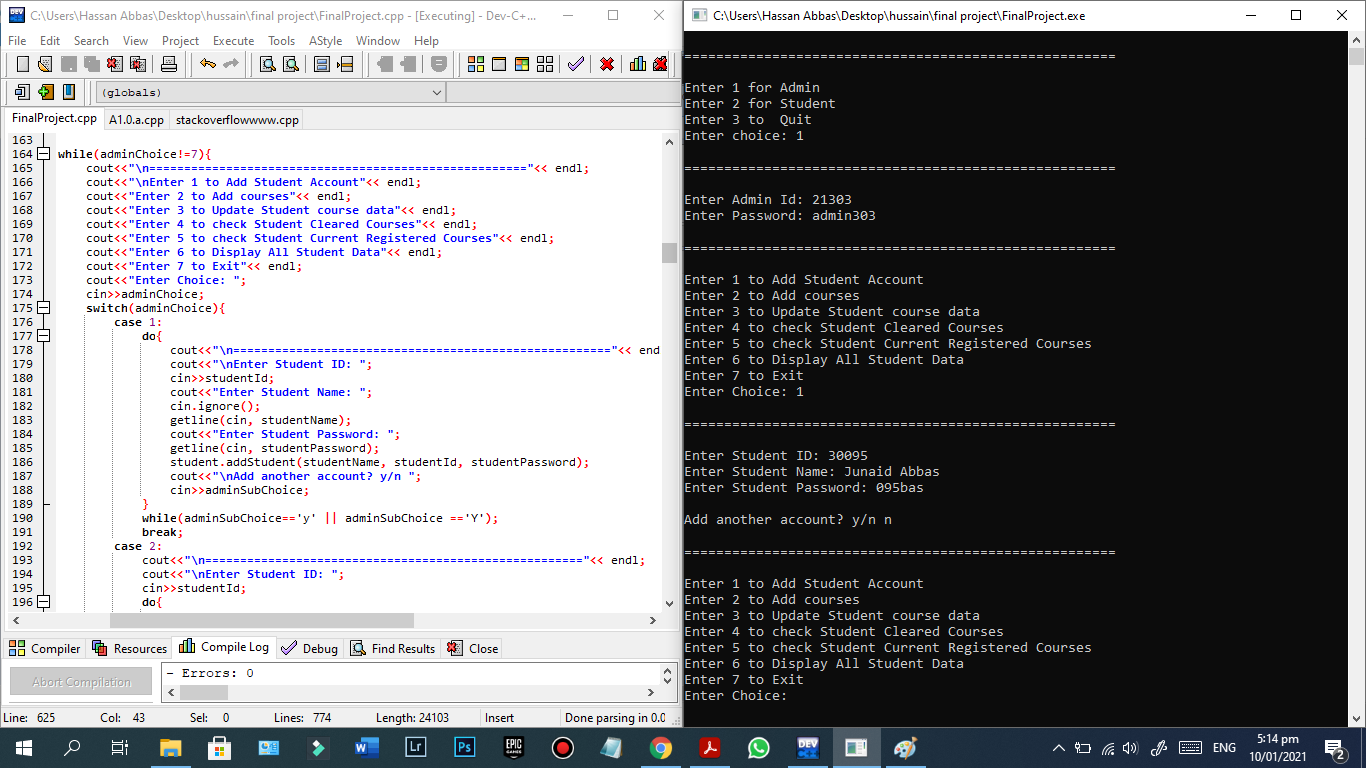
Following is the output of the program with explanation:

All the function discuss below are declared in the LinkedList class and all menus either submenu or main menu are looped and will execute until the user exit it.

