**Name:** Ahmed Mohamed ElGarhy

**Group:** A1

**Section:** 3

**Student Id:**

* **Description of the Program**

I made a program to store and manage user’s data like first name, last name, gender and city, ID, and some special data like:

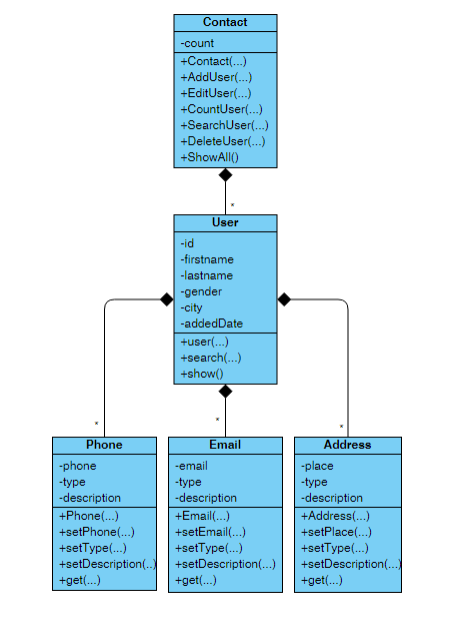
* Emails
* Phones
* Addresses

I made these data types to enable users to store more than one email, address or phone and he can also edit the rest of the data like his name, gender, …….. .

He can also access and edit a specific email, address or phone by the number (Order) of this data.

And every user has his unique ID for accessing to his own data.

User also can delete, add and edit users and the same on there’s phones, emails and addresses.

* **Class Diagram.** 
* **Data Structures I used to make this program.**

1. **Phone Class:**

To make functions to set Phone number, phone type and phone description *and*

Functions to get/return these data.

1. **Address Class:**

Functions like what in Phone class but with addresses.

1. **Mail Class:**

Functions like what in Address class but with mails.

1. **User Class:**

This class composes classes like

-Phone Class

-Mail Class

-Address Class

To make user has ability to deal with these classes’ data.

And here in this class there are functions like delete, edit and add (mail, address and phone).

