**C++ PROGRAMING**

**COM313**

**A program designed and implemented by Group 8 Members**

**Members include:**

|  |  |
| --- | --- |
|  | **IKEDI GODSWILL CHUKWUEBUKA** |
|  | **UWAGA PRECIOUS CHIDERA** |
|  | **IHEONU FRANKLIN ONYEBUCHI** |
|  | **OPARA BRUNO IKEDICHUKWU** |
|  | **OLORONDU CHINYEAKA FORTUNE** |
|  | **OLEKA CHIMAIJEM PRECIOUS** |
|  | **IGBOABUCHI CHIDINMA LOVELYN** |
|  | **AGBO JOSEPH CHIMEZIE** |
|  | **ONYEKWERE GIFT OLUCHI** |
|  | **EZEAGU CHIZOBA JENNIFER** |
|  | **OKORO IFEANYICHUKWU EMMANUEL** |
|  | **EZEALA STEFFI CHIDINMA** |

**GITHUB ADDRESS: https://github.com/GOMH2021/Group8**

**Question:**  
Study file handling in C++ and create Phonebook Management System.

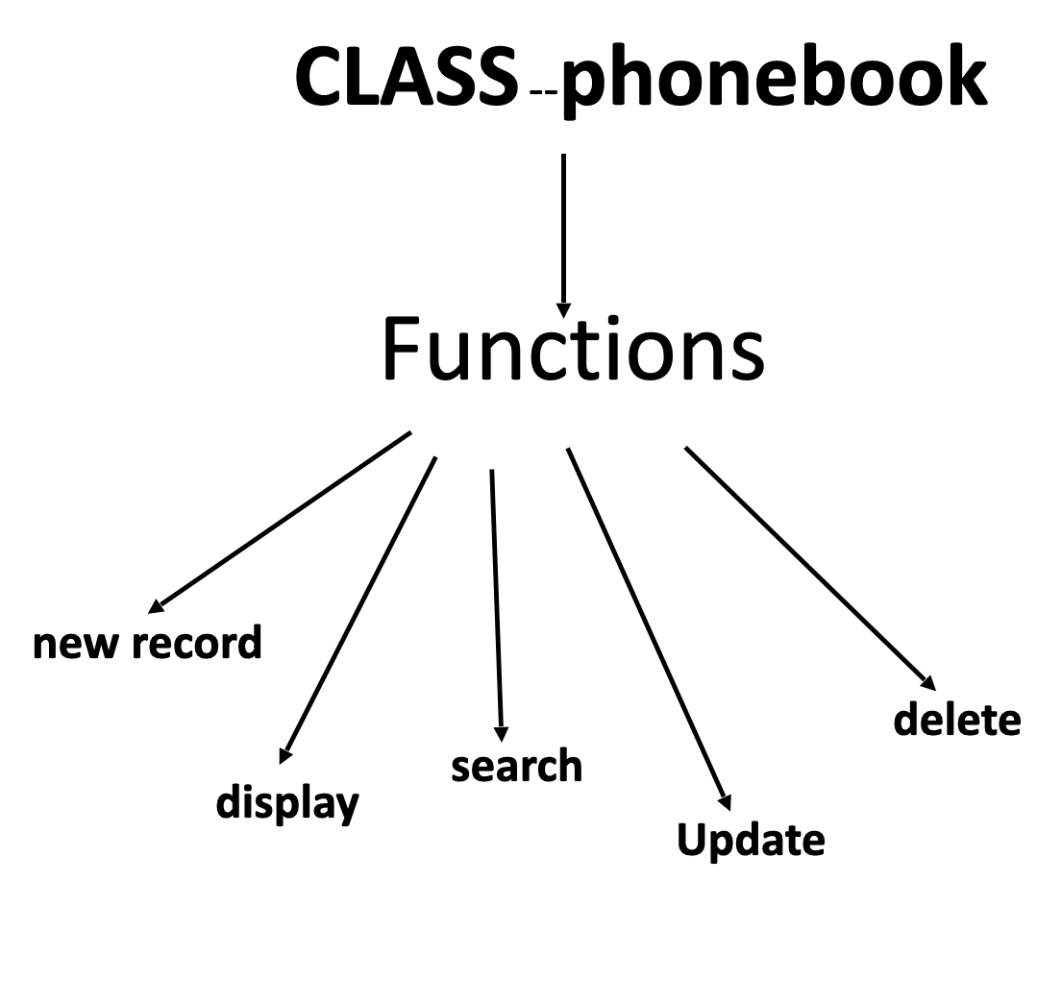
This should be a simple console application that will take you through some really nice file handling operations such as adding, searching, modifying, listing, and deleting records through the use of file.