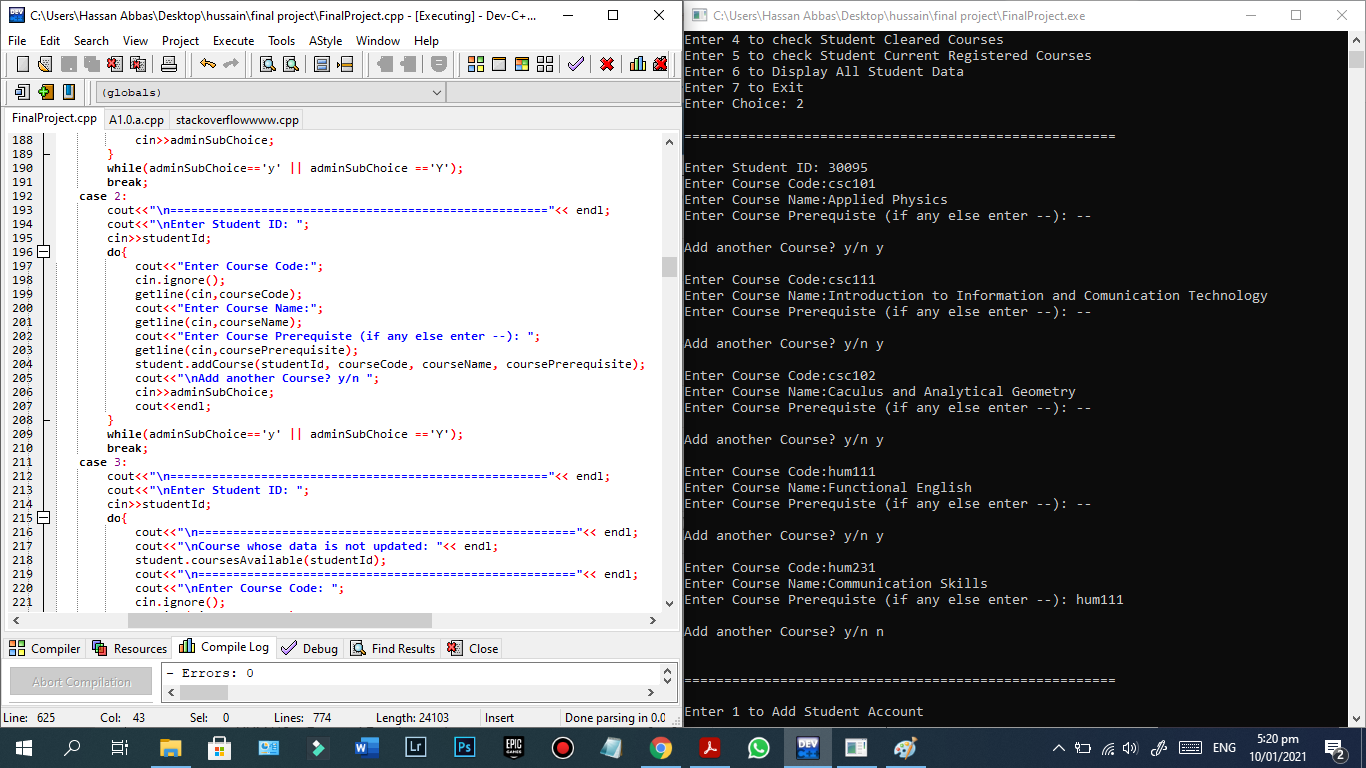


On start of the program we have main menu where user can enter the option based on who they are option 1 for the admin and option 2 for the student and option 3 if the user wants to quit the program

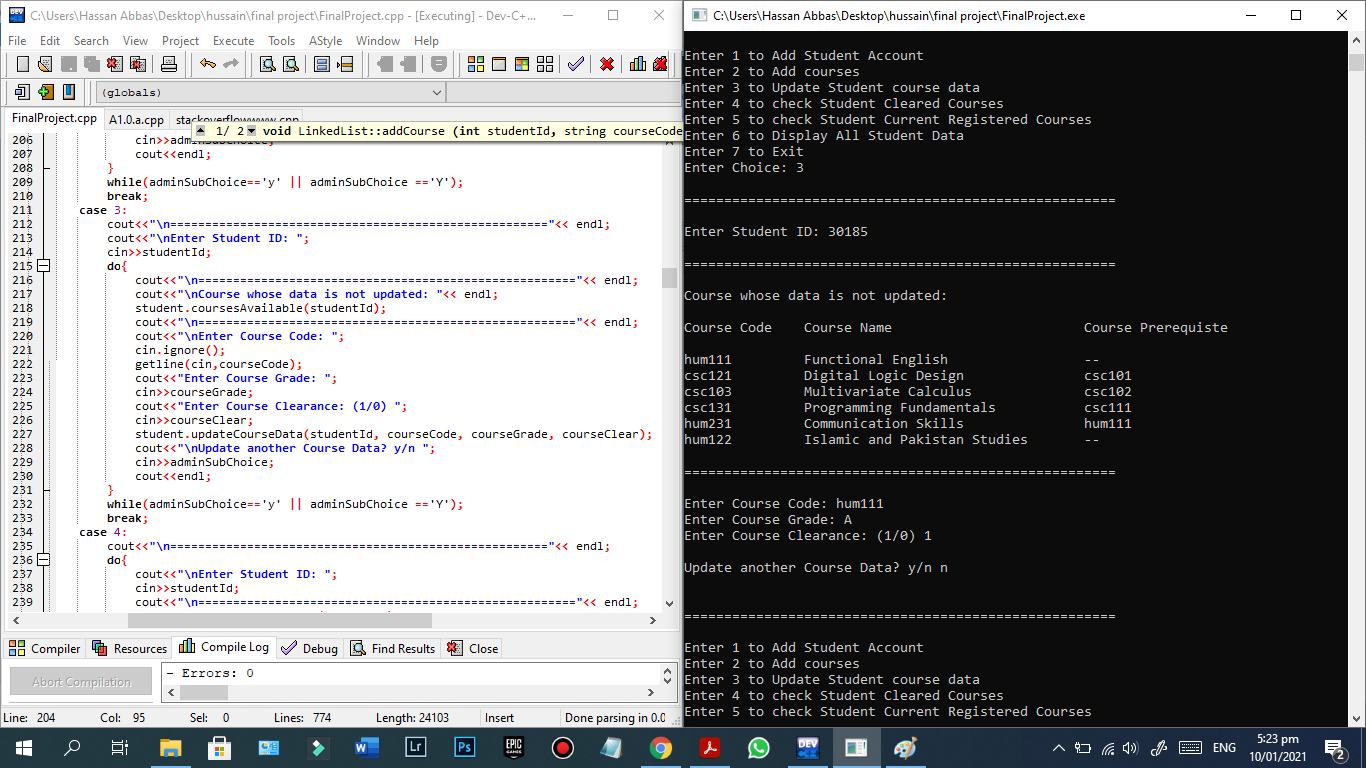
Upon entering option 1 for we enter the admin menu but before that the user will be asked his/her admin id and password, if the id and password is correct then we enter the menu else the program control will go back to the main menu. In the admin menu the admin has 7 option to select from. With every option admin can perform different function which are explained below.