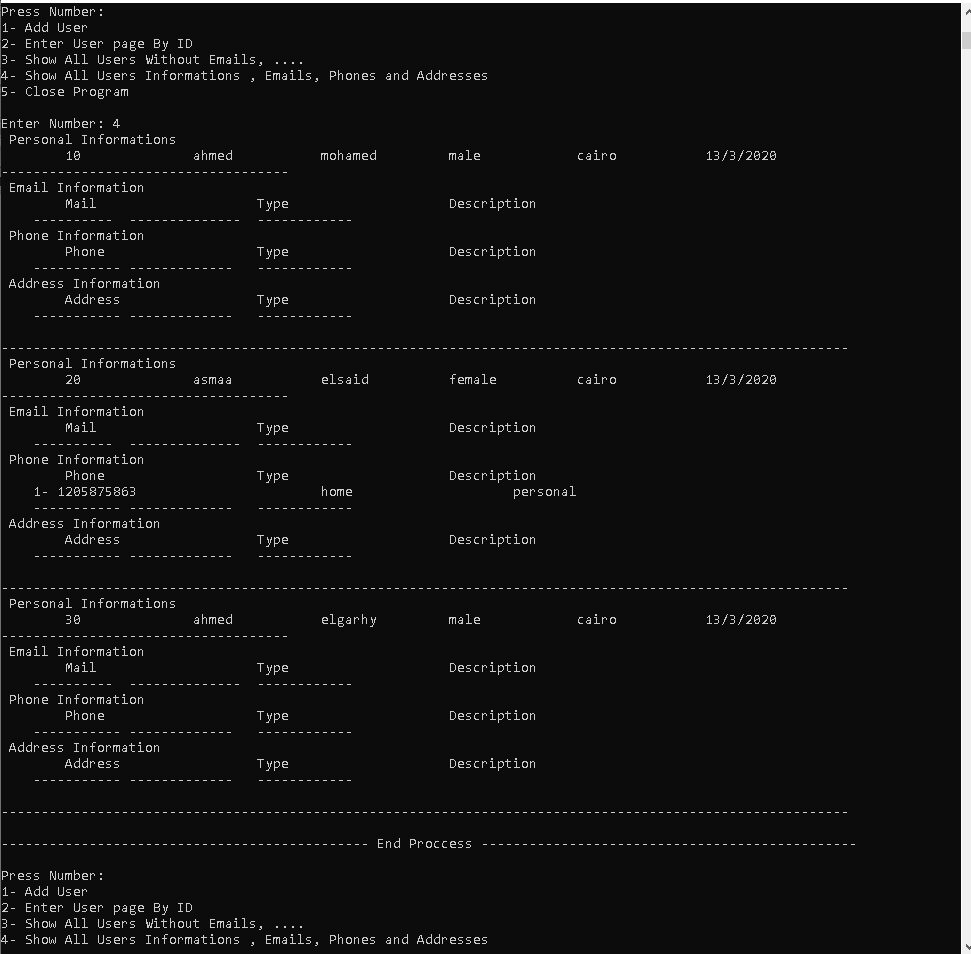
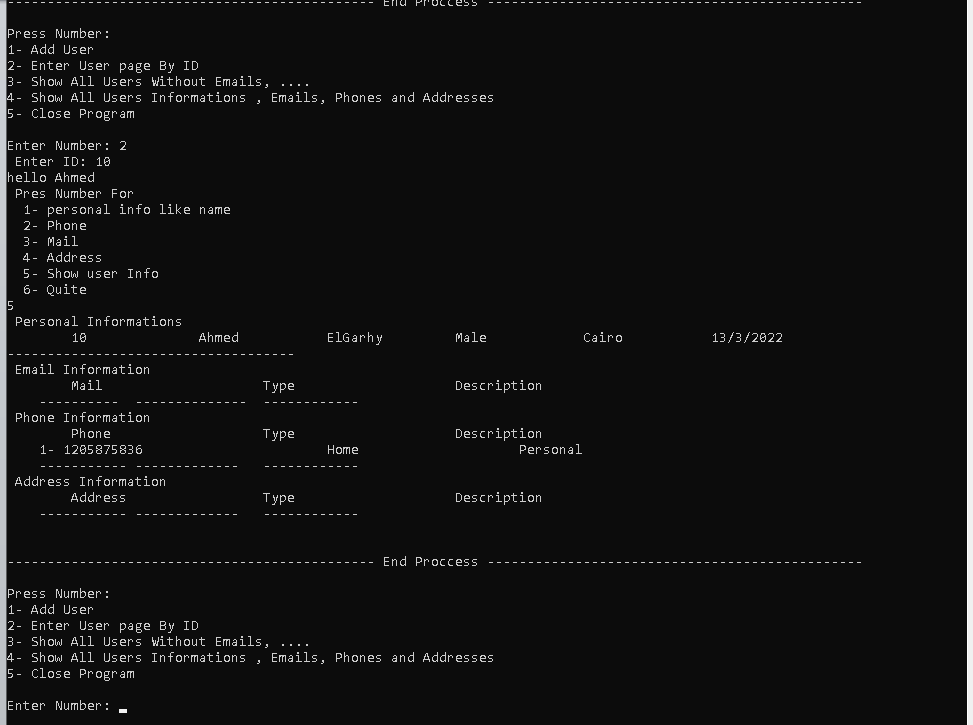
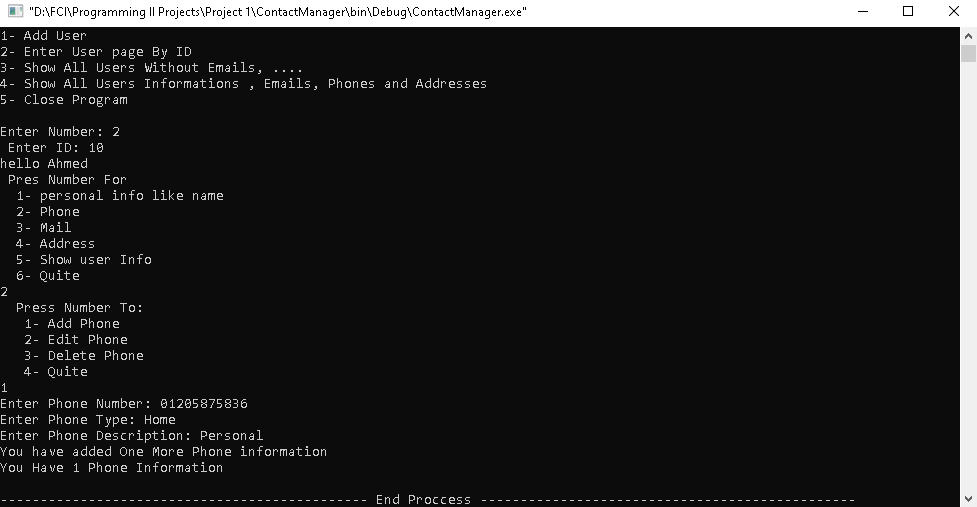
