***Documentation***

This project was done by Moaz Ashry, Lucca Moura, Jennifer Marku.

Moaz did the **AddOne function**, **searchOne function** and **addSubjectAndGrade function**.

Jennifer did **toLowercase** function and worked on the **int Main function**.

Lucca did the **searchSubjectAndGrade function**, **showTopStudent function** and made the code look nicer.

We all documented the code together and explained all parts of the code to each other.

Moaz made a GitHub account and added the .cpp and the .exe of the code along with the documentation of this project.

Lines:1-8

Including libraries.

Most important parts to pay attention to: fstream

vectors

sstream

Line:10

Make the code look cleaner.

Lines: 13-18

Defining the function which will be written below the main function.

String toLowercase

Lines:115-117

**Takes in the input which is referenced by using the operator (&).**

Line: 115

We initialize a variable called lowerCaseInput.

Line: 116

Changes all letters that are capital to small letters.

Line:117

Return back the value of lowerCaseInput.

Void addOne

Lines: 120-150

Line: 122-1225

It is the first function void addOne.

Initialize:

1. vector named names.
2. variable named input as string.
3. communicate with the user to ask them to enter the student names.

Lines: 126-130

Loop to take in continuous inputs until the user types stop/ all saved into the names vector.

Lines: 132-134

Checking using the “names.empty” function, if there is nothing stored inside the names “vector” it will communicate with the user and say no names entered.

Line: 137

Checking if the file names “searchOne.txt” where we will store all the names entered by the users, also checks if it is present or not.

Lines: 138-140

“If” function checks if this file is there or not, if not it will communicate with the users and say it could not open the file for writing “not there”.

Line: 143

Identifying a vector named “it” which contains an “iterator” operator which allows us to access all of the variables within the “it” vector.

Lines: 144-146

For loop- shows what is inside of the file after saving.

Line:147

Closes the file so it allows us to edit and save into other files. (C++ does not allow us to work on multiple files in parallel.)

Line: 149

Communicating with the user and telling them that the names are saved.

Void searchOne

Lines: 152-176

**Takes in 2 parameters, a referenced fileName variable and a nameToSearch variable. Both are referenced by the operator(&).**

Lines: 154-157

Check if the file is present or not.

Line: 154

Opens the file and converts it into a required readable format.

Line: 155-157

If statement that tells the user that the file is not found if (!file)is true.

Lines: 160

Created a variable named “line” as a string.

Line: 161

Created a variable named “found” as boolean and gave it the value false.

Line: 163

Creates a new variable as a string and that variable stores the value of nameToSearch after it has converted all the capital letters to small letters.

Lines: 165-171

“While” loop that takes in the input from the user to search up that input inside of the file and the “if” loop from lines 167-169 inside of the while loop searches the names through the files. If its found it's going to communicate with the users that it is found and then change the value of found to true.

Line: 173-174

“If” statement that communicates with the users “name not found”, if not found is true.

Void addSubjectAndGrade

Line: 178-218

Line: 180

We initialize the variables name and subject as strings.

Line: 181

We initialize the variable grade as a double.

Line: 182

We initialize the variable choice as a boolean.

Line: 184

Communicating with the user and asking them to enter the students full name.

Line: 185

Taking in input from the user and saving it into the variable “name”.

Line: 191

Communicating with the user and asking them to input a grade between (0-100).

Line: 192

Taking in the input from the use and saving it into the variable grade.

Line: 193

Clears the input buffer.

Line: 195-197

If statement that gives a range for grade checks if it is between 0 and 100.

Line: 200

The ofstream function tries to open the file “searchSubjectAndGrade.txt”

Lines: 201-203

“If” function that runs when (!outFile) is true. When (!outFile) is true, we communicate with the user and say that the file couldn't be opened.

Line:206

Double check the grade by showing the user the information they just added in the format (name:subject:grade)

Line: 207

Closes the file so we can work on other files.

Line: 209

Communicates with the user that the subject and grade has been saved successfully.

Line: 211

Asks the user if they want to add another subject for the same name entered above.

Line: 212

Saves the users answer in a variable named choice.

Line: 213

Clears the input buffer in order for a more accurate answer to be saved.

Lines: 215-217

While loop that determines if the choice is a lowercase y or uppercase Y. it will return to the menu in the add subjects and grades function.