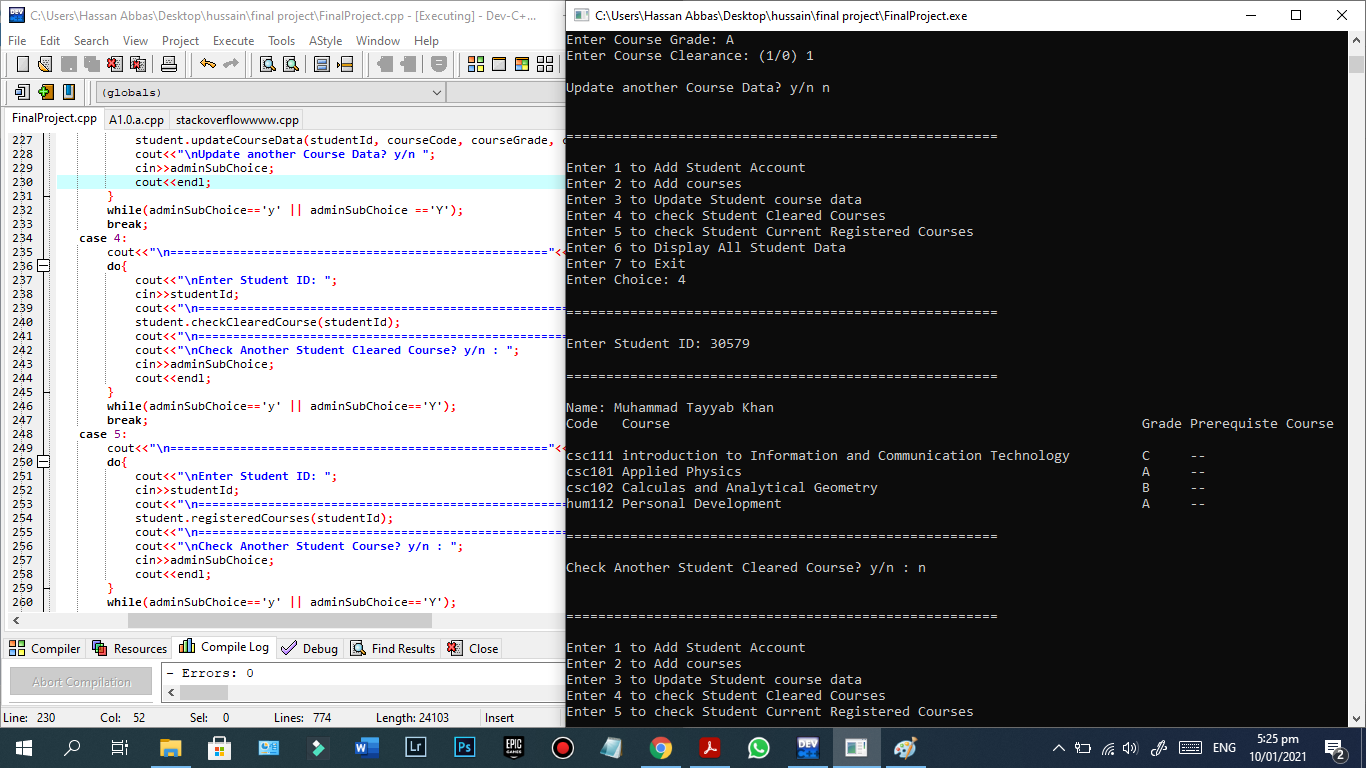
On entering option 1 admin can add student account (mean adding SNode in student linked list). The admin will be asked to enter student id, his/her name and a password for him/her so he/she can access the system later. The admin can keep on adding student accounts as many as he wants and if not, he can enter “n” and exit back to the admin menu and from there can perform other operation.



