Course Number: 420-P16-AS

Course Title: Structured Programming

Session: Autumn 2016

**final project**

**TITLE: Grade tracking application in c++**

**Due DATE: 2016/12/06**

**submitted**

**to**

**TEACHER: Quang Hoang Cao**

**By**

**HAURYLCHYK EDUARD/ 1632491**

Heading

[**I.** **Project description** 3](#_Toc468738951)

[1. A short description of the project 3](#_Toc468738952)

[**II.** **project development** 5](#_Toc468738953)

[Phase i Analysis and design 5](#_Toc468738954)

[Program structure 5](#_Toc468738955)

[The list of functions: 6](#_Toc468738956)

[Algorithms (Pseudocode/Flowcharts) 8](#_Toc468738957)

[The flowchart of the Main Function and Teachers’ menu. 8](#_Toc468738958)

[The flowchart of Students’ menu. 9](#_Toc468738959)

[Phase ii implementation 10](#_Toc468738960)

[1. Header File (s) 10](#_Toc468738961)

[2. Implementation File(s) 12](#_Toc468738962)

[3. Application File (The function main() must be in this file) 40](#_Toc468738963)

[phase iii testing the program 43](#_Toc468738964)

[**III.** **conclusion** 59](#_Toc468738965)

1. **Project description**
2. A short description of the project

The project is created in Microsoft Visual Studio.

Requirement: The teacher must login with the following valid user name and password:

Username: 5257

Password: quanghoang.

**The application allows the teacher to perform the following operations:**

1. List all the courses he is teaching

2. Enter students’ grades for a given course

3. Search a student’s grades by Student ID/First Name/Last

Name/First Name and Last Name

4. Sort the student list by Student ID

5. List all the students’ grades for a given course

6. Quit the application

**Teacher options:**

Option 1: List all the courses he is teaching

The information displayed includes Course Number, Course Title, Group

Number and Total Number of Students

Option 2: Enter a student’s grades

The teacher has to enter the following data:

- Student ID (7-digit number, unique value)

- First Name

- Last Name

- Project Grade

- Midterm Exam Grade

- Final Exam Grade

- Password

To keep it simple, username is Student ID.

Note: A grade entered must be between 0.0 and 100.0 included.

Option 3: Search a student’s grades by Student ID / Firstname/Lastname/

Firstname and Lastname (the example is given)

Option 4: Sort the student list by Student ID in ascending order

Option 5: List all the students’ grades (the example is given)

Option 6: Quit the application

The teacher selects this option to exit the application. Confirmation message

must be shown to the teacher.

**Student options:**

1. List all the courses he/she is taking

2. View the grade for a given evaluation component

(Midterm Exam, Final Project or final Exam) related to

a course he/she is taking in this session (Autumn 2016)

3. List all the grades for a given course

4. Quit the application

**Users and Operations**

|  |  |
| --- | --- |
| **User** | **Operations** |
| Teacher | 1. See the list of courses he is teaching. 2. Create the list of students for three groups of students. 3. Enter students’ grades for a given course. 4. Search the student by student ID, First Name, Last Name, First and Last name. 5. Sort the student list by Student ID in ascending order. 6. List all the students’ grades for a given course. 7. Quit the application. | |
| Student | 1. Log in to application using its’ ID and password, created by the teacher. 2. List all the courses he/she is taking 3. View the grade for a given evaluation component(Midterm Exam, Final Project or final Exam) related to a course he/she is taking in this session (Autumn 2016) 4. List all the grades for a given course 5. Quit the application | |

1. **project development**

## Phase i Analysis and design

### Program structure

The main structure of the program:

struct Student

{

string StudentId;

float Project\_Grade;

float Midterm\_exam\_grade;

float Final\_exam\_grade;

float Final\_Result;

string firstName;

string lastName;

string stud\_password;

string group\_number;

string course\_number;

string course\_description;

int StudentGroupQuantity;

};

The Structure Student involves all the data of the students, courses, groups, grades. Also the structure involves the calculation of final result.

The structure Student is formed, when user insert the data of students in the function inputStudentData(Student arrStudent[]).

### The list of functions:

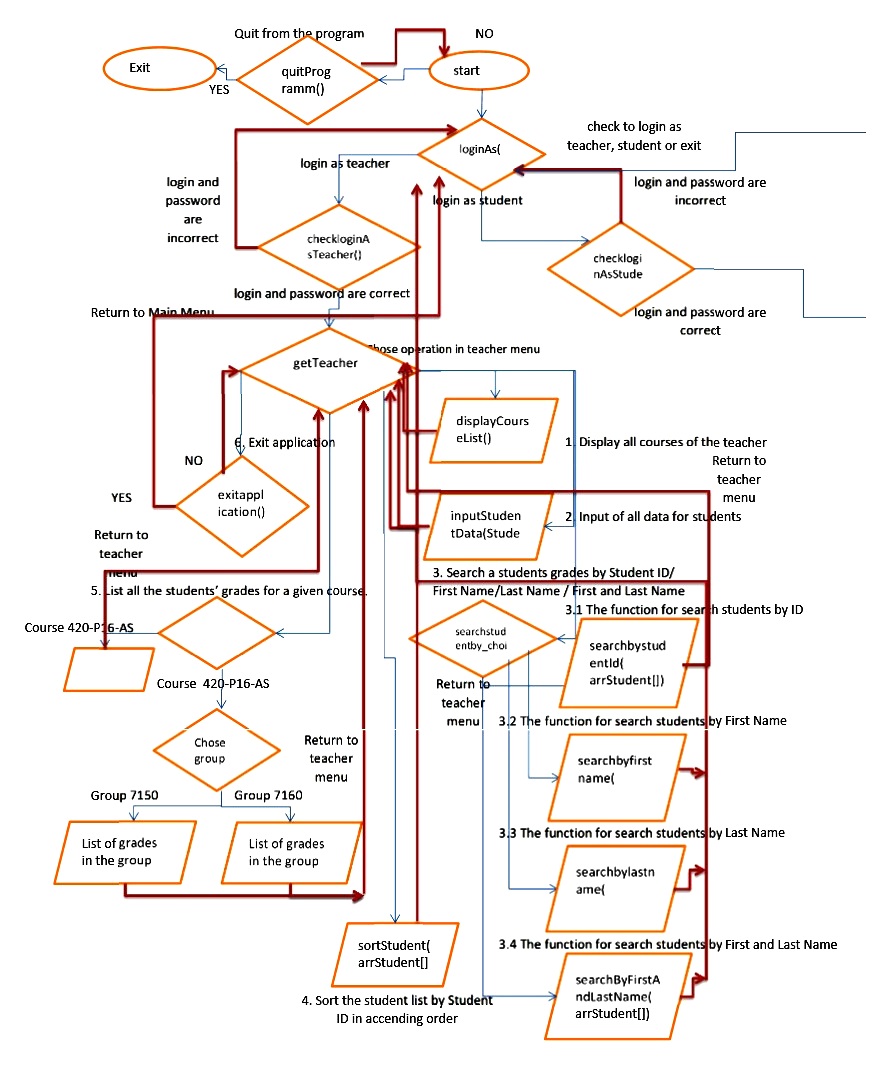
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Name of function | Header function | Menu | Short description |
| 1 | int loginAs(); | The Main menu function | Main menu | Choise to login as Teacher or as a Student |
| 2 | int checkloginAsStudent(Student arrStudent[]); | int loginAs() | Main menu | Check the login and password of the student |
| 3 | void checkloginAsTeacher(); | int loginAs() | Main menu | Check the login and password of the teacher. Returns whether login inserted is equal to login of the teacher. |
| 4 | int getTeacherMenuChoice(); | checkloginAsTeacher() | Teacher menu | Calls main menu call of the teacher, returns teachers choice. |
| 5 | void displayCourseList(); | getTeacherMenuChoice() | Teacher menu | Displays the List of all courses of the teacher |
| 6 | int inputStudentData(Student arrStudent[]); | getTeacherMenuChoice() | Teacher menu | The function, allowing to input the data of the students. Retuns total quantity of the students. |
| 7 | int chosecourse\_select(); | getTeacherMenuChoice() | Teacher menu | List all the students’ grades for a given course. Menu for choise of the course; Returns choice of the course; |
| 8 | void displayGrades(Student arrStudent[]); | getTeacherMenuChoice() | Teacher menu | displays grades for all the student, in dependance of the course and the group; |
| 9 | void sortStudent(Student arrStudent[]); | getTeacherMenuChoice() | Teacher menu | Sort the student list by Student ID in accending order |
| 10 | int searchstudentby\_choise(); | getTeacherMenuChoice() | Teacher menu | Menu for Search a students grades by Student ID/First Name/Last Name / First Name and Last Name"; Returns chosen criteria for search; |
| 11 | void searchstudentby(Student arrStudent[]); | searchstudentby\_choise() | Teacher menu | The swich-case function for search of the student by different parameters |
| 12 | void searchbystudentId(Student arrStudent[]); | searchstudentby\_choise() | Teacher menu | The function for search students by ID |
| 13 | void searchbyfirstname(Student arrStudent[]); | searchstudentby\_choise() | Teacher menu | The function for search students by First name |
| 14 | void searchbylastname(Student arrStudent[]); | searchstudentby\_choise() | Teacher menu | The function for search students by Last Name |
| 15 | void searchByFirstAndLastName(Student arrStudent[]); | searchstudentby\_choise() | Teacher menu | The function for search students by First and Last Name |
| 16 | bool quitApplication(); | getTeacherMenuChoice() and studentmenu\_choise() | Teacher and Student menu | log out from of the application |
| 17 | bool quitProgramm(); | The Main menu function | Main menu | The Main menu function, choice whether to exit the programm; |
| 18 | bool isValidId(string, int); | inputStudentData(arrStudent) | Teacher menu | Check whether student ID consists from 7 digits; |
| 19 | bool NumberBetween0and100(float number); | inputStudentData(arrStudent) | Teacher menu | Check whether the grades are between 0 and 100; |
| 20 | bool uniqueStudId(Student arrStudent[], int, string); | inputStudentData(arrStudent) | Teacher menu | Check whether student ID is unique; |
| 21 | float isACorrectFloat(string); | inputStudentData(arrStudent) | Teacher menu | Check whether the grades are correct floats; |
| 22 | void stud\_evaluation\_grade(Student arrStudent[], int, string); | student\_menu( arrStudent, int, string, string) | Student Menu | View the grade for a given evaluation component (Midterm Exam, Final Project or final Exam) related to a course he / she is taking in this session(Autumn 2016) |
| 23 | void student\_menu(Student arrStudent[], int, string, string); | checkloginAsStudent( arrStudent) | Student Menu | Use switch - case function ti call the functions of student menu. |
| 24 | bool isValidgroup\_quant(string tempgroup); | inputStudentData(arrStudent) | Teacher menu | Check whether the inserted number of students in the group is integer |
| 25 | int studentmenu\_choise(); | student\_menu( arrStudent, int, string, string) | Student Menu | The main menu for the student; |
| 26 | int stud\_eval\_component\_choise(); | stud\_evaluation\_grade(arrStudent, int, string) | Student Menu | Choise of the evaluation component of students |