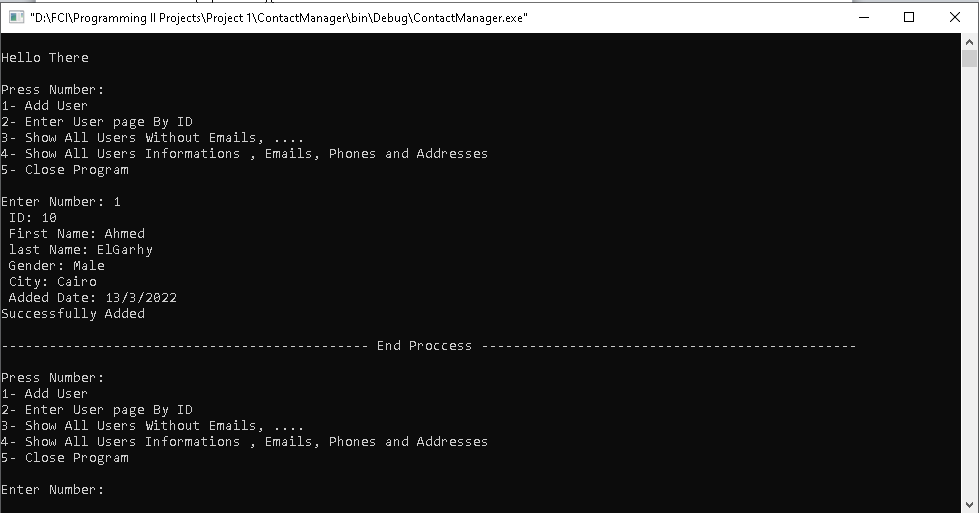
To make every user able to delete one or more mail, phone or address by its number (It’s Order)