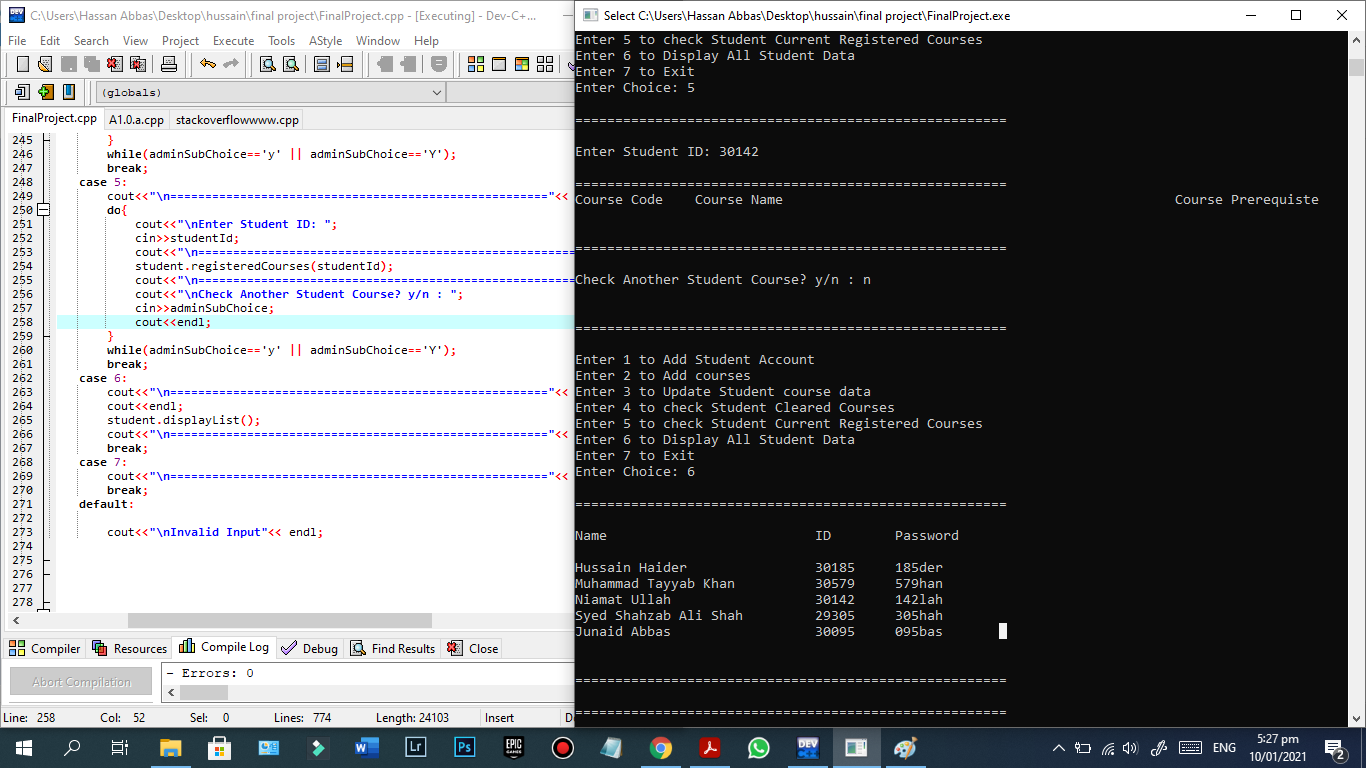
On entering option 2 the admin can add courses for any student by entering his/her id and then enter as many courses as he wants to and if the admin want add courses to another for another student, he has to exit this submenu of admin and select the option 2 again for add courses or do other operation.



On option 3 the admin can update data of courses of any student. The admin will be provided with courses of which data is updated and the admin enter any course code and update its data like grade and its clear status and can keep on doing for as many as he wants. The coursesAvailable function will call to display the list of courses of which data is not updated and updateCourseData will be call to update course data.