## Algorithms (Pseudocode/Flowcharts)

### The flowchart of the Main Function and Teachers’ menu.

The return to the previous menu is marked with red arrow

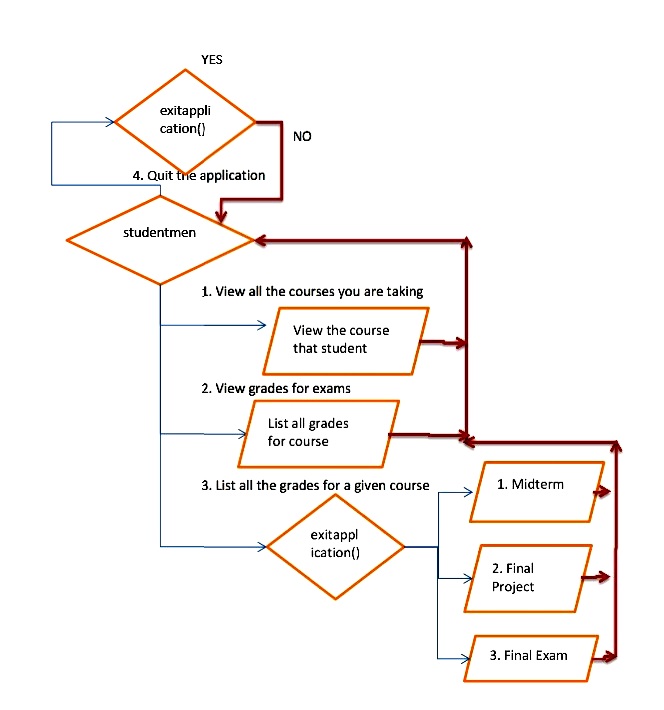
The direction of logic flow is marked with blue arrow



### The flowchart of Students’ menu.

The return to the previous menu is marked with red arrow

The direction of logic flow is marked with blue arrow

****

# Phase ii implementation

The focus of this phase is on the HOWs.

Included in this phase are the source code files of your program.

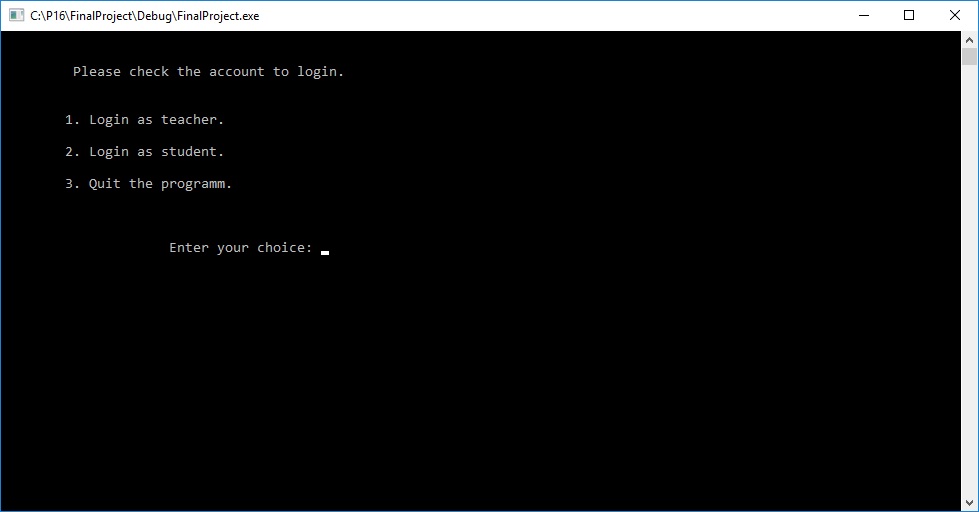
1. Header File (s)
2. #pragma once
3. #ifndef Student
4. #include <regex> //used for function isACorrectFloat(string)
5. #include<iostream>
6. #include<string>
7. #include <stdlib.h>
8. #include<iomanip>
9. #include<math.h>
10. #define COURSE\_QUANT 3 //Quantity of all the groups in the courses
11. #define NUM\_STUDENT 20 //The average number of students in the group
12. using namespace std;
13. // The Structure Student involves all the data students, courses, groups, grades.
14. struct Student
15. {
16. string StudentId; //The ID of students
17. float Project\_Grade; //Progect Grades of students
18. float Midterm\_exam\_grade; // Midterm exam grades of students
19. float Final\_exam\_grade;// Final exam grades of students
20. float Final\_Result; // Final result of students, is calculated during data entry
21. string firstName; // The first name of the student
22. string lastName; // The last name of the student
23. string stud\_password; //Student password
24. string group\_number; //Group number of the student, assigned to every student during data entry
25. string course\_number; //Course number of the student, assigned to every student during data entry
26. string course\_description; //Course description, assigned to every student during data entry
27. int StudentGroupQuantity; //the number of student in each group
28. };
29. int loginAs(); // The Main menu function, Choise to login as Teacher or as a Student
30. int checkloginAsStudent(Student arrStudent[]);//Main menu - int loginAs().Check the login and password of the student
31. void checkloginAsTeacher(); //Main menu - int loginAs(). Check the login and password of the teacher. Returns whether login inserted is equal to login of the teacher.
32. int getTeacherMenuChoice(); //Teacher menu - checkloginAsTeacher(). Calls main menu call of the teacher, returns teachers choice.
33. void displayCourseList();//1. Teacher Menu - getTeacherMenuChoice();. Displays the List of all courses of the teacher
34. int inputStudentData(Student arrStudent[]); //2. Teacher Menu - getTeacherMenuChoice(); The function, allowing to input the data of the students. Retuns total quantity of the students.
35. int chosecourse\_select();// 5. Teacher Menu - getTeacherMenuChoice() - displayGrades(arrStudent); List all the students’ grades for a given course. Menu for choise of the course; Returns choice of the course;
36. void displayGrades(Student arrStudent[]);//Teacher Menu - getTeacherMenuChoice(); displays grades for all the student, in dependance of the course and the group;
37. void sortStudent(Student arrStudent[]);//Teacher Menu - getTeacherMenuChoice(); Sort the student list by Student ID in accending order
38. int searchstudentby\_choise(); // 3.Teacher Menu - getTeacherMenuChoice(); Menu for Search a students grades by Student ID/First Name/Last Name / First Name and Last Name"; Returns chosen criteria for search;
39. void searchstudentby(Student arrStudent[]); //getTeacherMenuChoice()-searchstudentby\_choise() - The swich-case function for search of the student by different parameters
40. void searchbystudentId(Student arrStudent[]); //getTeacherMenuChoice()-searchstudentby\_choise()-searchstudentby() - The function for search students by ID
41. void searchbyfirstname(Student arrStudent[]);//getTeacherMenuChoice()-searchstudentby\_choise()-searchstudentby() - The function for search students by First name
42. void searchbylastname(Student arrStudent[]); //getTeacherMenuChoice()-searchstudentby\_choise()-searchstudentby() - The function for search students by Last Name
43. void searchByFirstAndLastName(Student arrStudent[]); //getTeacherMenuChoice()-searchstudentby\_choise()-searchstudentby() - The function for search students by First and Last Name
44. bool quitApplication(); //3. Teacher Menu - getTeacherMenuChoice(); And Student Menu int studentmenu\_choise() - log out from of the application
45. bool quitProgramm(); //The Main menu function, choice whether to exit the programm;
46. bool isValidId(string, int);//Teacher Menu -inputStudentData(arrStudent).Check whether student ID consists from 7 digits;
47. bool NumberBetween0and100(float number);//Teacher Menu -inputStudentData(arrStudent).Check whether the grades are between 0 and 100;
48. bool uniqueStudId(Student arrStudent[], int, string);//Teacher Menu -inputStudentData(arrStudent).Check whether student ID is unique;
49. float isACorrectFloat(string);//Teacher Menu -inputStudentData(arrStudent).Check whether the grades are correct floats;
50. void stud\_evaluation\_grade(Student arrStudent[], int, string); //Student Menu -student\_menu(arrStudent, int, string, string). View the grade for a given
51. //evaluation component (Midterm Exam, Final Project or final Exam) related to a course he / she is taking in this session(Autumn 2016)
52. void student\_menu(Student arrStudent[], int, string, string); //Student Menu - checkloginAsStudent( arrStudent) - Use switch - case function ti call the functions of student menu.
53. bool isValidgroup\_quant(string tempgroup); //Teacher Menu -inputStudentData(arrStudent).Check whether the inserted number of students in the group is integer
54. int studentmenu\_choise(); //Student Menu - student\_menu( arrStudent, int, string, string).The main menu for the student;
55. int stud\_eval\_component\_choise();// stud\_evaluation\_grade(arrStudent, int, string). Choise of the evaluation component of students
56. //Constant values of the exams in the final result, is used in the evaluation of the final grade;
57. const float MidtermExamWeight = 0.3f; //The constant value of the midterm
58. const float ProjectWeight = 0.3f; //The constant value of the Project
59. const float FinalExamWeight = 0.4f;//The constant value of the final exam
60. //List of phrases, asking user, whether user wants to return to the previous menu, or choice
61. const string BACK\_TO\_Main\_MENU = "\n\n\tPress ENTER to return to the Main Menu.";
62. const string BACK\_TO\_Teacher\_MENU = "\n\n\tPress ENTER to return to the teacher Menu.";
63. const string BACK\_TO\_Student\_MENU = "\n\n\tPress ENTER to return to the student Menu.";
64. const string BACK\_Stud\_Exam\_Choice = "\n\n\tPress ENTER to return to the choice of student exam.";
65. const string BACK\_TO\_Stud\_Search\_MENU = "\n\n\tPress ENTER to return to the student search choice.";
66. //Strings for displaying the number of students, who passed or failed the course.
67. const string string\_student\_passed = "\n\n\tThe number of students, who passed the course: ";
68. const string string\_student\_failed = "\n\n\tThe number of students, who failed the course: ";
69. //List of phrases, the header of the output for the search student funstion, and for the search of the course grade function.
70. const string COURSE0 = "ASSESSMENT SUMMARY";
71. const string COURSE1 = "Course Number:";
72. const string COURSE2 = "Course Title:";
73. const string COURSE3 = "Teacher: Quang Hoang Cao";
74. const string COURSE4 = "Session: Autumn 2016";
75. const string COURSE5 = "Group:";
76. #endif;
77. Implementation File(s)