And there is one more function to make Search if the entered id is the same id of the user it returns 1.

1. **Contact Manager**

This Class composes the user class to make an array of users every user has its own information and id

In this class I made functions to add, edit and delete user.



* **Copy of c++ code.**

**Phone.h**

#ifndef PHONE\_H

#define PHONE\_H

#include<string>

using namespace std;

class Phone

{

public:

Phone();

Phone(int, string, string);

void setPhone(int);

void setPhoneType(string);

void setPhoneDescription(string);

int getPhone();

string getPhoneType();

string getPhoneDescription();

private:

int phone;

string phone\_type, phone\_description;

};

#endif // PHONE\_H

**Phone.cpp**

#include "Phone.h"

Phone::Phone() : phone(-1), phone\_type("Undefind"), phone\_description("Undefind") {}

Phone::Phone(int number, string type, string description) : phone(number), phone\_type(type), phone\_description(description) {}

void Phone::setPhone(int n) {

phone = n;

}

void Phone::setPhoneType(string s) {

phone\_type = s;

}

void Phone::setPhoneDescription(string s) {

phone\_description = s;

}

int Phone::getPhone() {

return phone;

}

string Phone::getPhoneType() {

return phone\_type;

}

string Phone::getPhoneDescription() {

return phone\_description;

}

**Email.h**

#ifndef EMAIL\_H

#define EMAIL\_H

#include<string>

using namespace std;

class Email {

public:

Email();

Email(string, string, string);

void setEmail(string);

void setEmailType(string);

void setEmailDescription(string);

string getEmail();

string getEmailType();

string getEmailDescription();

private:

string email, email\_type, email\_description;

};

#endif // EMAIL\_H

**Email.cpp**

#include "Email.h"

Email::Email() : email("Undefind"), email\_type("Undefind"), email\_description("Undefind"){}

Email::Email(string e, string et, string ed) : email(e), email\_type(et), email\_description(ed) {}

void Email::setEmail(string e) {

email = e;

}

void Email::setEmailType(string e) {

email\_type = e;

}

void Email::setEmailDescription(string e) {

email\_description = e;

}

string Email::getEmail() {

return email;

}

string Email::getEmailType() {

return email\_type;

}

string Email::getEmailDescription() {

return email\_description;

}

**Address.h**

#ifndef ADDRESS\_H

#define ADDRESS\_H

#include<string>

using namespace std;

class Address

{

public:

Address();

Address(string, string, string);

void setAddress(string);

void setAddressType(string);

void setAddressDescription(string);

string getAddress();

string getAddressType();

string getAddressDescription();

private:

string address, address\_type, address\_description;

};

#endif // ADDRESS\_H

**Address.cpp**

#include "Address.h"

Address::Address() : address("Undefind"), address\_type("Undefind"), address\_description("Undefind") {}

Address::Address(string addres, string type, string description) : address(addres), address\_type(type), address\_description(description){}

void Address::setAddress(string a) {

address = a;

}

void Address::setAddressType(string a) {

address\_type = a;

}

void Address::setAddressDescription(string a) {

address\_description = a;

}

string Address::getAddress() {

return address;

}

string Address::getAddressType() {

return address\_type;

}

string Address::getAddressDescription() {

return address\_description;

}

**User.h**

#ifndef USER\_H

#define USER\_H

#include<Email.h>

#include<Phone.h>

#include<Address.h>

#include<string>

#include<iostream>

using namespace std;

class User

{

public:

User();

void edit(string, string, string, string, string, int);

int Search(int);

/// These Functions add Information about user => ( EX:Phone, Type, Description )

void addPhone(int, string, string);

void addEmail(string, string, string);

void addAddress(string, string, string);

/// These Functions Edit Information about user => ( index which will be edit, EX:Phone, type, description)

void editPhone(int, int, string, string);

void editEmail(int, string, string, string);

void editAddress(int, string, string, string);

/// These Functions Delete Information about user => ( index which will be edit)

void delPhone(int);

void delEmail(int);

void delAddress(int);

/// Show Information Functions

void showPersontalInfo();

void showEmails();

void showAddresses();

void showPhones();

void showAll();

//

string getFName();

private:

Phone phone[11];

Email email[11];

Address address[11];

string first\_name, last\_name, gender, city, added\_date;

int id, e\_counter, a\_counter, p\_counter;

};

#endif // USER\_H

**User.cpp**

#include "User.h"

User::User() :

first\_name("Undefind"), last\_name("Undefind"), city("Undefind"), gender("Undefind"),

added\_date("Undefind"), id(-1), e\_counter(0), a\_counter(0), p\_counter(0){}

void User::edit(string fn, string ln, string c, string g, string ad, int i) {

first\_name = fn;

last\_name = ln;

city = c;

gender = g;

added\_date = ad;

id = i;