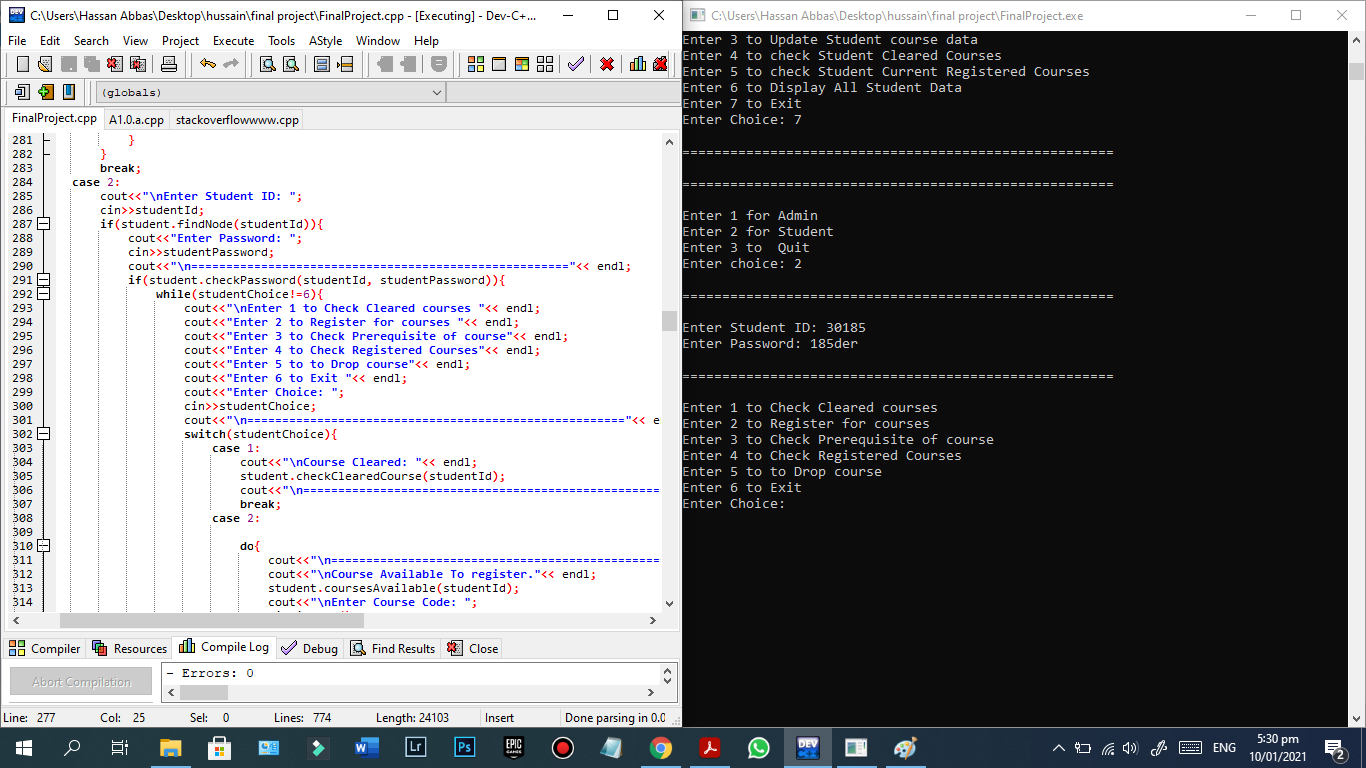
On option 4 the admin can check any student cleared courses with its grades, prerequisite course etc. checkClearedCourse function will be call for this operation.