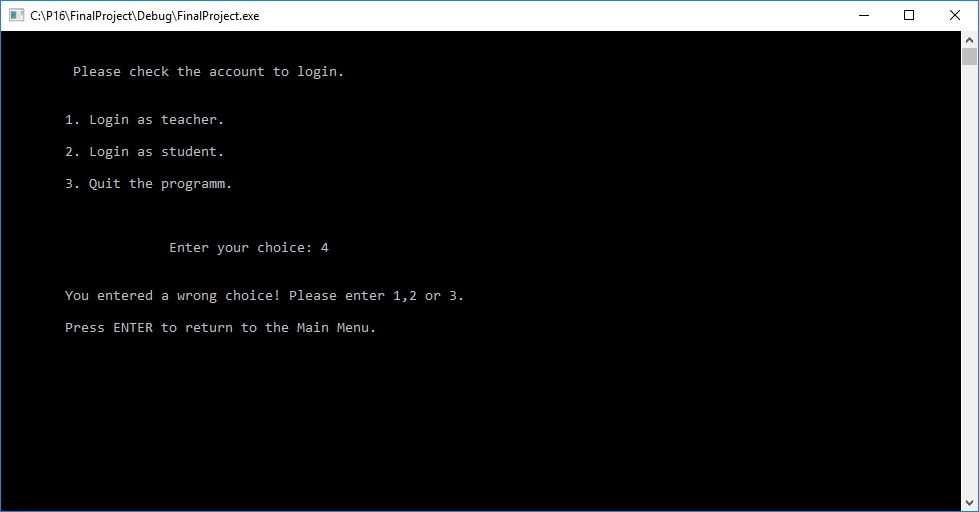
……………………………………………………..

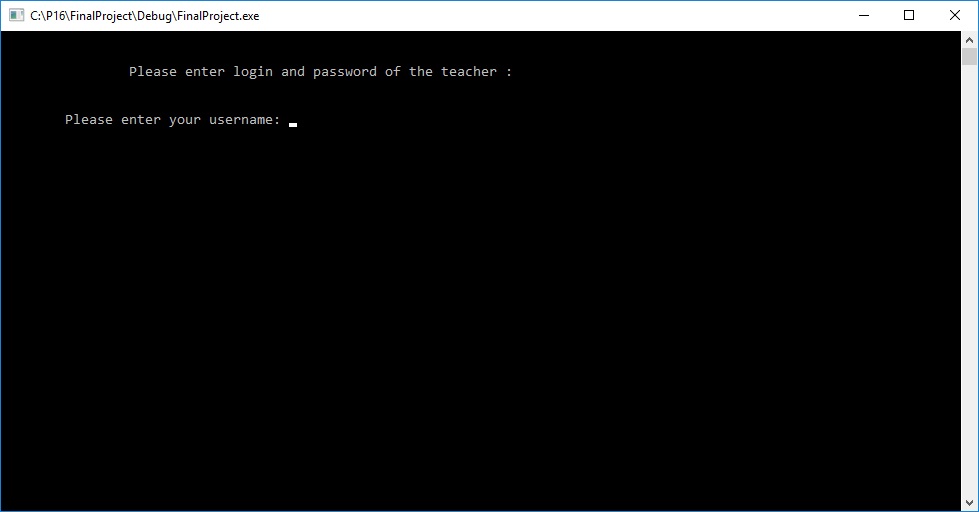
1. Application File (The function main() must be in this file)
2. #include "FinalProject.h";
3. Student arrStudent[NUM\_STUDENT\*COURSE\_QUANT]; //Structure Student
4. bool exitApp = true;
5. bool exitProgr = true;
6. int main()
7. {
8. int loginAs();
9. do
10. {
11. system("cls");
12. int choice2 = loginAs(); // The Main menu function, Choise to login as Teacher or as a Student
13. switch (choice2)
14. {
15. case 1:
16. system("cls");
17. checkloginAsTeacher(); // The Main menu function, checking login of Teacher for correctness
18. {
19. do
20. {
21. system("cls");
22. int choice = getTeacherMenuChoice(); //The menu of teacher
23. switch (choice)
24. {
25. case 1:
26. system("cls");
27. displayCourseList(); // List all the courses that you are teaching";
28. exitApp = false;
29. system("pause>>null");
30. break;
31. case 2:
32. // 2. Enter students’ grades for a given course";
33. system("cls");
34. inputStudentData(arrStudent); //Input of student data with validation
35. exitApp = false;
36. break;
37. case 3:
38. // 3. Search a student’s grades by Student ID/First Name/Last Name / First Name and Last Name";
39. system("cls");
40. searchstudentby(arrStudent); // Searching students by different parameters
41. exitApp = false;
42. break;
43. case 4:
44. // 4. Sort the student list by Student ID ";
45. system("cls");
46. sortStudent(arrStudent); // The function for sorting students
47. exitApp = false;
48. system("pause>>null");
49. break;
50. case 5:
51. // 5. List all the students’ grades for a given course";
52. system("cls");
53. displayGrades(arrStudent);
54. exitApp = false;
55. system("pause>>null");
56. break;
57. case 6:
58. // 6.Quit the application";
59. system("cls");
60. exitApp = quitApplication(); //Quit the application
61. break;
62. default:
63. cout << "\n\n\tYou entered a wrong choice! Please enter 1,2,3,4,5 or 6.";
64. cout << BACK\_TO\_Teacher\_MENU; //Return to teacher menu
65. system("pause>>nul");
66. exitApp = false;
67. break;
68. }
69. } while (!exitApp);
70. exitProgr = false;
71. break;
72. }
73. case 2:
74. system("cls");
75. checkloginAsStudent(arrStudent); //Check the login and password of the student
76. {
77. exitProgr = false;
78. break;
79. }
80. case 3:
81. system("cls");
82. exitProgr = quitProgramm(); //Quit if the program function
83. break;
84. default:
85. cout << "\n\n\tYou entered a wrong choice! Please enter 1,2 or 3.";
86. cout << BACK\_TO\_Main\_MENU;
87. system("pause>>nul");
88. exitProgr = false;
89. break;
90. }
91. } while (!exitProgr);
92. cout << "\n\tPress any key to exit the program..";
93. system("pause>>null");
94. return EXIT\_SUCCESS;
95. }