}

// Search Function it returns 1 if entered ID is the same of this User

int User::Search(int n) {

if(n == id) return 1;

else return 0;

}

//-------------------- add ----------------------

void User::addPhone(int n, string type, string description) {

p\_counter++;

phone[p\_counter].setPhone(n);

phone[p\_counter].setPhoneType(type);

phone[p\_counter].setPhoneDescription(description);

cout<<"You have added One More Phone information\nYou Have "<<p\_counter<<" Phone Information\n";

}

void User::addEmail(string n, string type, string description) {

e\_counter++;

email[e\_counter].setEmail(n);

email[e\_counter].setEmailType(type);

email[e\_counter].setEmailDescription(description);

cout<<"You have added One More mail information\nYou Have "<<e\_counter<<" mail Information\n";

}

void User::addAddress(string n, string type, string description) {

a\_counter++;

address[a\_counter].setAddress(n);

address[a\_counter].setAddressType(type);

address[a\_counter].setAddressDescription(description);

cout<<"You have added One More Address information\nYou Have "<<a\_counter<<" address Information\n";

}

// -------------------- Edit ------------------

void User::editPhone(int index, int n, string type, string description) {

phone[index].setPhone(n);

phone[index].setPhoneType(type);

phone[index].setPhoneDescription(description);

cout<<"Done!!\n";

}

void User::editEmail(int index, string n, string type, string description) {

email[index].setEmail(n);

email[index].setEmailType(type);

email[index].setEmailDescription(description);

cout<<"Done!!\n";

}

void User::editAddress(int index, string n, string type, string description) {

address[index].setAddress(n);

address[index].setAddressType(type);

address[index].setAddressDescription(description);

cout<<"Done!!\n";

}

// ---------------------- delete ----------------

void User::delPhone(int n) {

for(int i = n+1; i<= p\_counter; i++) {

phone[i-1] = phone[i];

}

p\_counter--;

}

void User::delEmail(int n) {

for(int i = n+1; i<= e\_counter; i++) {

email[i-1] = email[i];

}

e\_counter--;

}

void User::delAddress(int n) {

for(int i = n+1; i<= a\_counter; i++) {

address[i-1] = address[i];

}

a\_counter--;

}

//----------------- show info -----------------

void User::showPersontalInfo() {

cout<<"\t"<<id<<"\t\t"<<first\_name<<"\t\t"<<last\_name<<"\t\t"

<<gender<<"\t\t"<<city<<"\t\t"<<added\_date<<endl

<<"------------------------------------\n";

}

void User::showEmails () {

cout<<"\tMail\t\t\tType\t\t\tDescription\n";

for(int i = 1; i<= e\_counter; i++) {

cout<<" "<<i<<"- "

<<email[i].getEmail()<<"\t\t\t"

<<email[i].getEmailType()<<"\t\t\t"

<<email[i].getEmailDescription()<<endl;

}

cout<<" ----------\t--------------\t------------\n";

}

void User::showPhones () {

cout<<"\tPhone\t\t\tType\t\t\tDescription\n";

for(int i = 1; i<= p\_counter; i++) {

cout<<" "<<i<<"- "

<<phone[i].getPhone()<<"\t\t\t"

<<phone[i].getPhoneType()<<"\t\t\t"

<<phone[i].getPhoneDescription()<<endl;

}

cout<<" -----------\t-------------\t------------\n";

}

void User::showAddresses () {

cout<<"\tAddress\t\t\tType\t\t\tDescription\n";

for(int i = 1; i<= a\_counter; i++) {

cout<<" "<<i<<"- "

<<address[i].getAddress()<<"\t\t\t"

<<address[i].getAddressType()<<"\t\t\t"

<<address[i].getAddressDescription()<<endl;