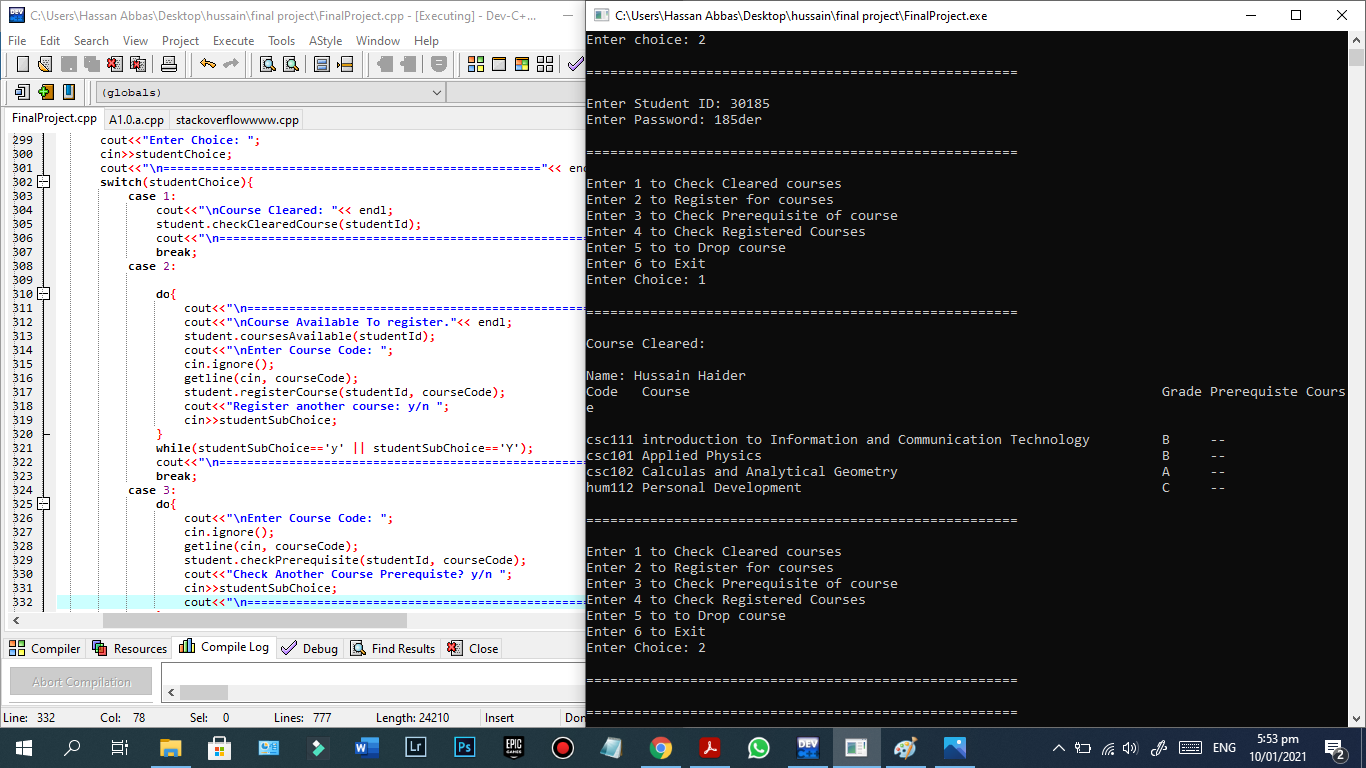
On entering option 5, the admin can check any student currently registered course by entering his id. courseRegistered function will be called to display list of registered courses.

On entering option 6, the admin can check all student added into the linked list. displayList function will be called to display the list of students with their id, name and password.

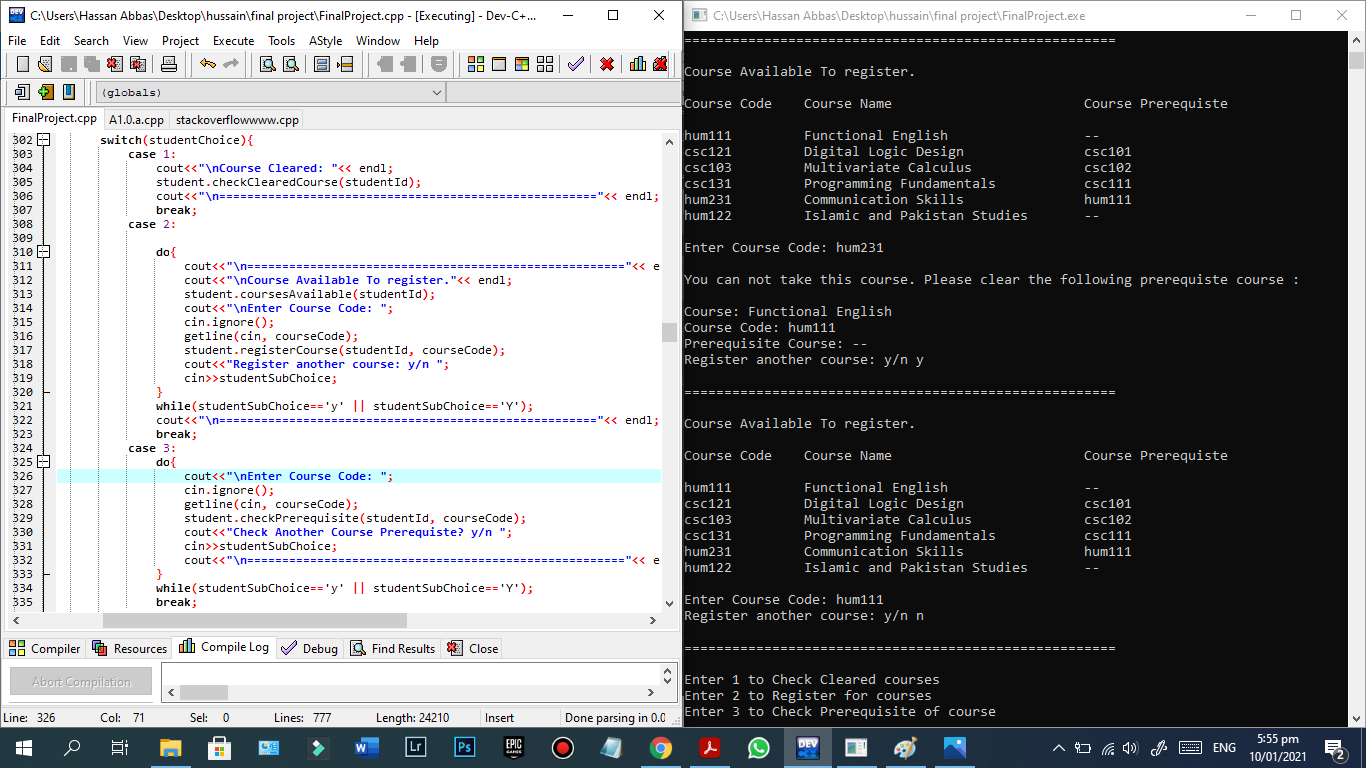
On entering option 7, the admin will exit the admin menu and enter the main meu.