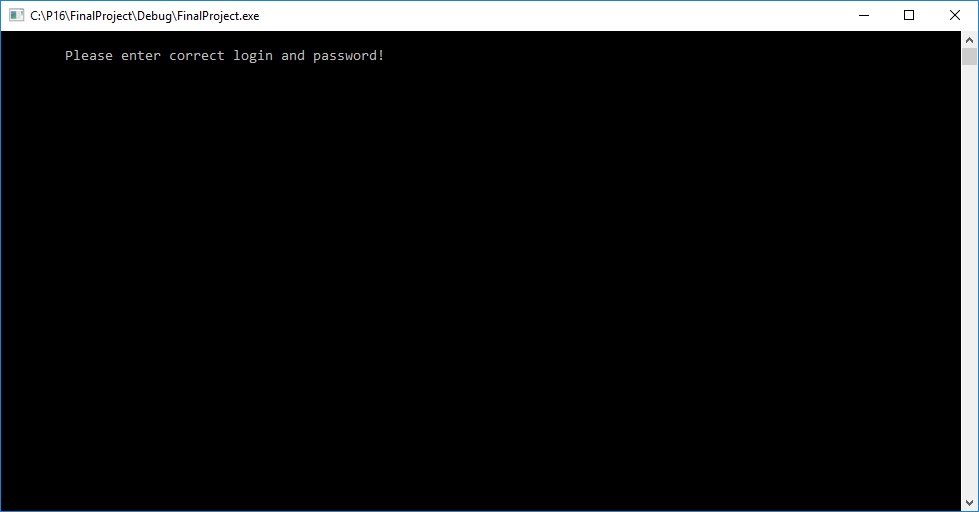
# phase iii testing the program

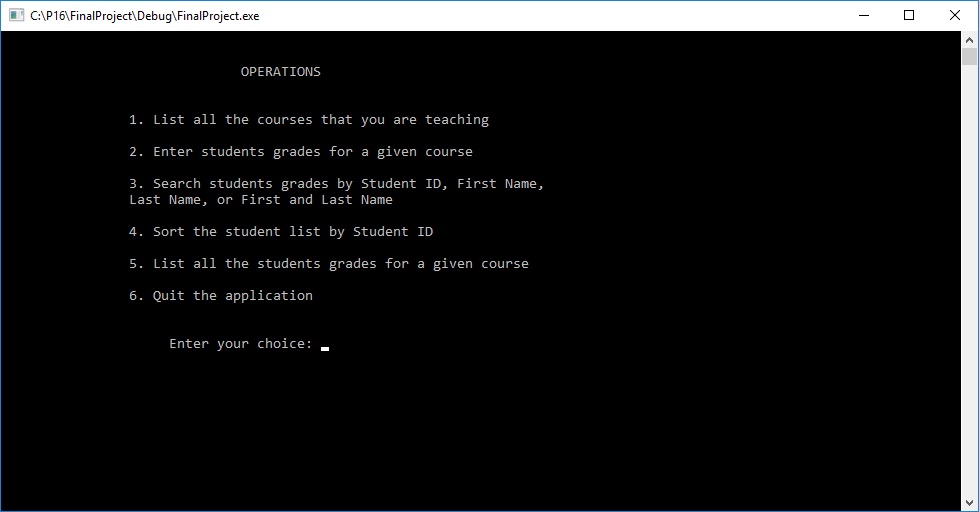
Included in this phase are the test results (in the table format) of your program.

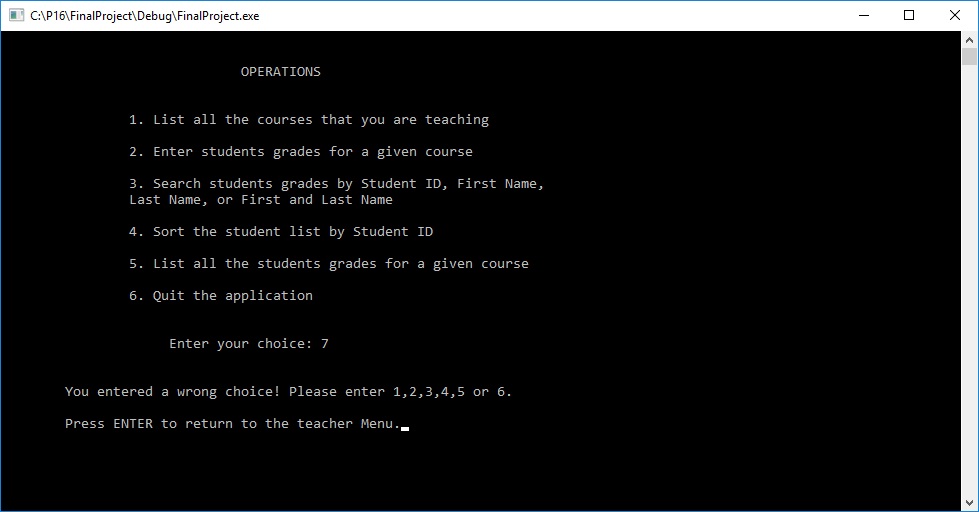
Step1. Main Menu choice

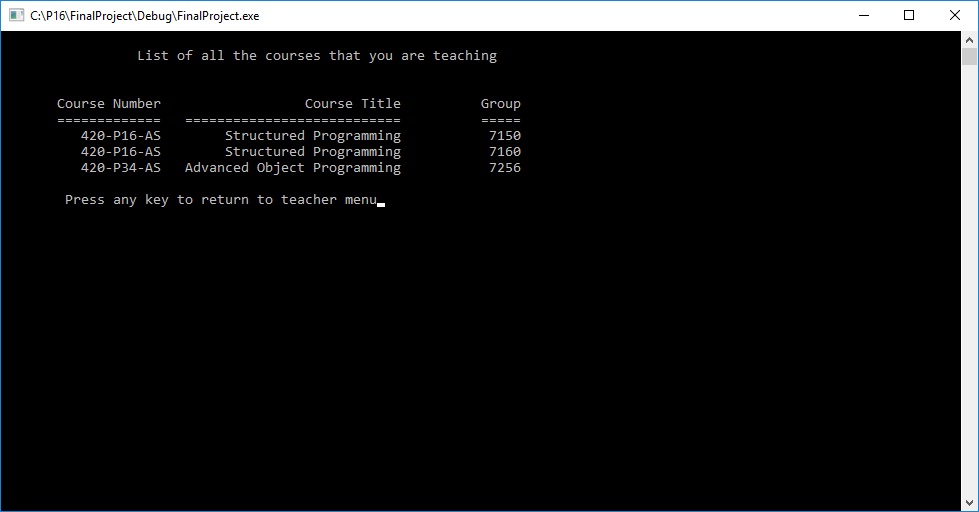
Step2. Check if mistake, Main Menu

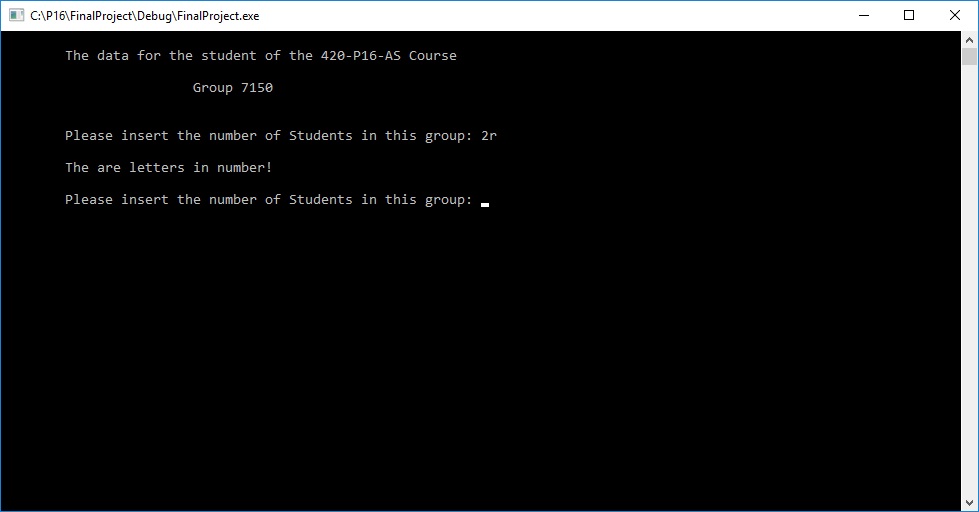
Step3. Login as teacher.