}

cout<<" -----------\t-------------\t------------\n";

}

void User::showAll() {

cout<<" Personal Informations\n";

showPersontalInfo();

cout<<" Email Information\n";

showEmails();

cout<<" Phone Information\n";

showPhones();

cout<<" Address Information\n";

showAddresses();

}

string User::getFName() {

return first\_name;

}

**Manager.h**

#ifndef CONTACT\_H

#define CONTACT\_H

#include<iostream>

#include<User.h>

#include<string>

using namespace std;

class Contact

{

public:

Contact(int);

~Contact();

int searchUser(int);

// add User's iInfo

void addUser(string, string, string, string, string, int);

void addUserEmail(string, string, string);

void addUserPhone(int, string, string);

void addUserAddress(string, string, string);

// Edit User's iInfo

void editUser(int, string, string, string, string, string, int);

void editUserEmail(int, string, string, string);

void editUserPhone(int, int, string, string);

void editUserAddress(int, string, string, string);

// Delete User's Info

void delUser(int);

void delUserEmail(int);

void delUserPhone(int);

void delUserAddress(int);

// Count Users

int countUser();

// show all

void showUserInfo();

void showUsers();

void showAllUsers();

private:

User\* user;

int counter, index;

};

#endif // CONTACT\_H

**Manager.cpp**

#include "Contact.h"

Contact::Contact(int n) : counter(0){

user = new User[n+1];

}

Contact::~Contact()

{

delete[] user;

}

int Contact::searchUser(int n) {

index = -1;

for(int i = 1; i<= counter; i++) {

if(user[i].Search(n) == 1) {

index = i;

cout<<"hello "<<user[index].getFName()<<endl;

break;

}

}

if(index == -1) {

cout<<"Not Found Please Input a Valid ID\n";

return 1;

}

}

// ----------------- Add -------------

void Contact::addUser(string fn, string ln, string c, string g, string ad, int id) {

counter++;

user[counter].edit(fn, ln, c, g, ad, id);

cout<<"Successfully Added\n";

}

void Contact::addUserEmail( string a, string type, string description) {

user[index].addEmail(a, type, description);

}

void Contact::addUserPhone(int a, string type, string description) {

user[index].addPhone(a, type, description);

}

void Contact::addUserAddress(string a, string type, string description) {

user[index].addAddress(a, type, description);

}

// ---------------Edit --------------------

void Contact::editUser(int n, string fn, string ln, string c, string g, string ad, int id) {

user[n].edit(fn, ln, c, g, ad, id);

cout<<"Successfully Edited\n";

}

void Contact::editUserEmail(int n, string a, string type, string description) {

user[index].editEmail(n, a, type, description);

}

void Contact::editUserPhone(int n, int a, string type, string description) {

user[index].editPhone(n, a, type, description);

}

void Contact::editUserAddress(int n, string a, string type, string description) {

user[index].editAddress(n, a, type, description);

}

//---------------- Delete --------------

void Contact::delUser(int n) {

for(int i = n+1; i<=counter; i++) {

user[i-1] = user[i];

}

counter--;

cout<<"Successfully Deleted\n";

}

void Contact::delUserEmail(int n) {

user[index].delEmail(n);

}

void Contact::delUserPhone(int n) {

user[index].delPhone(n);

}

void Contact::delUserAddress(int n) {

user[index].delAddress(n);

}

// Count Users

int Contact::countUser() {

return counter;

}

// show Functions

void Contact::showUserInfo() {

user[index].showAll();

cout<<endl;

}

void Contact::showUsers() {

cout<<"\t\tID\t\tFirstName\t\tLastName\t\tGender\t\tCity\t\tAddedDate\n";

for(int i = 1; i<= counter; i++) {

cout<<i<<"- \n";

user[i].showPersontalInfo();

cout<<endl;

}

cout<<"----------------\t------------------------\t-----------------------\t----------------------";