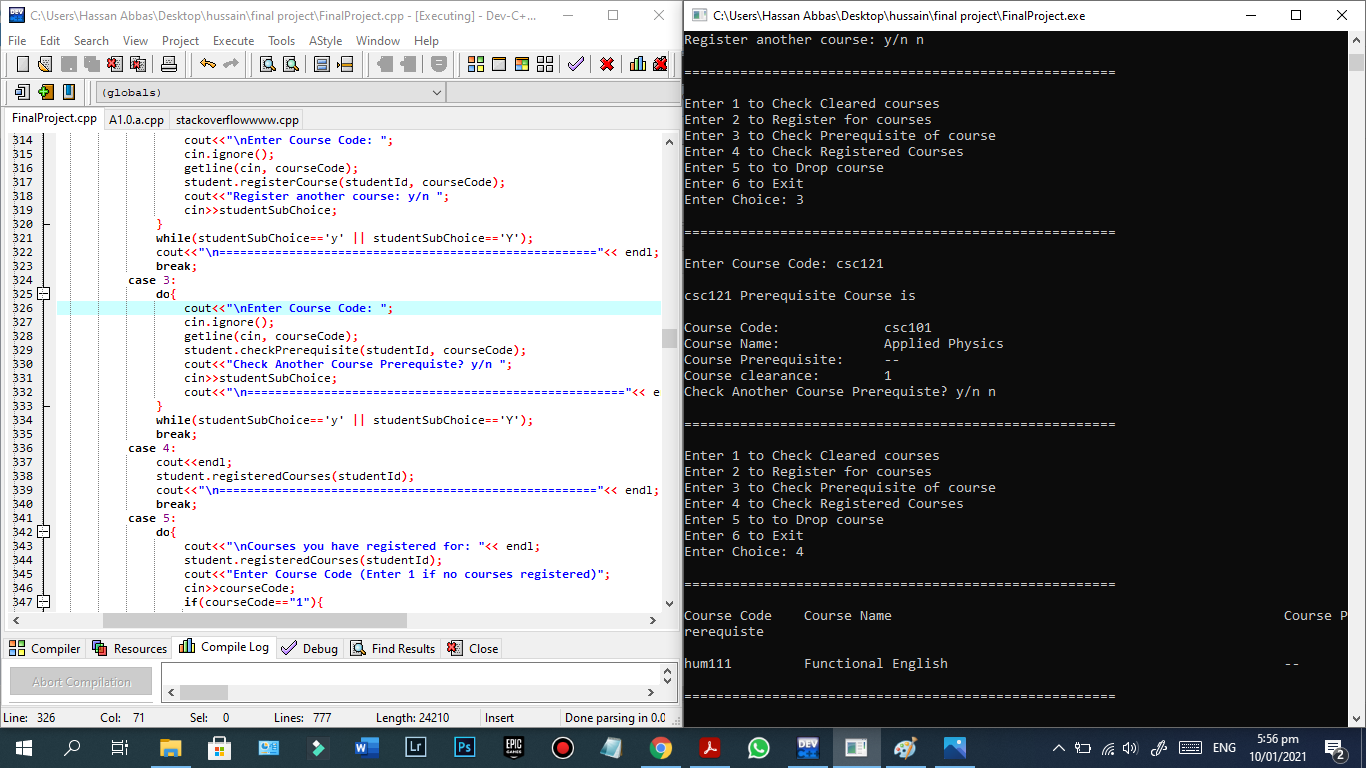
If the user is student, he can enter the student menu by entering option 2. On entering option 2 the student will be asked his id and findNode function will be called check if the student account exist and if it exists, he will be asked his password and his correctness will be checked by calling checkPassword function and he/she will be able to enter the student if he/she has entered correct id and password which are set by the admin for them. After that he/she can perform different operation with 6 different option.