**Features:**

You are to create this project in a way users will be able to

1. add new phone record,
2. show existing phone records,
3. search a particular phone record
4. delete phone record

**PROGRAM FLOWCHART**



**These functions, under the class phonebook, perform the following operations.**

**newrecord**

* newrecord function takes the input from the user as A NAME and A CONTACT NUMBER.

getdata function is called in newrecord .

* In getdata function we check the validity of the number entered with the help of exception handling.
* We will check the uniqueness of the contact entered. If the contact is already existing it will show the message.
* After a successful addition of the Record another message will appear on the screen .

**display**

* We have used function overloading to display records.
* One function is designed to display all the records of the contact list.
* Other function is designed to display a contact number with a given name.

**searchpname**

With a given telephone number it searches and displays the name of the person holding that phone number.

**update**

Update function has 2 options of updating %u2013

1. Updating name of the contact
2. Updating conact number.

**delete**

* Delete function deletes the record by taking the name of the the person holding the contact number.
* If two person have the same name then it will give the option of deleting the record by looking at their contact

CODING

\*File Name – main.cpp\*

#include <iostream>

#include <fstream>

#include <string.h>

#include <iomanip>

#include "phonebook.h"

using namespace std;

using namespace register1;

int main()

{

phoneBook rec;

int choice;

while(1)

{

char nm[20];

cout<<"\n\*\*\*\*\*Phone Book -Developed by CS HND1 evening GROUP 8 \*\*\*\*\*\n";

cout<<"1) Add New Record\n";

cout<<"2) Display All Records\n";

cout<<"3) Search Telephone No.\n";

cout<<"4) Search Person Name\n";

cout<<"5) Update Telephone No.\n";

cout<<"6) Delete a record \n";

cout<<"7) Exit\n";

cout<<"Choose your choice : ";

cin>>choice;

switch(choice)

{

case 1 : rec.newrecord();

break;

case 2 : rec.display();

break;

case 3 :rec.searchpname();

break;

case 4 :

cout<<"\n\nEnter Name : ";

cin>>nm;

rec.display(nm);

break;

case 5 :rec.updatetelno();

break;

case 6: rec.deleterec();

break;

case 7: exit(1);

default:

cout<<"Invalid Option!!"<<endl;

} }

return 0;

}

\*File Name – phonebook.h\*

#include <iostream>

#include <fstream>

#include <string.h>

#include <iomanip>

#include <stdio.h>

using namespace std;

namespace register1

{

class phoneBook

{

char name[20],phno[11];

public:

void getdata();

void showdata();

char \*getname(){ return name; }

char \*getphno(){ return phno; }

void update(char \*nm,char \*telno)

{

strcpy(name,nm);

strcpy(phno,telno);

}

void newrecord();

void display();

void display(char \*nm);

void searchpname();

void updatetelno();

void deleterec();

};

void phoneBook :: getdata()

{

cout<<"\nEnter Name : ";

cin>>name;

repeat:

cout<<"Enter Phone No. : ";

cin>>phno; //CAN CREATE A FUNCTION THAT CHECKS WHETHER STRING IS DIGIT OR NOT

for(int i=0;i<strlen(phno);i++)

{

try

{

if(phno[i]<48 || phno[i]>57||strlen(phno)!=10)

throw phno[i];

}

catch(char a)

{

cout<<"Enter a valid number"<<endl;

cout<<"Try again"<<endl;

goto repeat;

}

}

}

void phoneBook :: showdata(){

cout<<"\n";

cout<<setw(20)<<name;

cout<<setw(15)<<phno;

}

void phoneBook :: newrecord()

