[Zagazig University]

Contact Book Application

Prepared for

[Dr. Mahmoud El-Mahdi]

Prepared by

[Mohamed Amin Helal]
[Section 15 , A2]
[maqmohamed8@gmail.com]

Introduction

Today, The World is Larger than Ever has more People, Our population has become larger and remembering information has become more difficult because of the large number of information that we are required to memorize, and each of us has a lot of acquaintances and friends, and since our world today has become sophisticated and uses advanced technology, so I had to program an application that allows us to save contacts.

Project was made to Boost My student Skills in Programming and it did what it was meant to do such as Boosting my skills in Object Oriented Programming in C++ in such an important Topics Like [Classes & Objects & Constructors & Destructor & Class Members], Besides it Made me Search for More information about other Topics You will find my Resources and How I have written my programs in the other sections. [Now, I'm Miles away]

P.S : Thanks For The Project, Because You made me Implement & Search For the information and that is what I was missing to complete my Skills.

Executive Summary

The Contact Book Application is a Program Designed to help Users Keep Track of their Contacts, Store their Phone Numbers and Email Address, and Easily Access This Information When they need it.
The project is Contact Book Application that has the Ability to Add, Delete, Modify, Search and Displaying Contacts. More Info, it has the Fastest and Smoothest and Efficient Process.

Project Details

Implementation :
[in this section will explain how the program works and the use of each method, function, variable and parameter]

Contact Book Application built was C++ Programming language, The App has 3 classes and main.cpp, in this section I will explain how every method work.
1st – Phone Number Class : has 3 variables (Number, Type and PhoneTypeLetter) and 5 void functions (get, show, set Phone, find number and find type).
PhoneTypeLetter Variable was made to store fisrt Letter from the user like ('m' for Mobile , 'w' for Work ..etc).
Then let's get to the Functions, at the first, getPhone() : it gets Data from the user then store it then it has a switch function to get fisrt letter of Phone Type to store Phone Type inside type to make it fast with the user instead of writing the whole world, he just need to write the fisrt letter of Phone type Like ('m' for Mobile , 'w' for Work ..etc) and that's explain why I made a variable called PhoneTypeLetter.
Then, showPhone(): it displays each of Phone Number and Phone Type.

Then, setPhone(): I made this function to help me with the other class in case I needed to switch Contacts without making changes or affecting other contacts and helped me with other things (will be explained in other classes).
Then, findNum() , findType() : to access Phone Number and Phone Type outside the class.
2nd – Contact Class : has 6 variables (fname, lname, id, email, phonecount(= 0) and ch) and 9 void functions (get, show, set Contact, find Fname, find Lname, find Id, find Email, find FLetter, PhoneCount and setPhoneCount) and Phone Number Class is linked to this Class, to make an array of objects to store multiple Phone Numbers inside one Contact and has the ability to make changes inside Phone Number like (Add, Access ..etc).
At first I made an array of objects called b[10] with size of 10, to store many Phone Numbers in one Contact, I made b[10] in public to access it in other classes, then I made phonecount=0 to keep trck with Phone Numbers inside the Contact and also to know how many Phone Numbers to Display because if I displayed all the array, there will be displayed Phone Numbers then 0s because it will be blanks inside the array which will be displayed as 0.
Then, getContact(): it stores the data from the user inside Contact Class then it asks for the user to input Phone Number and Phone Type in do..while loop phonecount is increased by 1 for every Phone Number then it breaks when the user type ‘n’ or ‘N’ to stop do..while loop.
Then, setContact(): it sets contacts variables like First name, Last name, Email and ID, it was made to make switch between contacts without affecting data used.
Then, showContacts(): it displays contact and it’s Phone Number\s with for..loop and it limits was phonecount variables that was used in getContact() Function to keep track of Phone Number Stored inside the Contact instead of displaying all Phone Numbers which was 10 based on size of the array (b[10]) .
Then, setPhoneCount(): it was made for the switching inside ContactBook Class, beacuase of sorting in Alphabitic order and Deleting Contacts by overwriting Contacts over another and other Functions …etc which PhoneCount() Function helped with also by returning phonecount variable stored inside Contact Class.
Then, findFname(), findLname(), FindEmail(), findId() and findFletter() : they all return its data to use it inside setContact() in other Functions used in Contact Book, findFletter() has a special mission which is returning the first letter of the first name to sort in alphabetic order.
3rd – Contact Book Class : has 6 variables (n(=0), i(=0), id, email, phone, delnumber and ch) and 6 void functions (Add, Delete and DisplayAll Contact, Find By ID, Find By Phone and sort Contacts) ,
Contact Class is linked to this Class, to make an array of objects to store multiple Contacts inside one Book and has the ability to make changes inside Contacts like (Add, Delete, Display and sort ..etc).
At first I made an array of objects called a[10] with size of 10, to store many Contacts in one Book and then, I made an object called temp which used for switching contacts in alphabetic order which will be discussed in sortContacts() Function and then, I made an integer variable called (n) to store count of contacts and char ch to store input of the user whether to continue adding contacts or not and then, I made int phone to store the input data from the user to search for Phone Nuumber (Provided by the User) and data stored in all Contacts in Phone Number Class then displaying matched contact/s and the same thing with int id and then, I made int delnumber to store count of deleted Contacts to use it to get

the difference between contacts the displaying new count Contacts rather than displaying unwanted contacts or 0s.

