Project Report:

Contact Book System

Lizbeth Sanchez

COP 2006: Programming 1

Content

* With this project I tried to add as many concepts as possible that were taught in class. So, the project does contain pointers, control and loops, structures, and other types of functions. So, for this project the main goal was for me to be able to add a concept to the main code and that it would run.

Abstract

* In the project I was able to add a fifth component to the code and this means adding a different section to the code where the person using the code can view the contacts they have added, in a “look up” section from the system. Since this project is a contact book system, I thought this would be the best thing to add to be able to look up or search up the contacts the user added to the system.

Goal of Project

* My goal for this project is to be able to find a way to add a new organization material to the original code and practice all I’ve learned to apply it for a task that is used on a daily basis. Also, to be able to use what I’ve learned and challenge myself to create and understand something useful that I can further my skills with.

Software Requirements

* For this project I used CLion with C++, but this project can be run on any system that has the C++ program. I used my personal computer that is a Dell laptop and has Windows 11, so anything like this would work.

Why I Chose this Project:

* I chose this project concept because I wanted to combine something I learned this semester with something that I could use daily. Also, I wanted to be part of a project that I know I could understand and was within my abilities and knowledge.

Existing technique and my improvisations to the existing concept:

* The existing technique had four original choices to make. These choices were adding a contact, displaying the contact, deleting the contact, and then exiting the code program. So, with my added concept to the code was just adding a section where the user can have the option to “look up” the contacts that they have added, and this gave the user the option to put in the name from all the contact they added to be able to display the contacts of just one person. This added concept was not very difficult to add because I basically had to follow all the same steps from the choice before and add a function to just pop up the certain named entered. So, that was it, I decided this was a project I could use my knowledge and was within my capabilities to complete.

Project Concept and procedure in detailed steps:

* Here I will first add the flowchart for the project.

A diagram of a flowchart

Description automatically generated

* Now I can explain the project and the steps that are taken when the program runs. First, when the user runs the program, they are introduced to five options to choose from, which are add contact, display contact, look up contact, delete contact, and exit. Now, the user can choose any of these options, but it is designed for the user to first choose option 1. So, when option one is chosen the user can add the name, phone number and email of that person. Then once that is finished the code will take you back to the display of options, now the user can choose to display the contact they have inputted and then the display will be shown. After this the screen will be taken back to the display of options and then the user can either look up or delete a contact they have inputted, and they just need to enter the name to do either option. After that the choices will be displayed again, and you can exit the program or continue adding to playing around with the options.

A screen shot of a computer screen

Description automatically generatedScreenshots of Code:

A screen shot of a computer program

Description automatically generated

1. A screen shot of a computer program

   Description automatically generated
2. A screenshot of a computer screen

   Description automatically generated

A screen shot of a computer program

Description automatically generated

1. A screen shot of a computer screen

   Description automatically generated

A screen shot of a computer

Description automatically generatedOutputs of Code:

A screen shot of a computer

Description automatically generated

More Outputs:

A screen shot of a computer

Description automatically generated

1. A screen shot of a computer

   Description automatically generated
2. A screen shot of a computer screen

   Description automatically generated

A screenshot of a computer screen

Description automatically generated

Conclusion:

What do you conclude from the work you have done and is there any technique that  
you would consider to improvise your work?

* My answer to the question above is that I conclude from this project that I have learned many things in this class and to the best of my abilities tried to show that with this project, but I know that there is so much more on which I can practice. A technique I would consider that could improve my work is just practicing my knowledge on all kinds of structures and kinds of functions that are compatible with C++ and just doing any kind of activities to use everything I learned in class to create more complex and difficult projects that can challenge me and help me gain more knowledge and experience. Like for this specific project I think something I could improve on just this project is using a class for the contact instead of struct, because this would make me able to add more things for the program and have better flexibility with certain functions and stuff. So, if I were also able to really go in depth with all the functions and data types in C++, I could really make very cool and interesting projects. So, that is what I conclude from the project.

Appendix:

* Code of the Project:

#include <iostream>  
#include <vector>  
#include <string>  
#include <algorithm>  
struct Contact  
{  
 std::string name;  
 std::string phone;  
 std::string email;  
};  
void addContact(std::vector<Contact> &contacts)  
{  
 Contact newContact;  
 std::cout << "Enter name: ";  
 std::cin >> newContact.name;  
 std::cout << "Enter phone number: ";  
 std::cin >> newContact.phone;  
 std::cout << "Enter email: ";  
 std::cin >> newContact.email;  
 contacts.push\_back(newContact);  
 std::cout << "Contact added." << std::endl;  
}  
  
void displayContacts(const std::vector<Contact> &contacts)  
{  
 for (const auto &contact : contacts)  
 {  
 std::cout << "Name: " << contact.name << ", Phone Number: "  
 << contact.phone << ", Email: " << contact.email << std::endl;  
 }  
}  
  
void lookupContact(const std::vector<Contact> &contacts)  
{  
 std::string searchTerm;  
 std::cout << "Enter name to search: ";  
 std::cin >> searchTerm;  
  
 bool found = false;  
 for (const auto &contact : contacts)  
 {  
 if (contact.name == searchTerm)  
 {  
 std::cout << "Name: " << contact.name << ", Phone Number: "  
 << contact.phone << ", Email: " << contact.email << std::endl;  
 found = true;  
 break;  
 }  
 }  
  
 if (!found)  
 {  
 std::cout << "Contact not found." << std::endl;  
 }  
}  
  
void deleteContact(std::vector<Contact> &contacts)  
{  
 std::string nameToDelete;  
 std::cout << "Enter name of contact to delete: ";  
 std::cin >> nameToDelete;  
  
 auto it = std::remove\_if(contacts.begin(), contacts.end(), [&](const Contact &contact)  
 { return contact.name == nameToDelete; });  
  
 if (it != contacts.end())  
 {  
 contacts.erase(it, contacts.end());  
 std::cout << "Contact deleted." << std::endl;  
 }  
 else  
 {  
 std::cout << "Contact not found." << std::endl;  
 }  
}  
int main()  
{  
 std::vector<Contact> contacts;  
 int choice;  
  
 do  
 {  
 std::cout << "\n1. Add Contact" << std::endl;  
 std::cout << "2. Display Contacts" << std::endl;  
 std::cout << "3. Look Up Contact" << std::endl;  
 std::cout << "4. Delete Contact" << std::endl;  
 std::cout << "5. Exit" << std::endl;  
 std::cout << "Enter choice: ";  
 std::cin >> choice;  
  
 switch (choice)  
 {  
 case 1:  
 addContact(contacts);  
 break;  
 case 2:  
 displayContacts(contacts);  
 break;  
 case 3:  
 lookupContact(contacts);  
 break;  
 case 4:  
 deleteContact(contacts);  
 break;  
 case 5:  
 break;  
 default:  
 std::cout << "Invalid choice." << std::endl;  
 }  
 } while (choice != 5);  
  
 return 0;  
}

References:

* [Find the best online Programming courses and Tutorials - Hackr.io](https://hackr.io/)
* [Contact Management System - C++ Project (youtube.com)](https://www.youtube.com/watch?v=yDfTA1BLuOk&t=148s)

My GitHub Link:

[GitHub Dashboard](https://github.com/dashboard)

[lizbeth789/Final-Project-for-Programming-1-C-: My presentation project for programming 1 using C++ (github.com)](https://github.com/lizbeth789/Final-Project-for-Programming-1-C-)