{

int found=0;

char rrep[11];

char nmopi[20];

getdata();

strcpy(nmopi,name);

strcpy(rrep,phno);

fstream file;

file.open("phone.txt", ios::in | ios::app | ios::binary );

while(file.read((char \*) this, sizeof(\*this)))

{

if(strcmp(rrep,getphno())==0)

{

found=1;

break;

}

}

file.clear();

file.close();

if(found==1)

cout<<"\n\n---Contact already exist---\n";

else

{

file.open("phone.txt", ios::app | ios::binary );

update(nmopi,rrep);

file.write((char \*) this, sizeof(\*this));

cout<<"Record Added Succesfully\n";

file.close();

}

}

void phoneBook :: display()

{

//Display All Records

fstream file;

file.open("phone.txt", ios::ate | ios::in | ios::out | ios::binary );

file.seekg(0,ios::beg);

cout<<"\n\nRecords in Phone Book\n";

while(file)

{

file.read((char \*) this, sizeof(\*this));

if(!file.eof())

showdata();

}

file.clear();

file.close();

}

void phoneBook :: display(char \*nm) //function overloading

{ //Search Tel. no. when person name is known.

char c,telno[11];

int cnt,found=0;

fstream file;

file.open("phone.txt", ios::ate | ios::in | ios::out | ios::binary );

file.seekg(0,ios::beg);

while(file.read((char \*) this, sizeof(\*this)))

{

if(strcmp(nm,getname())==0)

{

found=1;

showdata();

}

}

file.clear();

if(found==0)

cout<<"\n\n---Record Not found---\n";

file.close();

}

void phoneBook :: searchpname()

{

//Search name on basis of tel. no

char c,ch,nm[20],telno[11];

int cnt,found=0;

cout<<"\n\nEnter Telephone No : ";

cin>>telno;

fstream file;

file.open("phone.txt", ios::ate | ios::in | ios::out | ios::binary );

file.seekg(0,ios::beg);

while(file.read((char \*) this, sizeof(\*this)))

{

if(strcmp(telno,getphno())==0)

{

found=1;

showdata();

}

}

file.clear();

if(found==0)

cout<<"\n\n---Record Not found---\n";

file.close();

}

void phoneBook :: updatetelno()

{ //Update Telephone No.

int k;

fstream file;

file.open("phone.txt", ios::ate | ios::in | ios::out | ios::binary );

cout<<"Enter 1 to update name"<<endl;

cout<<"Enter 2 to update phone number"<<endl;

cin>>k;

if(k==1)

{ char c,ch,nm[20],newnm[15],teleno[15];

int cnt=0,found=0;

cout<<"\n\nEnter Name : ";

cin>>nm;

file.seekg(0,ios::beg);

while(file.read((char \*) this, sizeof(\*this)))

{

cnt++;

if(strcmp(nm,getname())==0)

{

found=1;

break;

}

}

file.clear();

if(found==0)

cout<<"\n\n---Record Not found---\n";

else

{ strcpy(teleno,phno);

int location = (cnt-1) \* sizeof(\*this);

cin.get(ch);

if(file.eof())

file.clear();

cout<<"Enter New Name: ";

cin>>newnm;

file.seekp(location);

update(newnm,teleno);

file.write((char \*) this, sizeof(\*this));

file.flush();

}

}

if(k==2)

{

char c,ch,nm[20],telno[15],newteleno[15];

int cnt=0,found=0;

cout<<"\n\nEnter Phone number : ";

cin>>telno;

file.seekg(0,ios::beg);

while(file.read((char \*) this, sizeof(\*this)))

{

cnt++;

if(strcmp(telno,getphno())==0)

{

found=1;

break;

}

}

file.clear();

if(found==0)

cout<<"\n\n---Record Not found---\n";

else

{ strcpy(nm,name);

int location = (cnt-1) \* sizeof(\*this);

cin.get(ch);

if(file.eof())

file.clear();

cout<<"Enter New Phone number: ";

cin>>newteleno;

file.seekp(location);

update(nm,newteleno);

file.write((char \*) this, sizeof(\*this));

file.flush();

}

}

file.close();

}

void phoneBook :: deleterec()