Void searchSubjectAndGrade

Lines: 220-256

**Takes in the filename as a parameter and is referenced by the operator (&).**

Line: 222

Initializing a variable named searchName as a string.

Line: 223

Tells the user to enter a name to search in the database.

Line: 224

Takes in the input from the user including all of the spaces present.

Line: 226

Opens the file and allows us to read its contents.

Lines: 227-229

“if “ function that checks displays “file not found” when (!inFile) is true.

Line: 232

Initializes a variable named line as a string

Line: 233

Initializes a variable named found.

Line: 235

Initializes a variable named lowerSearchName as a string which is set to the value of searchName after the function “toLowercase()” is applied.

Line: 237

Outputs to the user “subjects and grades for” (searchName).

Lines: 239-249

A while loop that reads the contents of the file.

Lines: 247-249

“If” statement that compares the name with lowercase letters and name with uppercase letters. And if found, give the subject along with the grade. Then changes the state of the “found” variable to true.

Lines: 253-254

“If” statement that tells the user that there are no grades or subjects for the name entered above. If (!found) is true

Void showTopStudent()

Lines: 258-294

Line: 260

Allows the file to be read and for information to be extracted.

Lines: 261-263

“If” statement that tells the user that there are no grades. If (!found) is true

Line: 266

Initializes the variables line and topStudent as string.

Line: 267

Initialize a variable highestGrade as a double with the value -1.

Lines: 269-282

A while loop that reads the contents of the file.

Lines: 287-292

“If else” statement that gives the top students name and grade when the condition highestGrade is not equal to one. Else it would say that there were no valid grades found.

Main function

Lines: 20-111

* Entire “int main” is enclosed in a “while loop” and this allows us to repeat the function until the user states otherwise by entering exit.
* Inside the “while loop” we are communicating with the user and we made a list with options to choose from 1 to 4 .
* We initialize choice as an integer and we took in the input from the user and saved it into choice. Then in line 32 is to clear the input buffer (it is like memory that you can read and write from that is considered extra storage, it clears out so it takes in fresh input).

Lines: 37-108

“Switch” case statement.

Case 1

Lines: 38-67

In case 1 we have another “while loop” that allows us to reiterate the contents of case 1.

Initialize the variable operation as string then we communicate with the user and give 3 choices:

1.add

2.search

3.back to main menu.

Then you enter the choice.

After it we initialize the variable sub-choice as an integer and we take in the input from the user and save to sub-choice. Then we clear the input buffer again.

Lines: 50-64

If statement that compares the input saved in sub-choice with the nr 1-2-3.

* If its nr 1: it will launch the function “addOne” and allow the users to add names.
* If its nr 2: it will allow the user to search up a name by initializing “name” as a string and then communicate with the user and ask them to enter a name, after that we use the get line function to take in the input including the spaces. After that we launch the “searchOne” function which allows the user to search up the name by passing through the name of the file which is “searchOne.txt” and the name entered.
* If choice is equal to 3: it will break and go back to the main menu. If anything else is entered it will give an invalid choice and ask you to try again.

Case 2

Lines: 68-97

Includes a while function that allows the user to exit whenever they please.

Initialize a variable named operation as a “string” then we communicate with the user by giving them a choice to either:

1) add student subject and grade

2) search student subject

3) search student grade

4) go back to the main menu- and then enter a choice.

We initialize sub-choice as an integer and we take in the input of the user and save it to sub-choice. Then we clear the input buffer again.

Lines: 81-95

“If” statement that compares the input given from the user with 1-2-3-4.

* If sub-choice equal to : iit will run the function “addSubjectAndGrade” which allows the user to add subjects and grades.
* If sub-choice equal 2 : it will run the function “searchSubjectAndGrade” which allows the user to search subjects by passing through “searchSubjectAndGrade.txt” file and true.
* If sub-choice equal to 3 : it will run the function “searchSubjectAndGrade” which allows the user to search grades by passing through “searchSubjectAndGrade.txt” file and false.
* If sub-choice equal to 4 : it will break and exit to the main menu. Anything else would give an error.

Case 3

Lines: 99-101

Runs the function showTopStudent which shows top students.

Case 4

Lines: 102-104

Exists the program by returning 0.

Lines: 105-107

Default which basically for everything that is not 1 through 4 it will give an error.