#include<iostream>

#include<fstream>

#include<cstdlib>

using std::cout;

using std::cin;

using std::endl;

using std::fstream;

using std::ofstream;

using std::ifstream;

using std::ios;

class account\_query

{

private:

char account\_number[20];

char firstName[10];

char lastName[10];

float total\_Balance;

public:

void read\_data();

void show\_data();

void write\_rec();

void read\_rec();

void search\_rec();

void edit\_rec();

void delete\_rec();

};

void account\_query::read\_data()

{

cout<<"\nEnter Account Number: ";

cin>>account\_number;

cout<<"Enter First Name: ";

cin>>firstName;

cout<<"Enter Last Name: ";

cin>>lastName;

cout<<"Enter Balance: ";

cin>>total\_Balance;

cout<<endl;

}

void account\_query::show\_data()

{

cout<<"Account Number: "<<account\_number<<endl;

cout<<"First Name: "<<firstName<<endl;

cout<<"Last Name: "<<lastName<<endl;

cout<<"Current Balance: Rs. "<<total\_Balance<<endl;

cout<<"-------------------------------"<<endl;

}

void account\_query::write\_rec()

{

ofstream outfile;

outfile.open("record.bank", ios::binary|ios::app);

read\_data();

outfile.write(reinterpret\_cast<char \*>(this), sizeof(\*this));

outfile.close();

}

void account\_query::read\_rec()

{

ifstream infile;

infile.open("record.bank", ios::binary);

if(!infile)

{

cout<<"Error in Opening! File Not Found!!"<<endl;

return;

}

cout<<"\n\*\*\*\*Data from file\*\*\*\*"<<endl;

while(!infile.eof())

{

if(infile.read(reinterpret\_cast<char\*>(this), sizeof(\*this))>0)

{

show\_data();

}

}

infile.close();

}

void account\_query::search\_rec()

{

int n;

ifstream infile;

infile.open("record.bank", ios::binary);

if(!infile)

{

cout<<"\nError in opening! File Not Found!!"<<endl;

return;

}

infile.seekg(0,ios::end);

int count = infile.tellg()/sizeof(\*this);

cout<<"\n There are "<<count<<" record in the file";

cout<<"\n Enter Record Number to Search: ";

cin>>n;

infile.seekg((n-1)\*sizeof(\*this));

infile.read(reinterpret\_cast<char\*>(this), sizeof(\*this));

show\_data();

}

void account\_query::edit\_rec()

{

int n;

fstream iofile;

iofile.open("record.bank", ios::in|ios::binary);

if(!iofile)

{

cout<<"\nError in opening! File Not Found!!"<<endl;

return;

}

iofile.seekg(0, ios::end);

int count = iofile.tellg()/sizeof(\*this);

cout<<"\n There are "<<count<<" record in the file";

cout<<"\n Enter Record Number to edit: ";

cin>>n;

iofile.seekg((n-1)\*sizeof(\*this));

iofile.read(reinterpret\_cast<char\*>(this), sizeof(\*this));

cout<<"Record "<<n<<" has following data"<<endl;

show\_data();

iofile.close();

iofile.open("record.bank", ios::out|ios::in|ios::binary);

iofile.seekp((n-1)\*sizeof(\*this));

cout<<"\nEnter data to Modify "<<endl;

read\_data();

iofile.write(reinterpret\_cast<char\*>(this), sizeof(\*this));

}

void account\_query::delete\_rec()

{

int n;

ifstream infile;

infile.open("record.bank", ios::binary);

if(!infile)

{

cout<<"\nError in opening! File Not Found!!"<<endl;

return;

}

infile.seekg(0,ios::end);

int count = infile.tellg()/sizeof(\*this);

cout<<"\n There are "<<count<<" record in the file";

cout<<"\n Enter Record Number to Delete: ";

cin>>n;

fstream tmpfile;

tmpfile.open("tmpfile.bank", ios::out|ios::binary);

infile.seekg(0);

for(int i=0; i<count; i++)

{

infile.read(reinterpret\_cast<char\*>(this),sizeof(\*this));

if(i==(n-1))

continue;

tmpfile.write(reinterpret\_cast<char\*>(this), sizeof(\*this));

}

infile.close();

tmpfile.close();

remove("record.bank");

rename("tmpfile.bank", "record.bank");

}

int main()

{

account\_query A;

int choice;

cout<<"\*\*\*Acount Information System\*\*\*"<<endl;

while(true)

{

cout<<"Select one option below ";

cout<<"\n\t1-->Add record to file";

cout<<"\n\t2-->Show record from file";

cout<<"\n\t3-->Search Record from file";

cout<<"\n\t4-->Update Record";

cout<<"\n\t5-->Delete Record";

cout<<"\n\t6-->Quit";

cout<<"\nEnter your choice: ";

cin>>choice;

switch(choice)

{

case 1:

A.write\_rec();

break;

case 2:

A.read\_rec();

break;

case 3:

A.search\_rec();

break;

case 4:

A.edit\_rec();

break;

case 5:

A.delete\_rec();

break;

case 6:

exit(0);

break;

default:

cout<<"\nEnter corret choice";

exit(0);

}

}

system("pause");

return 0;

}