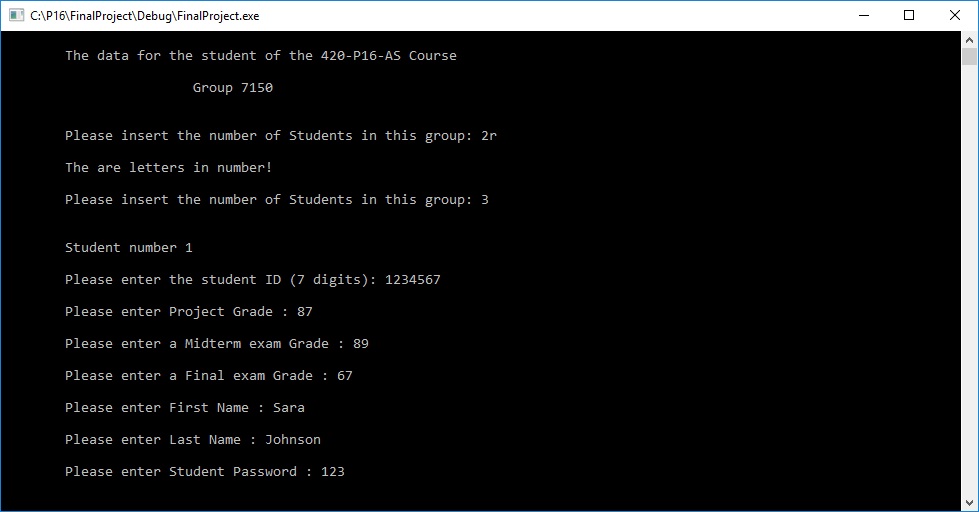
Step4. Check if login as teacher is wrong.

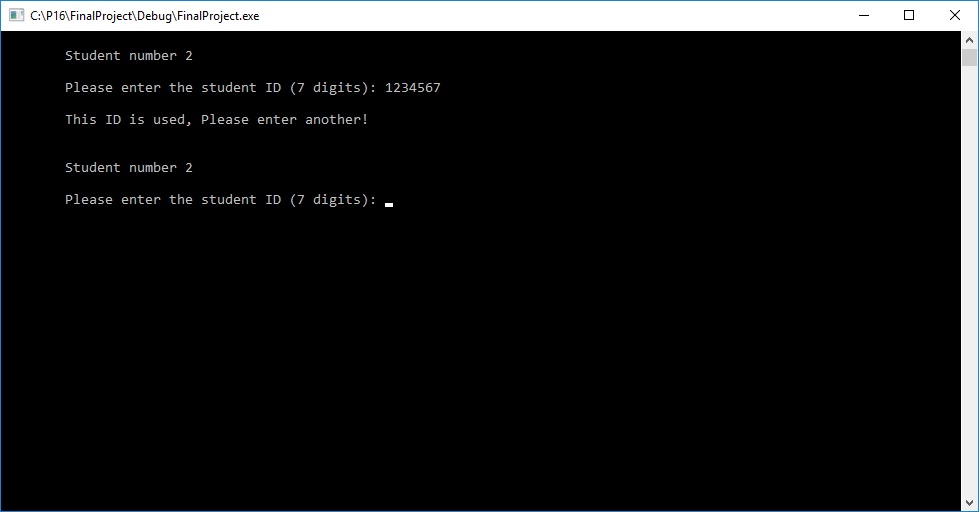
Step5. Teacher’s menu.

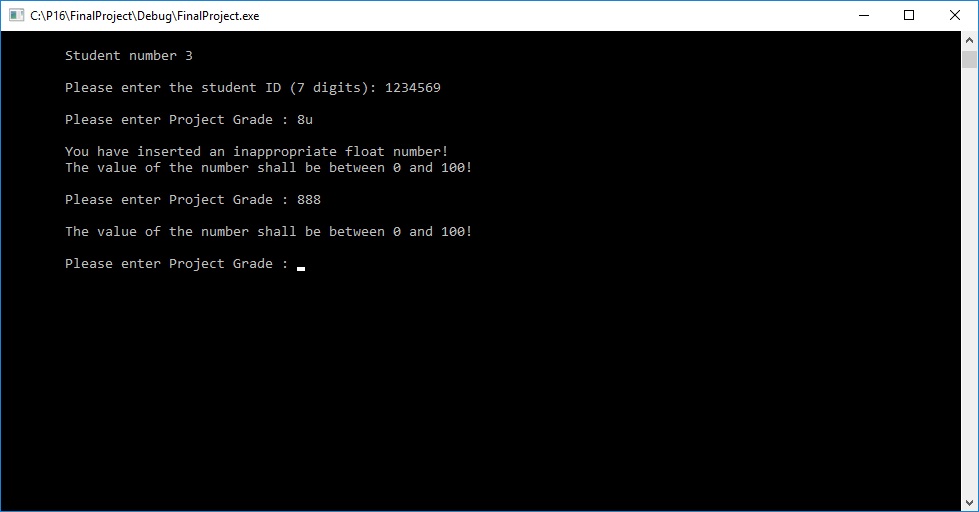
Step6. The choice in teachers’ menu is wrong.

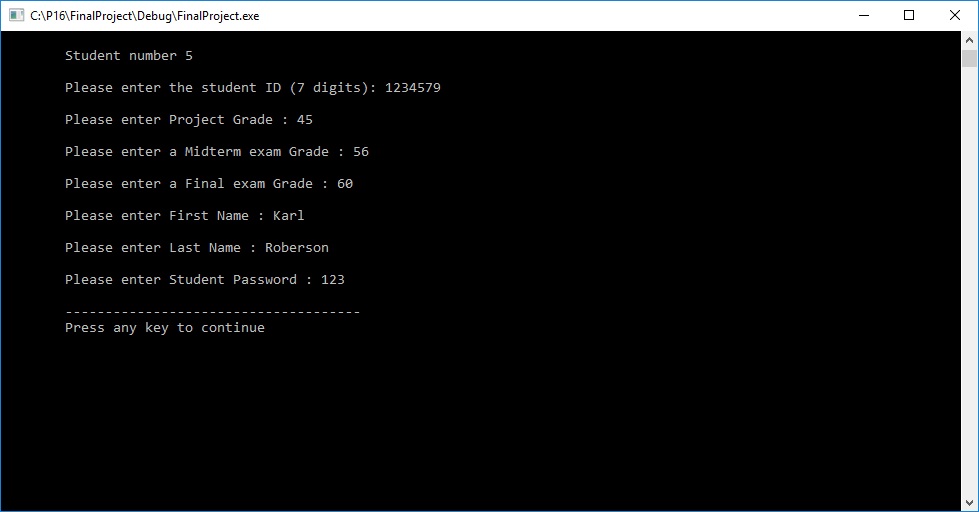
Step7.The list of all the courses that teacher teaching. 

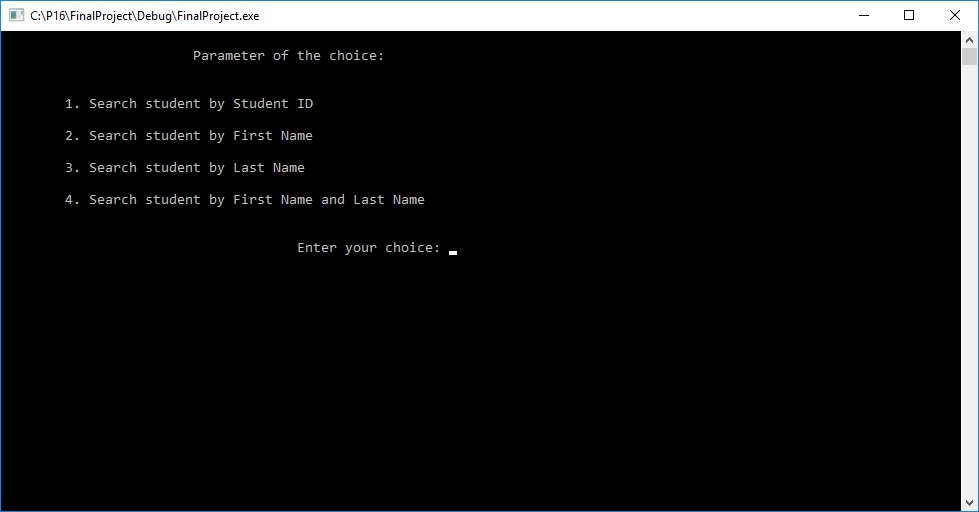
Step8.The mistake in typing of quantity of students in the group. 