On entering option 1 the student can check his/her cleared courses. checkCourses function will be called to display list of cleared courses.

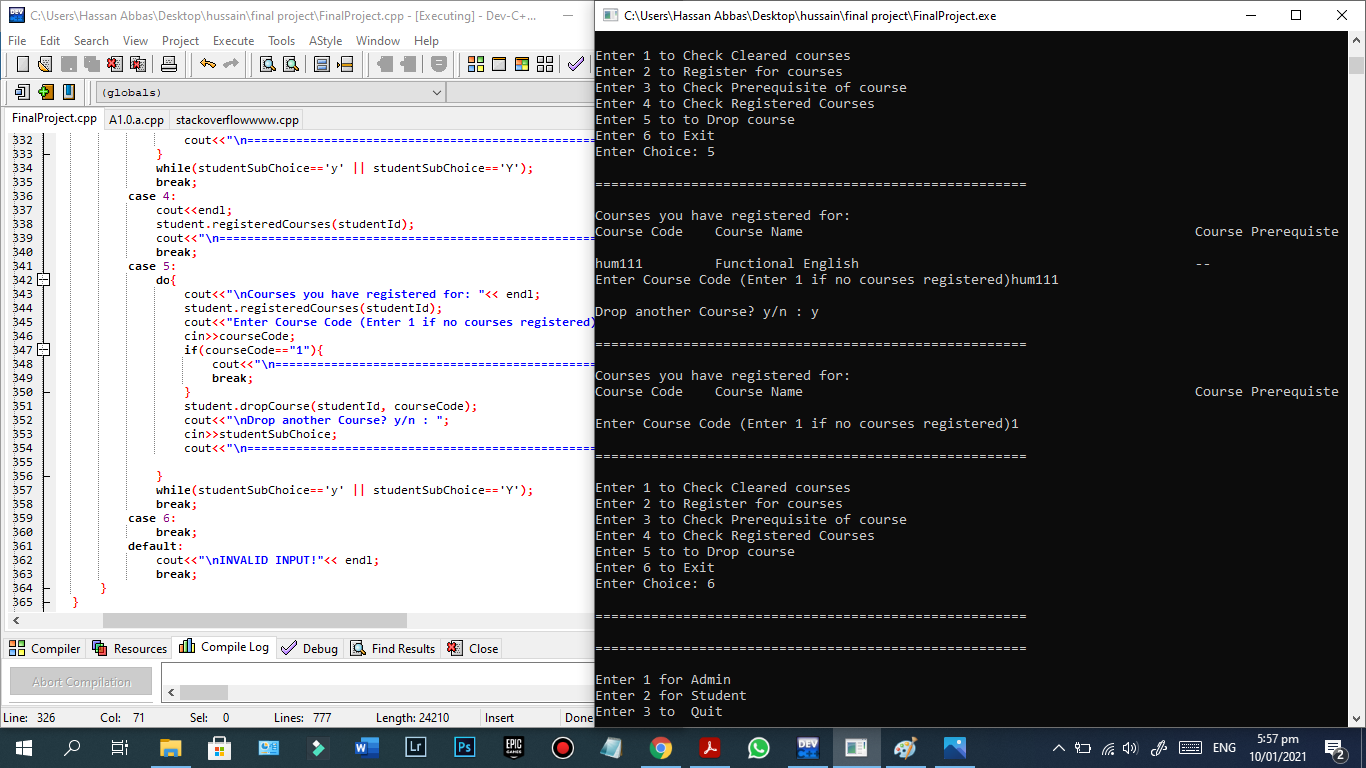


On entering option 2 the student can register for different courses available for him/her. availableCourses function is called for displaying the list of available courses. From the list student can enter any course code to register for that courses. registerCourse function will be called and the selection variable status will be changed to true upon registering courses. He/she can register for as many as available for him/her except the course of which the prerequisite course is not cleared. If the student tries to register for course of which the prerequisite course is not cleared he/she will get a message to clear its prerequisite course first by displaying that course name, code and prerequisite course if any.



On entering option 3 student can check prerequisite course of any course by entering the course code and the checkPrerequiste function will be called and it will display the prerequisite course if there is any.

On entering option 4 the student will be able to see his/her currently registered courses. registeredCourses function will be called which will display the display the currently registered courses.



On entering option 5 the student can drop out of any courses that he/she have registered for. registeredCourses will be called to display list of the currently registered courses (if any) and the student can enter any course code to drop out from that course. dropCourse function will be called to drop the courses. This function will set the selection variable value back to false indicating that it is not registered.

On last option, option 6 student will exit the student menu and will enter the main menu, from there another user can use the program (either admin or another student or same student) or can quit the program by entering the option 3. On option 3 the program will exit and a GOOD BYE message will be displayed.

**The end.**