}

void Contact::showAllUsers() {

for(int i = 1; i<=counter; i++) {

user[i].showAll();

cout<<endl;

}

}

**Main.cpp**

#include <iostream>

#include<Contact.h>

using namespace std;

int main()

{

//Welcome Text

int user\_number, input, id, phone;

string fn, ln, g, c, ad, type, description;

cout<<"\t\t\t\t\t Made By\n"

<<"\t\t\t\t\tAhmed Mohamed ElGarhy\n\n"

<<"\t\t\t\t\t Contact Manager\n";

cout<<"Enter Today's Date: ";

cin>>ad;

cout<<"Enter Number Of Users You Will Add: ";

cin>>user\_number;

cout<<"\nHello There\n\n";

Contact user(user\_number);

while (true) {

cout<<"Press Number:\n"

<<"1- Add User\n"

<<"2- Enter User page By ID\n"

<<"3- Show All Users Without Emails, ....\n"

<<"4- Show All Users Informations , Emails, Phones and Addresses\n"

<<"5- Close Program\n";

cout<<"\nEnter Number: ";

cin>>input;

if(input == 5) break;

switch(input) {

// case one add user

case 1:

cout<<" ID: ";

cin>>id;

cout<<" First Name: ";

cin>>fn;

cout<<" last Name: ";

cin>>ln;

cout<<" Gender: ";

cin>>g;

cout<<" City: ";

cin>>c;

cout<<" Added Date: "<<ad<<endl;

user.addUser(fn, ln, c, g, ad, id);

break;

// case 2 edit user

case 2:

cout<<" Enter ID: ";

cin>>id;

input = user.searchUser(id);

if(input == 1) break;

cout<<" Pres Number For\n"

<<" 1- personal info like name\n"

<<" 2- Phone\n"

<<" 3- Mail\n"

<<" 4- Address\n"

<<" 5- Show user Info\n"

<<" 6- Quite\n";

cin>>input;

// Quite Condition

if(input == 6)break;

switch (input) {

case 1:

cout<<" ID: ";

cin>>id;

cout<<" First Name: ";

cin>>fn;

cout<<" last Name: ";

cin>>ln;

cout<<" Gender: ";

cin>>g;

cout<<" City: ";

cin>>c;

cout<<" Added Date: "<<ad<<endl;

user.addUser(fn, ln, c, g, ad, id);

break;

// -------------------------- Phone -------------------------

case 2:

cout<<" Press Number To:\n"

<<" 1- Add Phone\n"

<<" 2- Edit Phone\n"

<<" 3- Delete Phone\n"

<<" 4- Quite\n";

cin>>input;

if(input == 4) break;

switch (input) {

case 1:

cout<<"Enter Phone Number: ";

cin>>phone;

cout<<"Enter Phone Type: ";

cin>>type;

cout<<"Enter Phone Description: ";

cin>>description;

user.addUserPhone(phone, type, description);

break;

case 2:

cout<<"Enter The Index Of Phone Information You Want To Change: \n"

<<"iF You Don't Know Press Zero => 0 To Show All Of Your Information: ";

cin>>input;

if(input == 0) {

user.showUserInfo();

cout<<"Enter The Index Of Phone Information You Want To Change: ";

cin>>input;

}

cout<<"Enter Phone Number: ";

cin>>phone;

cout<<"Enter Phone Type: ";

cin>>type;

cout<<"Enter Phone Description: ";

cin>>description;

user.editUserPhone(input, phone, type, description);

break;

case 3:

cout<<"Enter The Index Of Phone Information You Want To Delete: \n"

<<"iF You Don't Know Press Zero => 0 To Show All Of Your Information: ";

cin>>input;

if(input == 0) {

user.showUserInfo();

cout<<"Enter The Index Of Phone Information You Want To Delete: ";

cin>>input;

}

user.delUserPhone(input);

break;

default:

cout<<"Invalid Number\n";

}

break;

