Name: Aubierge Bikoi

Class: CS 499

Date: 8/19/2022

Assignment: Milestone 3 / Week 4 Assignment

In order to address the data structure and algorithm part of my project, I have selected the following artifact: Project one from CS 410. This artifact created in June 2022, aimed at using a binary code, which was later converted into a C++ code. The code generated provided the names of the clients in the database, with their usernames and passwords. Users’ authentication is very important in coding since we only need to grant access to those who have permission to access data. However, the C++ code auto generated came with some coding errors. That is why, in this section, I have decided to enhance the data structure related to this code, in order to have a fully functional code. Please note, this code can only convert into Assembly, if the initial code is working properly. I will also convert the Updated C+ + code into an assembly code, and then I will go over how assembly translate into C++.

**Initial Artifact C++:**

#include<iostream>

using namespace std;

// Declare and initialize global arrays for usernames, passwords, and clients

string usernames [] = {"j.smith", "s.brown", "r.garcia"};

string passwords [] = {"banana", "kitty", "amor"};

string clients [] = {"Bob Jones", "Sarah Davis", "Amy Friendly", "Johnny Smith",

"Carol Spears"};

/\*

\* Allows the user to change a client's service selection

\*

\*/

void ChangeCustomerChoice() {

// Initialize local variables

int client\_number = 0;

int service\_choice = 0;

int k = 0;

// Prompt for and read the number of the client to be changed

cout << "Enter the number of the client that you wish to change." << endl;

cin >> client\_number;

// Prompt for new choice selection and read input from user

cout << "Please enter the client's new service choice (1 = Brokerage, 2 =

Retirement)" << endl;

cin >> service\_choice;

// Iterates through the clients array to find a matching client number (array

index)

for (k = 0; k < sizeof(clients) ; ++k) {

if (client\_number == k) {

cout << "Selected option: " << k << endl;

}

}

}

/\*

\* Prompts user for a username and password.

\* Checks for the existence of each and then compares them

\* to determine if they match

\*

\* @return boolean: True if username and password are valid and match. False

otherwise.

\*

\*/

bool CheckUserPermissionAccess() {

// Declare local variables

string password;

string username;

int i = 0;

int j = 0;

bool match = false;

// Prompt user for username and read their input

cout << "Enter your username: ";

cin >> username;

// Prompt user for password and read input

cout << "Enter your password: ";

cin >> password;

// Search the usernames array for a matching string.

for (i = 0; i < sizeof(usernames) + 1; ++i) {

//If found break from the loop

if (username == usernames[i]) {

break;

}

// If not found, set i to a value greater than the length of the array

else if (i == sizeof(usernames)) {

// i = username array length + 2

i = (sizeof(usernames) + 2);

}

}

// Search the passwords array for a matching string

for (j = 0; j < sizeof(passwords) + 1; ++j) {

// If found, break from loop

if (password == passwords[j]) {

break;

}

// If not found, set j to value > array length

else if (j == sizeof(passwords)) {

// j = passwords array length + 1

j = (sizeof(passwords) + 1);

}

}

// Compare i and j

if (i == j) {

// If equal, return True

match = true;

}

return match;

}

/\*

\* Displays the list of clients

\*

\*/

void DisplayInfo() {

int l = 0;

// Iterates through each item in the clients array, printing each one

preceded by a number starting with 1

for (l = 0; l < sizeof(clients); ++l) {

cout << l << ". " << clients[l] << endl;

}

}

int main() {

// Declare local variables

bool authentication = false;

bool exit = false;

int menu\_choice = 0;

// Print welcome message

cout << "Hello! Welcome to our Investment Company." << endl;

cout << "Program written by Aubierge Bikoi" << endl;

// Call the ChechUserPermissionAccess function until authentication passed

while (!authentication) {

authentication = CheckUserPermissionAccess();

if (authentication) {

break;

}

else {

cout << "Invalid Password. Please try again.";

break;

}

}

// The selection menu will display after each switch statement until the user

selects 3

while (!exit) {

cout << "What would you like to do?" << endl;

cout << "DISPLAY the client list (enter 1)" << endl;

cout << "CHANGE a client's choice (enter 2)" << endl;

cout << "EXIT the program (enter 3)" << endl;

cin >> menu\_choice;

cout << "You chose " << menu\_choice << endl;

switch (menu\_choice) {

case 1 :

DisplayInfo();

break;

case 2 :

ChangeCustomerChoice();

break;

case 3 :

exit = true;

break;

}

}

return 0;

}

Output:

Text

Description automatically generated

**Updated Artifact:**

/\*

\* Description: Assignment 5.2, Project 1, SC 410.