Now I will Explain all functions used

Fisrt, AddContact() : it was made to store data, entered by the user, in Every Contact in Contact Book with the array of objects (a[10]), it also has n++ which was used to keep track of contacts count, in order to know how many contacts has been entered by the user.

Then, DeleteContact() : it deletes contact from the contact book based on contact's id, now, it's time to know how does it work? , it deletes contacts by overwriting the next contact over the deleted contact then the deleted contact will be the last contact without affecting the order of the contact, then, displaying the contacts and minus the count of contacts by 1, I made this step because, when the function works it will display count of contacts at it last without displaying the deleted contact because it will be the last conatact and since I minused the contact count it won't display the deleted contact.

Then, DisplayContacts() : it displays the contacts in alphabetic order (As the last Dr.Mahdi instructions).

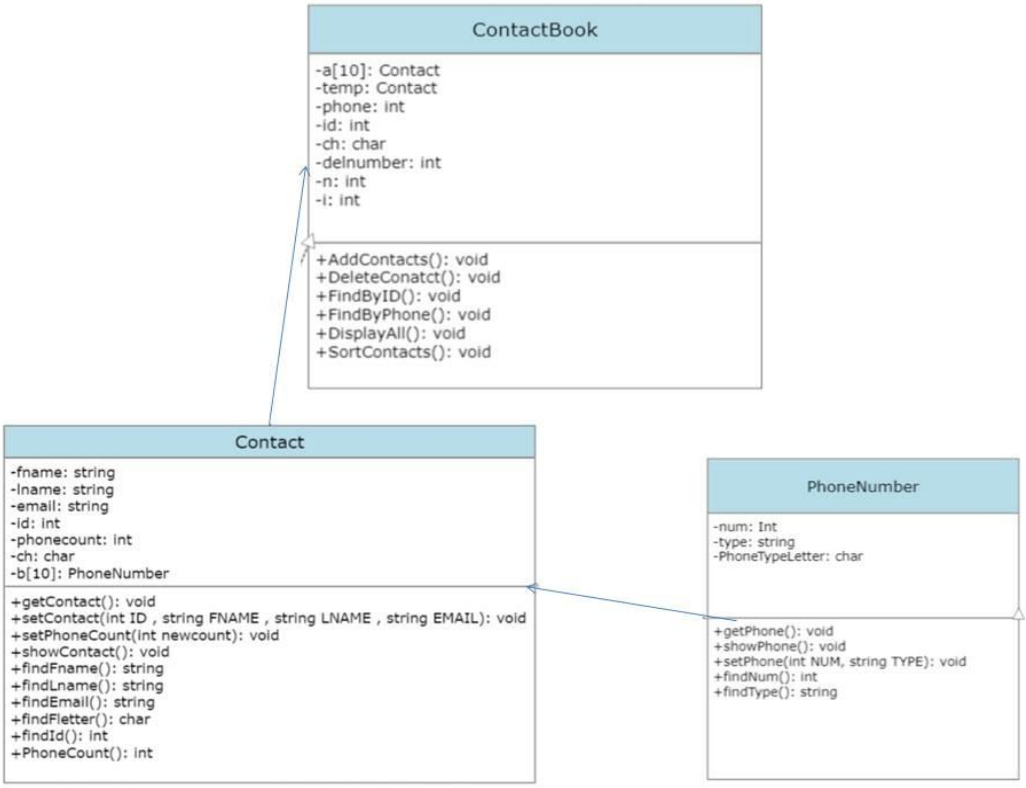
Then, SortContacts() : it sorts contacts in alphabetic order by switching contacts in alphabetic based on first name first letter as explained above, how does switching work? , I switch contacts using third contact object called Temp.

Then, FindByID() : it searches for the id in all conatacts and then displays the contact if it was found , if not it prints didn't find ID.

Then, FindByPhone() : it searches for the Phone Number in all conatacts and then displays the contact if it was found , if not it prints didn't find Phone Number.

Design:

Class Diagram



Testing :

As I said before, this Project has gone throw a lot of testing in order to get to the final result, at a specific time of the project, I was doing tests every second to make sure that the program was running successfully and been throw debugging process, every class was made and done with approximately 12 tests and was optimized with more tests to get to the best solution and the most efficient, finally the project was a success based on other people's opinion and Testing Results at a Specific Time.

Conclusion

In conclusion, the project implementation yielded positive results in terms of My Colleagues participation and test result. While the main goal of the project is to Help Users Save their Contacts and Keep Track of them, they were also able to Save, Edit, Delete, Display Them.
The Main Purpose of the Book was a success based on other's opinion and Test Results.

Upgrade :

After the big amount of testing cases and my colleagues opinion about the Project, I decided to make More Features in the Program (if the Professor approved the idea) Such as:

1. Add Save and Open File.

2. Make a good interface with Users.

3. Optimize the Execution.

Resources :
https://www.geeksforgeeks.org/
https://stackoverflow.com/
https://www.w3schools.com/cpp/default.asp
DR. Mahmoud El-Mahdi Slides and Videos.
one more resource but off-record : Having Inspiration from other People's Projects.