// ------------x--------------- Phone ------------x-------------

// ---------------------------- Email Case 3 ----------------------------

case 3:

cout<<" Press Number To:\n"

<<" 1- Add Email\n"

<<" 2- Edit Email\n"

<<" 3- Delete Email\n"

<<" 4- Quite\n";

cin>>input;

if(input == 4) break;

switch (input) {

case 1:

cout<<"Enter Email Number: ";

cin>>fn;

cout<<"Enter Email Type: ";

cin>>type;

cout<<"Enter Email Description: ";

cin>>description;

user.addUserEmail(fn, type, description);

break;

case 2:

cout<<"Enter The Index Of Email Information You Want To Change: \n"

<<"iF You Don't Know Press Zero => 0 To Show All Of Your Information: ";

cin>>input;

if(input == 0) {

user.showUserInfo();

cout<<"Enter The Index Of Email Information You Want To Change: ";

cin>>input;

}

cout<<"Enter Email: ";

cin>>fn;

cout<<"Enter Email Type: ";

cin>>type;

cout<<"Enter Email Description: ";

cin>>description;

user.editUserEmail(input, fn, type, description);

break;

case 3:

cout<<"Enter The Index Of Email Information You Want To Delete: \n"

<<"iF You Don't Know Press Zero => 0 To Show All Of Your Information: ";

cin>>input;

if(input == 0) {

user.showUserInfo();

cout<<"Enter The Index Of Email Information You Want To Delete: ";

cin>>input;

}

user.delUserEmail(input);

break;

default:

cout<<"Invalid Number\n";

}

break;

// ----------x----------- Email ---------------x--------------

//----------------------- Address --------------------------

case 4:

cout<<" Press Number To:\n"

<<" 1- Add Address\n"

<<" 2- Edit Address\n"

<<" 3- Delete Address\n"

<<" 4- Quite\n";

cin>>input;

if(input == 4) break;

switch (input) {

case 1:

cout<<"Enter Address Number: ";

cin>>fn;

cout<<"Enter Address Type: ";

cin>>type;

cout<<"Enter Address Description: ";

cin>>description;

user.addUserAddress(fn, type, description);

break;

case 2:

cout<<"Enter The Index Of Address Information You Want To Change: \n"

<<"iF You Don't Know Press Zero => 0 To Show All Of Your Information: ";

cin>>input;

if(input == 0) {

user.showUserInfo();

cout<<"Enter The Index Of Address Information You Want To Change: ";

cin>>input;

}

cout<<"Enter Address: ";

cin>>fn;

cout<<"Enter Address Type: ";

cin>>type;

cout<<"Enter Address Description: ";

cin>>description;

user.editUserAddress(input, fn, type, description);

break;

case 3:

cout<<"Enter The Index Of Address Information You Want To Delete: \n"

<<"iF You Don't Know Press Zero => 0 To Show All Of Your Information: ";

cin>>input;

if(input == 0) {

user.showUserInfo();

cout<<"Enter The Index Of Address Information You Want To Delete: ";

cin>>input;

}

user.delUserEmail(input);

break;

default:

cout<<"Invalid Number\n";

}

break;

//-----------x------------------ Address -------------x----------------

case 5:

user.showUserInfo();

break;

default:

cout<<"Invalid Number\n";

}

break;

// Case 3 Show All Users Without Emails, ....

case 3:

user.showUsers();

break;

case 4:

user.showAllUsers();

break;

default:

cout<<"Invalid Number \n";

//end of Main Switch

}

cout<<"Do You Want To Make Another Task??\nIf Yes Press 1, Else Press 0\n"

<<"NOTE Any Other Number Means Yes\n Enter Your Input: ";

cin>>input;

if(input == 0){

cout<<"\nThanks For Using The Program\n made By Ahmed ElGarhy\n";

break;

}

//end of While True

}

return 0;

}

Best wishes.

Ahmed ElGarhy