Prathyoosha Chaya, 11/03/2016

OBJECTIVE:

The purpose of this program is to archive the user’s data on subjects learned in CS 162 so far, what they learned in that subject, a rating on how comfortable they are in that topic, and what to do in order to improve. The functionality involved in this program allows the user to add new subjects that they have learned, display all their old subjects and corresponding information, edit their rating of how comfortable they are with previous subjects, and create/load new practice ideas. None of the following functions have any return value (void type), with the exception of the menu function.

FUNCTIONS:

Read function: The purpose of the write function is to take the information in the text file and store it into an array of structures. It receives the array of structures and an array index counter as parameters by reference, which is created in the main function, and uses the opened text file to write the data. Before entering the loop to read in, it first reads an array of characters until it encounters “;” as a delimiter, and stores this into the first member of the struct – subject. This delimiter is then ignored. The purpose of this is to set the EOF value before entering a reading loop that terminates if the end of file is reached. The loop is then entered, and the program remains in this loop until EOF. Here, the next field (“what have I learned”) is written into, and the “;” delimiter ignored. This continues with new members (rating, improvement ideas) until the “\n” delimiter is reached. This signifies that the first element of the array (first subject) has all its information members filled, and the program increments the array index counter. Before looping again, the program attempts to read the first member of the next subject, so that EOF is prepared for looping (if there are no more subjects).

Display function: The purpose of the display function is to display all subjects with their corresponding learned information, rating, and improvement ideas. This function will take the array of structures made in the main function (and revised in the read function) and output the first member (subject), second member (“what have I learned”), third member (rating), and fourth member (“improvement ideas”). It separates each array element with two newlines and each struct member with a single newline.

Write function: The purpose of this function is to write the contents of the array of structs back into the file, for the “edit rating” function and the “load/create improvements” function. It takes the array of structs and the array /counter variable as reference. In a loop a local index variable is created (type int). The object members will be outputted to the file (no append), separated with “;” delimiters. At the end of the subject, a “\n” delimiter is placed and the index is incremented. This loops until the index meets the array counter value. It is not necessary to be run for the “add subject” function, as the subject is appended to the end of the file within the function.

Menu function: This function takes no parameters and returns an int value. It outputs the menu for the user to select, and return the user’s selection. The menu displayed is shown in the main function description below.

Add subject function: The purpose of this function is to add new subjects at the user’s request. If the array counter is less than 20, the function will prompt the user to enter the name of the new subject, and will store this into the first struct member at the array of the index of the revised array counter + 1 (in the read program). Next, the user will be able to enter “what they have learned” in this subject, and this will be stored into the second structure member. This continues until all structure members have been filled out, while ignoring ‘\n’ at each step. At the end, this is written into the text file by appending to the end of the file, and separating each member with the “;” delimiter, and each subject with a newline delimiter. If the user would like to add another subject, this will loop again, but the loop will not enter if the array counter is greater than 20.

Edit rating function: The purpose of this function is to edit the rating of any previously entered subject. First, the function will output all of the subjects so far. To do this, the first member of each struct will be outputted, and this will loop until the array counter is met. Then, the user will be prompted to enter the name of the subject’s rating they want to change. The strcmp function is used to match what the user entered to the desired struct object (“Variables”, “Data Types”, etc.). Before matching, both the struct object and the user-inputted value is turned into all lowercase letters, to avoid case-sensitivity. If the subject desired is not found, then the user will be alerted that the subject does not exist. If the subject is found, then the user is asked what they would like to change the rating to. The third member of that struct is changed to the user-inputted value. The user is then asked if they would like to alter ratings again, and the function will loop from the beginning if desired.

Load/create improvement ideas function: The purpose of this function is to display the improvement ideas and give the option to add new improvements. The array of structures is passed into the function by reference. First, the subjects are all displayed and the user is asked which subject they would like to load improvements for. This user-inputted value is compared to the existing subjects (with both being converted to lowercase), and if a match is found, then the improvement ideas are displayed (by outputting the third member of the struct). If no match is found between subjects, then the user is alerted, and the program asks the user again. After a match is found and displayed, the user is given the option to write what they would like to add to the improvements section. If they would like to, then they can enter improvements and they are appended to the end of this array of characters, so that previous improvements are not discarded. Finally, the user is asked if they would like to run add more improvements to other subjects. If they say yes, then the function loops.

MAIN FUNCTION:

In the main function, there will be a menu structure to guide the user through the actions they want to perform. First, the subject struct and its members – “subject name” (array of characters), “what I have learned” (array of characters), “rating” (int), and “improvement ideas” (array of characters) are created. Then, the input file and output file variables (ifstream/ofstream type), array index counter (int type, to count how many subjects there are at certain stages of the program) are created, and a “menu option” variable (type int, to keep track of the menu item entered). The text file is opened, and the read function is called, with the parameters as the array of structures and the array counter (both passed as reference).

Then, the menu is outputted by calling the menu function. The user can choose which option they would like to execute from the following menu format:

1: Display all subjects.

2: Add a new subject.

3: Edit ratings of subjects.

4: Create/Load improvement ideas.

5: Quit archive.

The user will be prompted to enter a number, which is stored into the “menu option” variable once the function exits. If a 1 is entered, then the “display” function is entered. At this point, all data from the text file will have been read into the array of structs. Once the function is exited, the menu function is entered again. If a 2 is entered, then the “add subject” function is entered, and the menu function enters. If a 3 is entered, then the “edit rating” function is entered, then the menu function is entered. If a 4 is entered, then the “load/create improvement ideas” function is entered, and the menu is shown. If a 5 is entered, then the file is closed and the loop is exited.

Data Flow Diagram:

Array of structs, array index variable

Display function

Array of structs, array index variable

Chosen Number

Array of structs, array index variable

Load/create improvement idea function

Array of structs, array index variable

Add subject function

Array of structs, array index variable

Read function

Array of structs, array index variable

Write function

Menu function

Array of structs, array index variable

Edit rating function

Main function