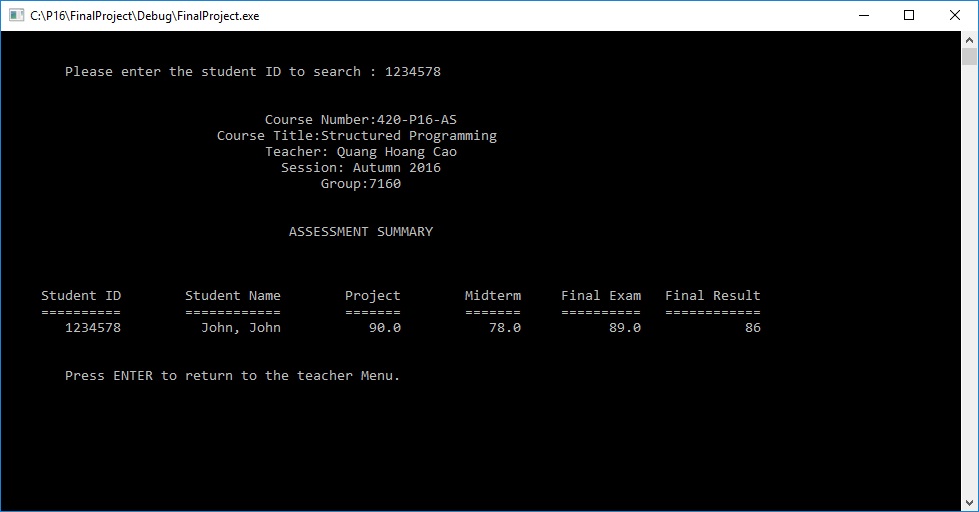
Step9. Insert of student data 

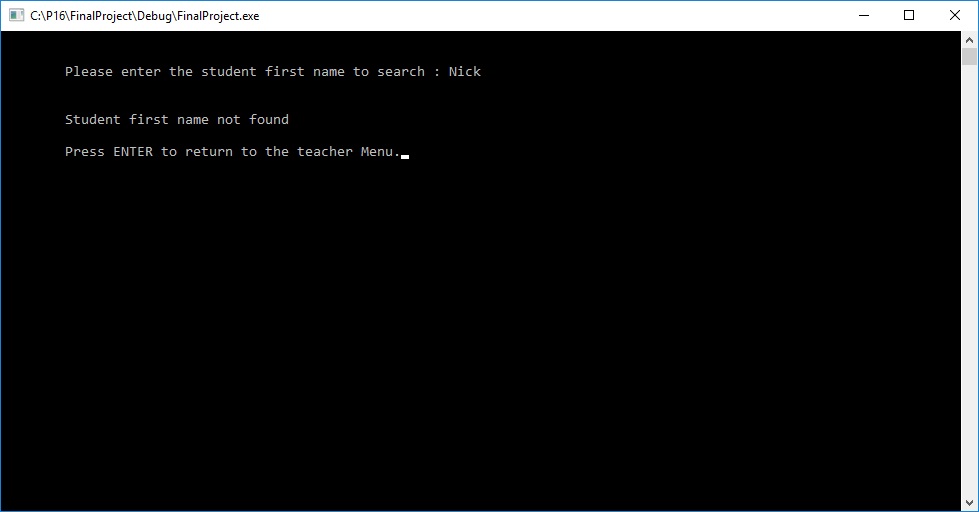
Step10.The duplicating of student ID

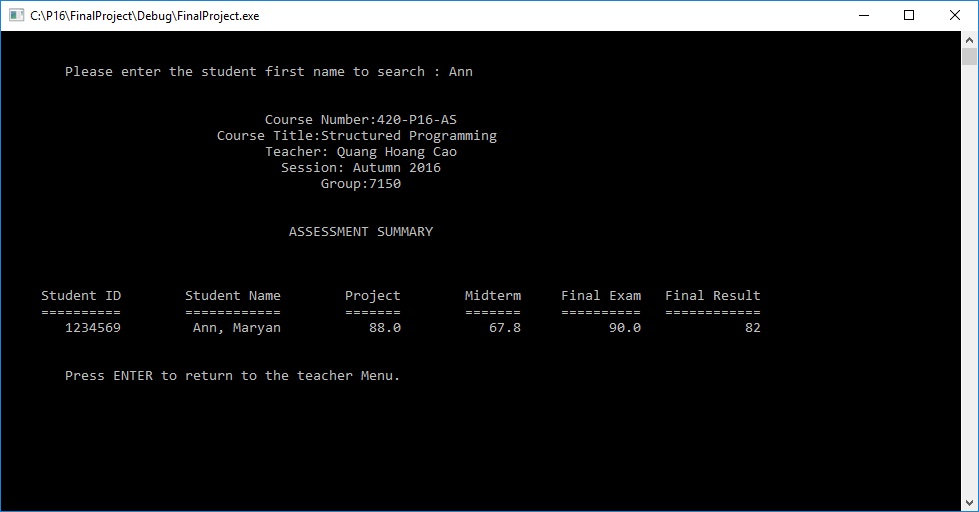
Step11.The mistake in writing project grade 

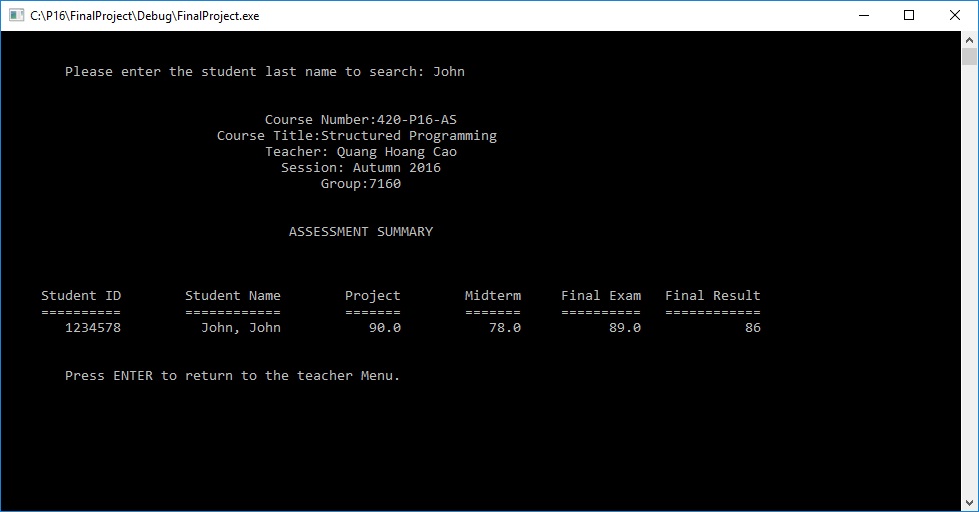
Step12.At the end of inserting the students the program asks for continue 

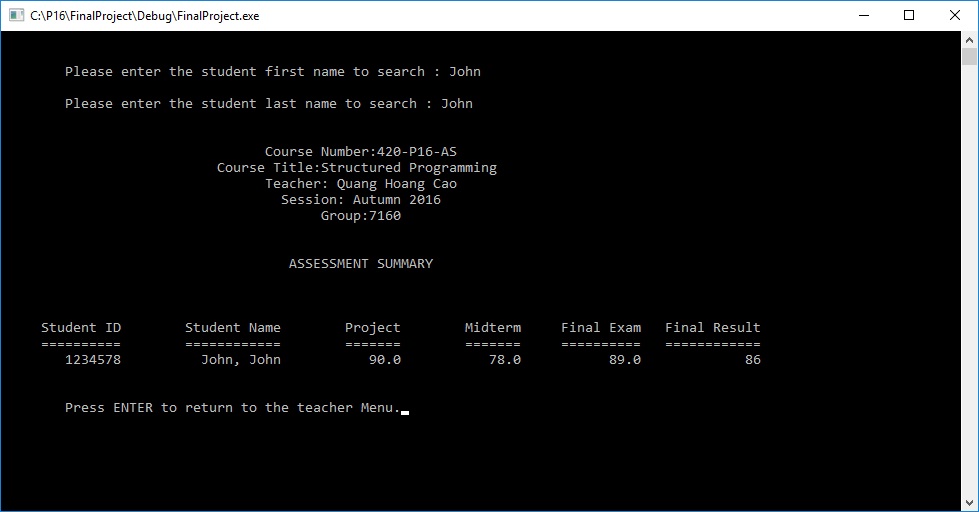
Step13.Menu of choice of the students by different parameters. 

