



STUDENT INFORMATION SYSTEM

PROG 2100 FINAL PROJECT



NOVEMBER 5, 2024

PETER VAUGHAN
NSS E-Campus

Contents

PROG 2100 Final Project: Student Information System.....	2
Due Date	2
Assignment Instructions	2
<i>Evaluation</i>	2
Final Project: Student Information System	2
Requirements.....	2
C++ Language Concepts	3
System Design & Solution	4
In-Code Documentation & Code Quality	5
Submissions	5
GitHub Source Code.....	5
Screen Recording:	5

PROG 2100 FINAL PROJECT: STUDENT INFORMATION SYSTEM

Assignment Value: 40% of overall the course mark.

Due Date: **29th Nov** (See due date designated on the assignment on D2L.)

Late submissions will receive the standard late submission penalty as stated in the course outline. (5% overall deduction per day late)

Assignment Instructions:

Use IDE to create applications (c++ files) in which you'll code the solution for the given problem.

Submissions:

You will submit your work for this assignment via GitHub. A GitHub Repo should include all required C++ files along with console output. **You must upload the solution to the public GitHub repo.**

Evaluation:

To ensure the greatest chance of success on this assignment, be sure to check the marking rubric contained at the end of this document or in D2L. The rubric contains the criteria your instructor will be assessing when marking your assignment.

Final Project: Student Information System

Requirements: -

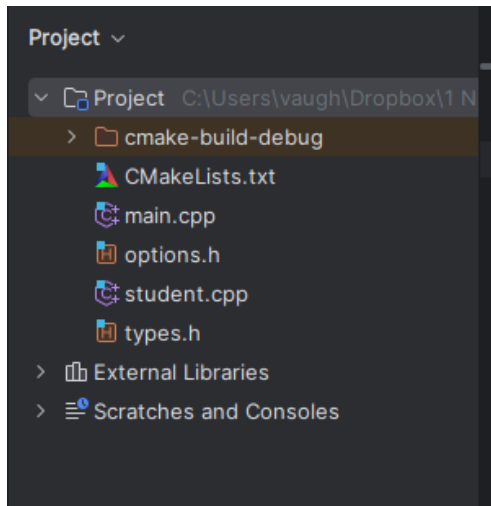
A program that stores information about the students in a class and enables users to manipulate the data regarding each student. The program has the following features:

1. Each student in the class has these attributes:
 - Last name
 - First name
 - Student number
 - Grade for midterm #1
 - Grade for midterm #2
 - Grade for final
2. Prints the list of all students in the class on the screen.
3. Print the list of students in alphabetical order of their last names (If two or more students have the same last names, the program will sort based on their first names)
4. Enables the user to add more students to the list.
5. Calculate the letter grades of all students based on the following rule:
 - A: For an average higher than 90%
 - B: For an average between 80%-90%
 - C: For an average between 70%-80%
 - D: For an average between 60%-70%
 - F: For an average below 60%
6. Print the data of a specific student.
7. Remove a particular student from the list.
8. Delete all student data.
9. Users can set the percentage that each of the midterm and final grades contribute to the average grade of students (By default there is 25% for midterm #1, 25% for midterm #2, and 50% for the final grade)

Your solution must contain examples demonstrating your understanding of the appropriate use of C++ language concepts.

C++ Language Concepts

I covered many C++ language concepts such as variables, datatypes and pointers. There are logic control statements (error-handling). There is no array but I did use a vector with respect to a struct. There is some OOPS but we did not cover much OOPS in C++. Also, there are functions, classes and File I/O. There are multiple types of types of files, too.



```
class startup : public Student {
public:

    static void data() {
        cout << "\n\nLoading Program";
    }
};
```

```
typedef struct Student {
    //Last Name, First Name, Student Number, Midterm 1, Midterm 2, Final Project
    string firstName;
    string lastName;
    int studentNumber;
    int midTermOne;
    int midTermTwo;
    int finalProject;
}Student;
```

```
/*
 * The main code that calls on other code.
 * The decisions are made here before calling other code.
 * options.h is the main menu option
 * types.h is the list of datatypes for the vector in the student.cpp for the student info.
 * While-loop that is always true until exit is selected.
 * Each option works as outlined.
 */
int main() {
    string choice;
    StudentList students;
    startup startup;
```

System Design & Solution

- Dynamic Input/Output: Completed
- Fulfill all the Mentioned Requirement: Completed

```
-----  
1) Print all Student Data  
2) Sort the list of students in alphabetical order of last name  
3) Add a new student and add marks  
4) Calculate the letter grade of all students  
5) Change the information of the previous student  
6) Print the data of a specific student  
7) Remove student from the list  
8) Delete All student data  
9) Change grading scheme  
10) Exit  
-----
```

```
2
```

```
2) Sort the list of students in alphabetical order of last name.
```

```
Sorting names...
```

```
Student 1 [Last name, First Name] H, Chris
```

```
Student 2 [Last name, First Name] Vaughan, Peter
```

```
    else if(choice == "10") {  
        //Complete - exits with error handling  
        exitOption();  
    }  
  
    else  
        //Error-handling  
        cout << "\nPlease enter a valid answer. \n";  
}
```

In-Code Documentation & Code Quality

- There is in-code documentation - each section is commented.
- There is error-handling. However, we should have a few extra classes on how to best approach error-handling. I have been learning lots about error-handling. There are probably better approaches to error-handling that has not been taught.

```
void exitOption() {
    char response;
    while(true) {
        cout << "\nAre you sure you want to exit? [Y/N]\n";
        cin >> response;

        if(response == 'Y' || response == 'y') {
            cout << "Exiting\n";
            exit(0);
        }
        if (response == 'N' || response == 'n') {
            cout << "Continue the program\n";
            break;
        }
        else {
            cout << "Invalid - Try again...\n";
        }
    }
};
```

```
/*
 * These are most of the option from the main menu - excluding quit.
 * Each function does what it is supposed to do in the parent class and child class.
 * The parent class named StudentList deals with the students.
 */
class StudentList {
public:
```

Submissions

GitHub Source Code

Link: <https://github.com/Fall2024-NSCC-ECampus/final-project-student-information-system-cadalac-don.git>

Screen Recording:

Completed