\*

\* Author: Aubierge Bikoi

\*

\* Date: 06/12/2022

\*/

#include<iostream>

using namespace std;

// Declare and initialize global arrays for usernames, passwords, and clients

string usernames[] = { "j.smith", "s.brown", "r.garcia" };

string passwords[] = { "banana", "kitty", "amor" };

string clients[] = { "Bob Jones", "Sarah Davis", "Amy Friendly", "Johnny Smith", "Carol Spears" };

/\*

\* Allows the user to change a client's service selection

\*

\*/

void ChangeCustomerChoice()

{ // <-- this marks the beginning of ChangeCustomerChoice. Where should it end?

// Initialize local variables

int client\_number = 0;

int service\_choice = 0;

int k = 0;

// Prompt for and read the number of the client to be changed

cout << "Enter the number of the client that you wish to change." << endl;

cin >> client\_number;

// Prompt for new choice selection and read input from user

cout << "Please enter the client's new service choice (1 = Brokerage, 2 = Retirement)" << endl;

cin >> service\_choice;

string clients[5] = { "Bob Jones", "Sarah Davis", "Amy Friendly", "Johnny Smith", "Carol Spears" };

for (int k = 0; k < 5; k++)

{

if (client\_number == k)

{

cout << "Selected option: " << k << endl;

}

}

}

/\*

\* Prompts user for a username and password.

\* Checks for the existence of each and then compares them

\* to determine if they match

\*

\* @return boolean: True if username and password are valid and match. False

otherwise.

\*

\*/

bool CheckUserPermissionAccess() // <-- Notice that we're still inside of the ChangeCustomerChoice() function!

{ // We cannot write a function inside of another function. (not usually at least).

// Declare local variables

string password;

string username;

int i = 0;

int j = 0;

bool match = false;

// Prompt user for username and read their input

cout << "Enter your username:";

cin >> username;

// Prompt user for password and read input

cout << "Enter your password:";

cin >> password;

// Search the usernames array for a matching string.

for (i = 0; i < sizeof(usernames) + 1; ++i) {

//If found break from the loop

if (username == usernames[i]) {

break;

}

// If not found, set i to a value greater than the length of the array

else if (i == sizeof(usernames)) {

// i = username array length + 2

i = (sizeof(usernames) + 2);

}

}

// Search the passwords array for a matching string

for (j = 0; j < sizeof(passwords) + 1; ++j) {

// If found, break from loop

if (password == passwords[j]) {

break;

}

// If not found, set j to value > array length

else if (j == sizeof(passwords)) {

// j = passwords array length + 1

j = (sizeof(passwords) + 1);

}

}

// Compare i and j

if (i == j) {

// If equal, return True

match = true;

}

return match;

}

/\*

\* Displays the list of clients

\*

\*/

void DisplayInfo()

{

int i = 0;

/\*

\*Iterates through each item in the clients array, printing each one

preceded by a number starting with 1

\*/

// Any sort of "flow control", like for loops, while loops, if statements,

// those things are \*not\* supposed to end with a semi-colon.

// v

for (i = 0; i < sizeof(clients); i++)

{

// = is for assigning a value to a variable.

// == is for comparing whether two values are equal (common when writing if statements and while loops)

// Printing text to the screen, however, doesn't require any of those operations!

//

// TODO:

// We want to print out the value that's at the ith index of our clients array.

cout << i << clients[i] << endl;

}

}

int main() {

// Declare local variables

bool authentication = false;

bool exit = false;

int menu\_choice = 0;

// Print welcome message with my name on it

cout << "Hello! Welcome to our Investment Company." << endl;

cout << "Program written by Aubierge Bikoi" << endl;

// Call the ChechUserPermissionAccess function until authentication passed

while (!authentication) {

authentication = CheckUserPermissionAccess();

if (authentication) {

break;

}

else {

cout << "Invalid Password. Please try again.";

break;

}

}

/\*

\*The selection menu will display after each switch statement until the user

selects 3

\*/

while (!exit) {

cout << "What would you like to do?" << endl;

cout << "DISPLAY the client list (enter 1)" << endl;

cout << "CHANGE a client's choice (enter 2)" << endl;

cout << "EXIT the program (enter 3)" << endl;

cin >> menu\_choice;

cout << "You chose" << menu\_choice << endl;

switch (menu\_choice) {

case 1:

DisplayInfo();

break;

case 2:

ChangeCustomerChoice();

break;

case 3:

exit = true;

break;

}

}

return 0;

}

**Output:**

Text

Description automatically generated

**What did I Enhance?**

First, I updated the comments as I was fixing the code’s errors. In line 22 of the code, a new function started, but the curly bracket was missing to mark the end of the function. As a result, I have added one curly bracket in line 44. In line 49 of the initial code, I removed the curly bracket next to bool CheckUserPermissionAccess(), since we were still inside of the ChangeCustomerChoice() function. Hence, we couldn’t add a function inside of another function. I have also updated the comment to reflect the changes I made. Given that I had the tendency to mistake the l for 1, I changed all the l from the initial artifact into i in the enhanced artifact. Line 100 to 101 was a long comment. However, the following symbols were use “//” rather than “/\*”. Once the comment went to line 101, C++ compiler compiled it as another command and thus there was an error message. I have updated the comment from “**// Iterates through each item in the client’s array, printing each one**

**preceded by a number starting with 1**” to “**/\***

**\*Iterates through each item in the client’s array, printing each one**

**preceded by a number starting with 1**

**\*/**”. Given that any sort of "flow control", like for loops, while loops, if statements, are not supposed to end with a semi-colon, I removed the extra semi colons, and I also updated the comment to reflect the changes I made. Finally, line 125 was a long comment which extended to 126. As a result, the compiler mistook it for a command and provided an error message. This has also been fixed.

**Assembly vs C++:**

In order to get the assembly code, I entered the following command into Codio’s terminal, once I corrected the cpp file for project 1.

g++ -S Project1Files/Project1.cpp

|  |  |
| --- | --- |
| Assembly vs C++ | |
| Assembly | C++ |
| .LFE1996: .size \_\_tcf\_2, .-\_\_tcf\_2 .section .rodata .LC18: .string "j.smith" .LC19: .string "s.brown" .LC20: .string "r.garcia" .LC21: .string "banana" .LC22: .string "kitty" .LC23: .string "amor" | string passwords [] = {"banana", "kitty", "amor"}; |
| .LC10: .string "Hello! Welcome to our Investment Company." .align 8 .LC11: .string "Program written by Aubierge Bikoi" .align 8 .LC12: .string "Invalid Password. Please try again." .LC13: .string "What would you like to do?" .align 8 .LC14: .string "DISPLAY the client list (enter 1)" .align 8 .LC15: .string "CHANGE a client's choice (enter 2)" .LC16: .string "EXIT the program (enter 3)" .LC17: .string "You chose" | cout << "Hello! Welcome to our Investment Company." << endl;  cout << "Program written by Aubierge Bikoi" << endl; // Call the ChechUserPermissionAccess function until authentication passed while (!authentication) { authentication = CheckUserPermissionAccess(); if (authentication) { break; } else { cout << "Invalid Password. Please try again."; break; } } // The selection menu will display after each switch statement until the user selects 3 while (!exit) { cout << "What would you like to do?" << endl; cout << "DISPLAY the client list (enter 1)" << endl; cout << "CHANGE a client's choice (enter 2)" << endl; cout << "EXIT the program (enter 3)" << endl; |

**What did I learn? What skills did I display?**

This assignment allowed me to showcase my understanding of various languages such as Assembly, and C++. The C++ code provided recursive functions with a lot of if-else statements. Looking at the C++ code, we can also notice that this code contains a partition algorithm using two indexes such i and j as shown below:

// Compare i and j

if (i == j) {

// If equal, return True

match = true.

}.

Doing this assignment allowed me to analyze the data structure and algorithm in a project. I was able to evaluate computing solutions in order to solve a problem using algorithm and data structure by making sure the errors in the data structure are addressed. It also allowed me to enhance my security mindset, and anticipate on potential attacks, by only granting access to those that have permission to access data. While reviewing the initial code generated by Codio, I was able to implement some of the core concepts of secure coding such as documentation (by commenting), debugging, but also reviewing and addressing compiling errors within the data structure of the code. This project allowed me to convert a binary file into an Assembly file and was able to explain the functionalities of the blocks of assembly codes regarding C++, with no inaccuracies, as shown by the output.