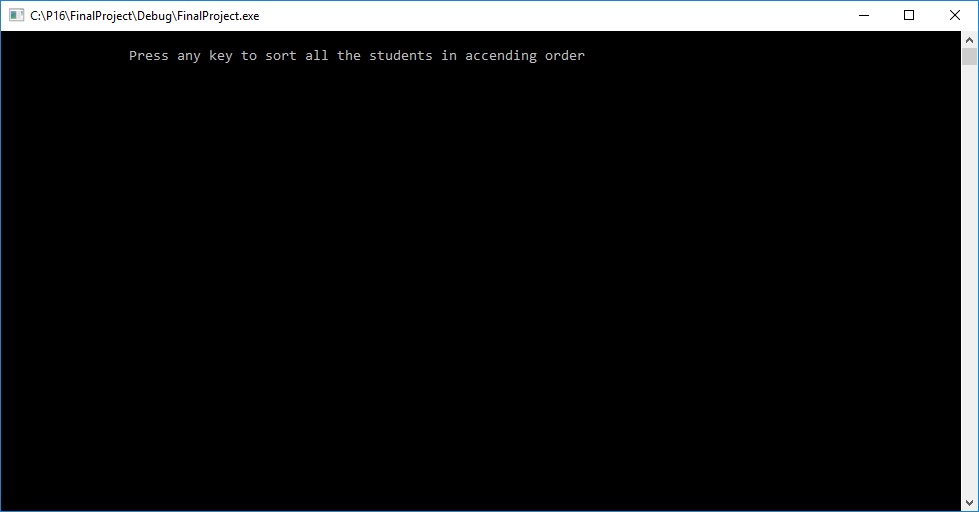
Step14.Result of search by student ID

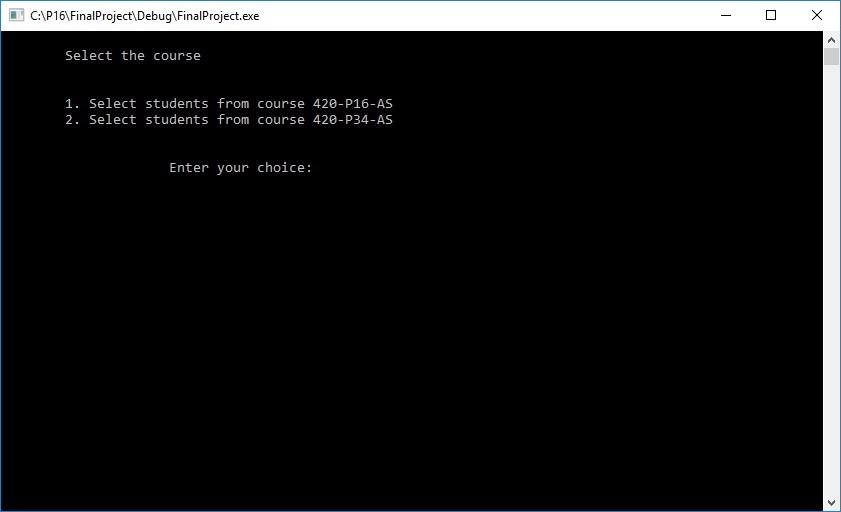
Step15.Mistake message if Name of student is not found

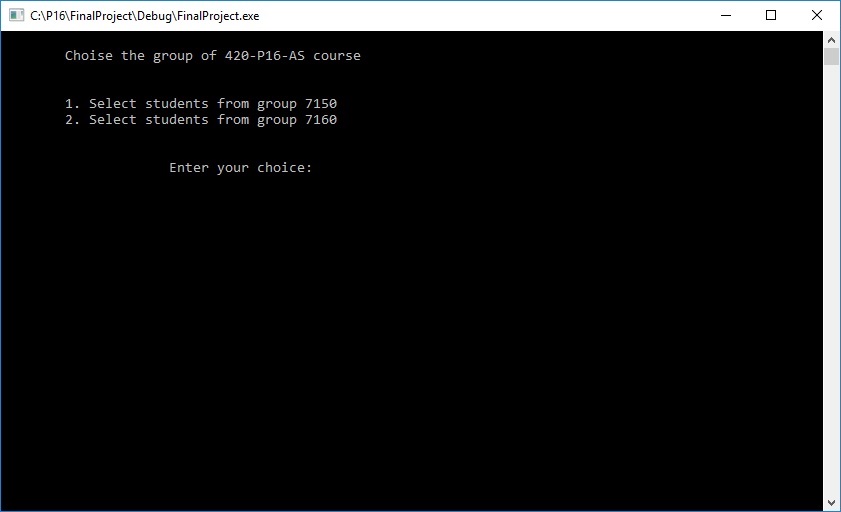
Step16.Result of search by Name of student

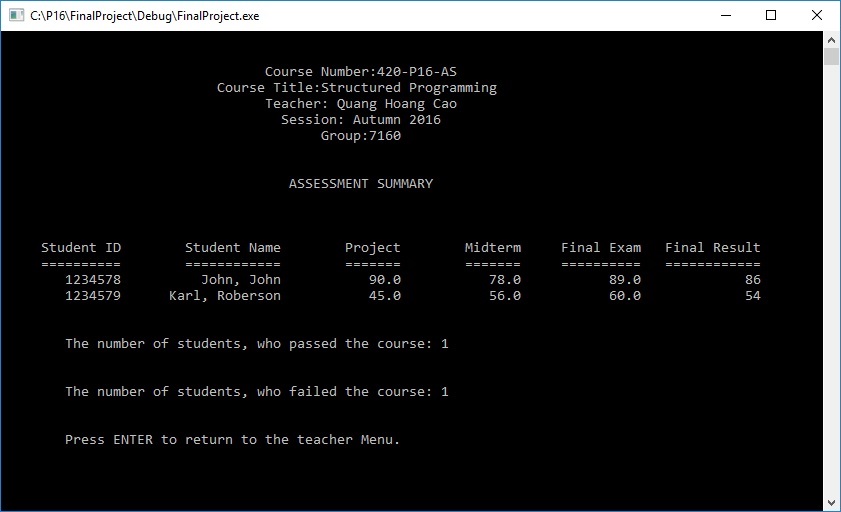
Step17.Result of search by last name 

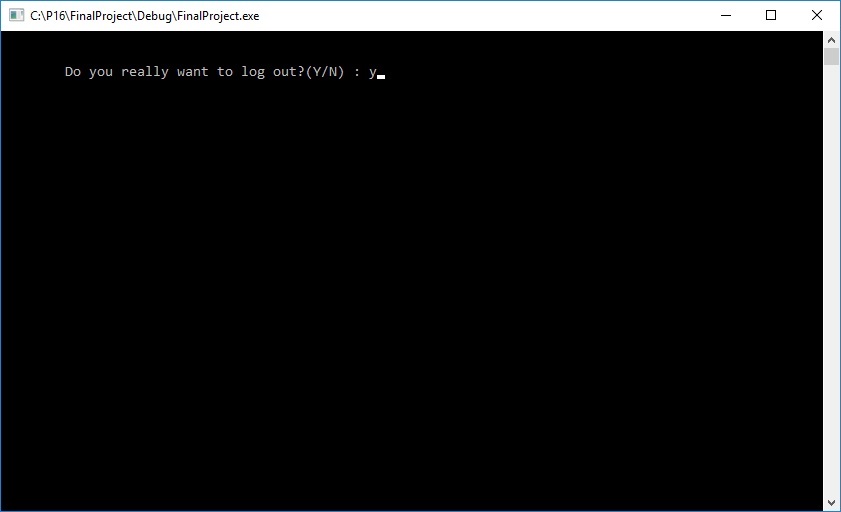
Step18.Result of search by first and last name

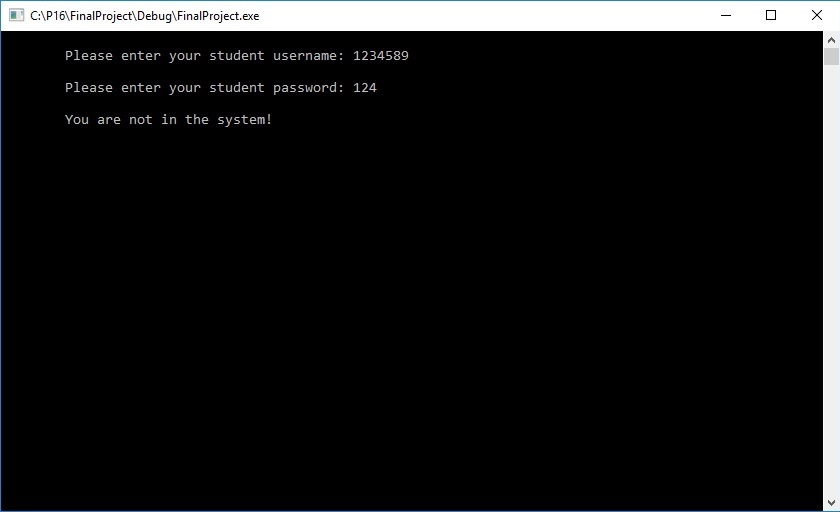
Step19.Sorting function, asks to press any key