{

char bb[20],teleno[11];

cout<<"Enter the name of the record which is to be deleted:";

cin>>bb;

fstream file;

file.open("phone.txt", ios::in | ios::binary );

fstream fout;

fout.open("tempfile.txt",ios::out | ios::binary);

int i=0,j=0,found=0;

file.read((char \*) this, sizeof(\*this));

while(!file.eof())

{

i++;

if(strcmp(bb,getname()))

{

fout.write((char\*)this,sizeof(\*this));

j++;

}

// found++

file.read((char\*)this,sizeof(\*this));

}

fout.close();

file.close();

if(i-j<=1)

{

file.close();

remove("phone.txt");

rename("tempfile.txt","phone.txt");

if(j<i)

cout<<"RECORD DELETED"<<endl;

if(j==i)

cout<<"RECORD NOT FOUND"<<endl;

}

else

{

cout<<"\n\nTWO SAME CONTACT NAMES EXIST\n"<<endl;

file.open("phone.txt", ios::ate | ios::in | ios::out | ios::binary );

file.seekg(0,ios::beg);

while(file.read((char \*) this, sizeof(\*this)))

{

if(strcmp(bb,getname())==0)

{

found=1;

showdata();

cout<<endl;

}

}

file.clear();

file.close();

cout<<"\n\n NOW ENTER THE CONTACT NUMBER YOU WANT TO DELETE : "<<endl;

cin>>teleno;

//fstream file;

file.open("phone.txt", ios::in | ios::binary );

// fstream fout;

fout.open("tempfileph.txt",ios::out | ios::binary);

// int i=0,j=0,found=0;

file.read((char \*) this, sizeof(\*this));

i=0;

j=0;

while(!file.eof())

{

i++;

if(strcmp(teleno,getphno()))

{

fout.write((char\*)this,sizeof(\*this));

j++;

}

// found++

file.read((char\*)this,sizeof(\*this));

}

fout.close();

file.close();

remove("phone.txt");

rename("tempfileph.txt","phone.txt");

if(j<i)cout<<"\n\nRECORD DELETED\n\n"<<endl;

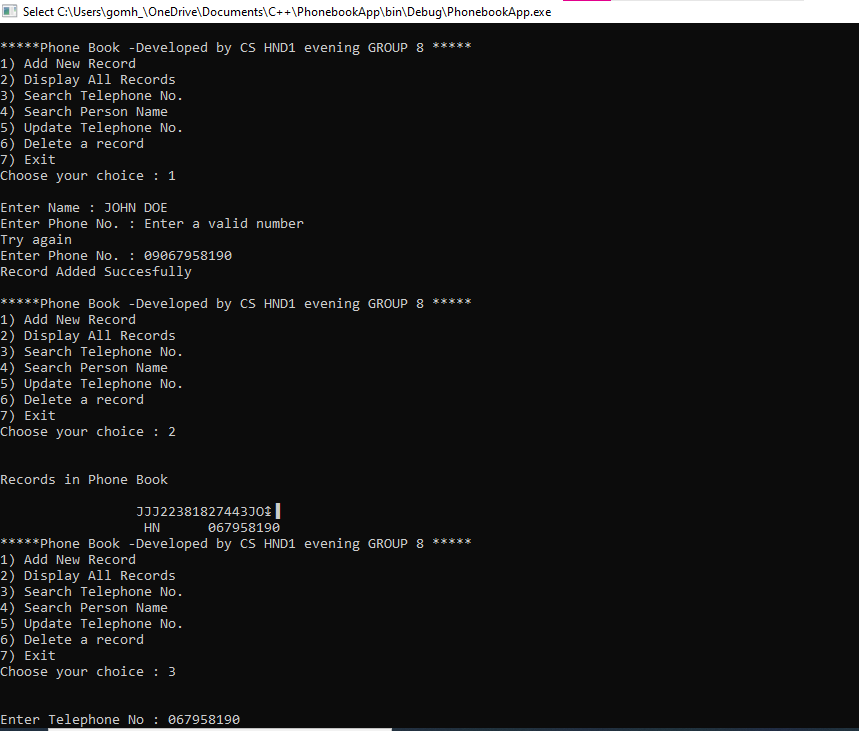
else cout<<"RECORD NOT FOUND"<<endl;

}

}

}

**OUTPUT:**

****