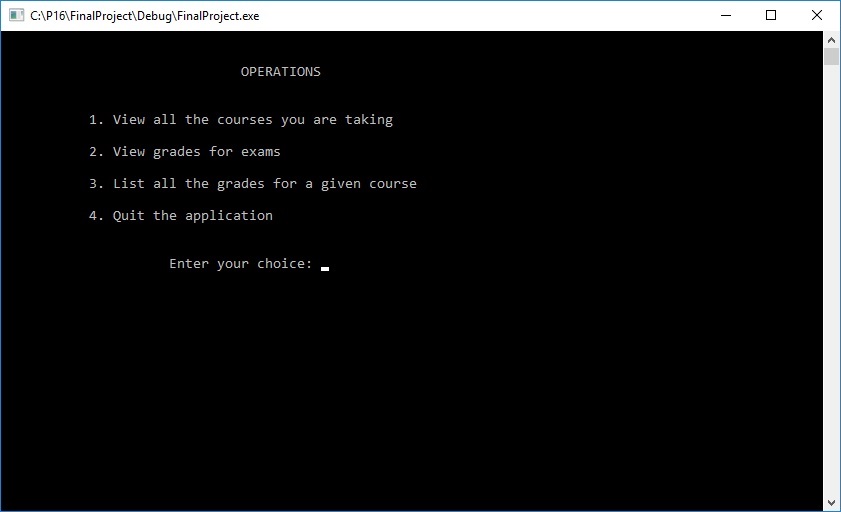
Step20.Search by course, selection between two courses

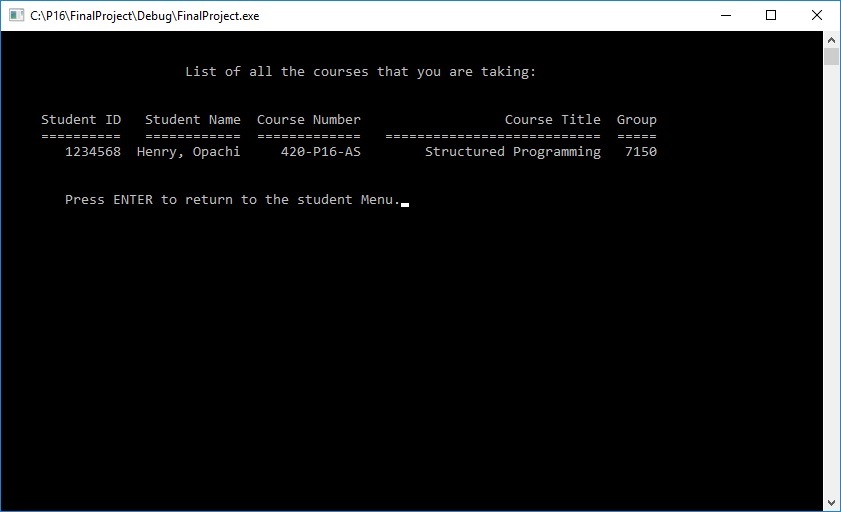
Step21.Search by group in course 420-P16-AS

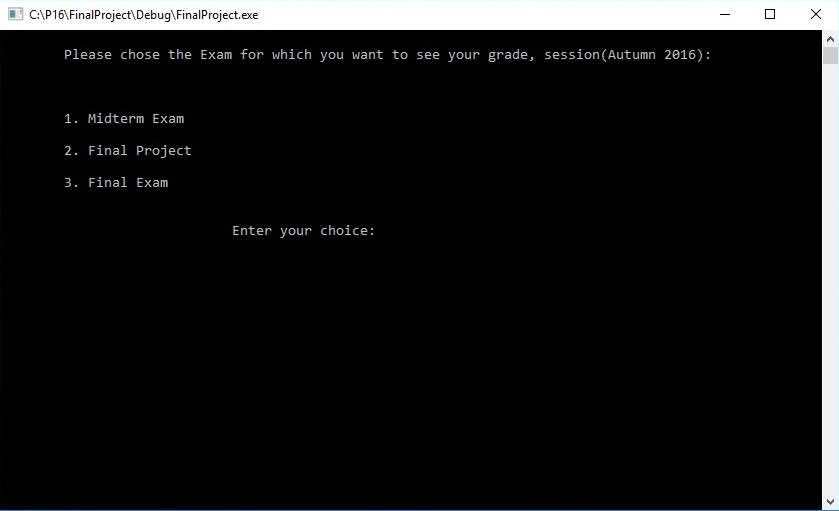
Step22.Result in search by course-group 

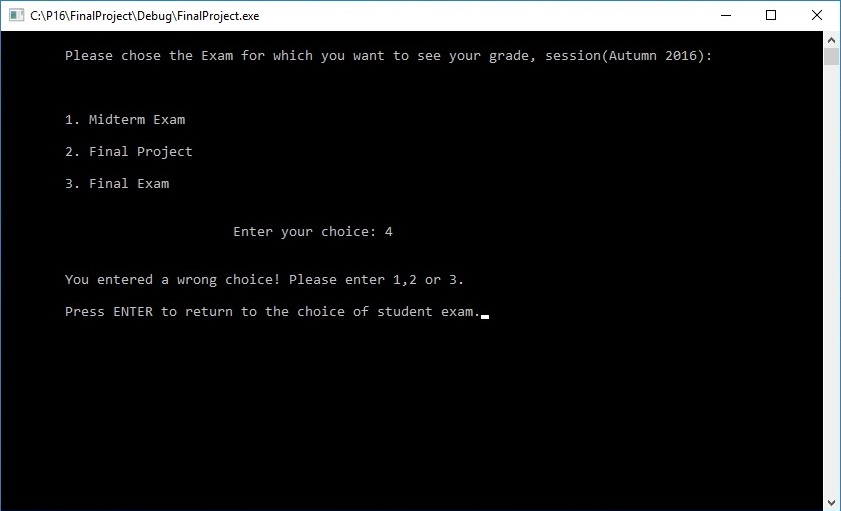
Step23.Log out of the teacher application

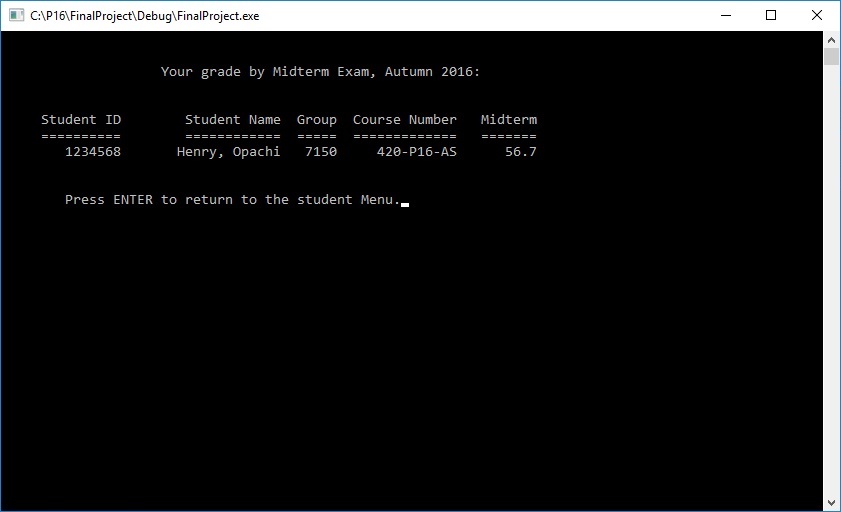
Step24.Login as student, if mistake

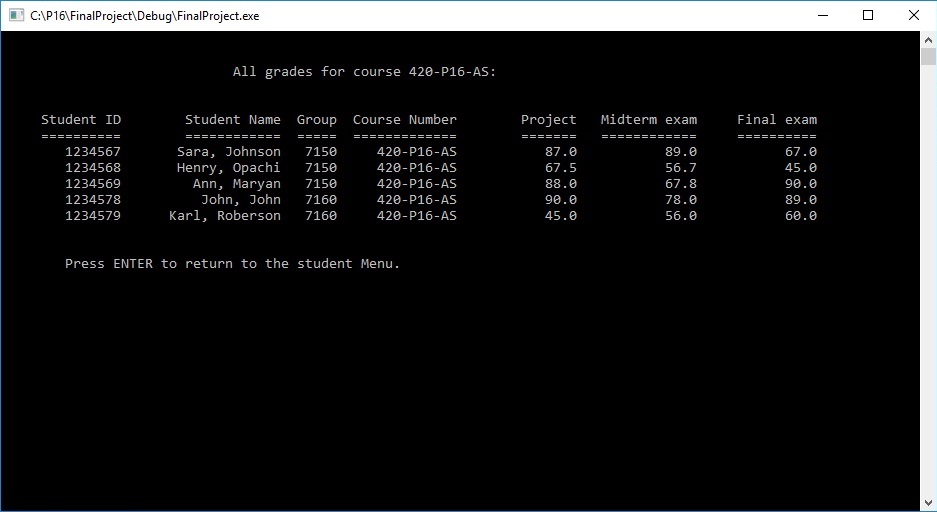
Step25.Student menu

Step26.List of courses of Student

Step27.Choice of exam, by which grade shall be selected. 

Step28.Mistake message, if choice of grade is wrong

Step29.The result, displaying to the student. The output is similar for all the exams

Step30.Displaying of all grades of the course****

Step31.Exit from the program 