

```
//  
//  main.cpp  
//  Banking System Project  
//  
//  Created by Gaurav Sharma on 15/10/21.  
//
```

```
#include <iostream>  
using namespace std;  
#include <string>  
#include <map>  
#include <ostream>  
#include <fstream>  
#define minbalance 500
```

```
class insufficientBalance{  
  
};
```

```
class account{  
private:  
    long accountno;  
    string firstname;  
    string lastname;  
    float balance;  
    static long nextaccno;  
public:  
    account();  
    account(string fname,string lname,float  
        balance);  
    long getaccno(){return accountno;};  
    string getfirstname(){return firstname;};  
    string getlastname(){return lastname;};  
    float getbalance(){return balance;};
```

```

void deposit(float amount);
void withdraw(float amount);
static void setlastaccno(long accno);
static long getlastaccno();

friend ostream & operator<<(ostream
    &ot,account &acc);
friend ifstream & operator>>(ifstream
    &ot,account &acc);
friend ostream & operator<<(ostream
    &ot,account &acc);
};

long account::nextaccno=0;

class Bank{
private:
    map<long,account> accounts;

public:
    Bank();
    account openaccount(string fname,string
        lname,float balance);
    account balanceenquiry(long accountno);
    account deposit(long accountno,float
        amount);
    account withdraw(long accountno,float
        amount);
    void CloseAccount(long accountno);
    void ShowAllAccounts();
    ~Bank();
};

int main() {

```

```
cout<<"***Banking System***"<<endl<<endl;
cout<<"          Select one option
    below"<<endl;
cout<<"          1.Open an Account"<<endl;
cout<<"          2.Balance Enquiry"<<endl;
cout<<"          3.Deposit"<<endl;
cout<<"          4.Withdrawl"<<endl;
cout<<"          5.Close an Account"<<endl;
cout<<"          6.Show all Accounts"<<endl;
cout<<"          7.Quit"<<endl;
ofstream
    out("accountdetails.txt",ios::app);
```

```
account acc;
Bank b;
string fname,lname;float
    amount,balance;long accno;
cout<<"Enter Your Choice:";
int choice;
cin>>choice;
while(choice!=7){
```

```
switch (choice) {
    case 1:

        cout<<"Enter First Name:";
        cin>>fname;
        cout<<"Enter Last Name:";
        cin>>lname;
        cout<<"Enter Initial Balance:";
        cin>>amount;
        cout<<endl;
```

```
acc=b
    .openaccount(fname,lname,amount);
cout<<"Congratulations Account is
    created"<<endl;
cout<<acc;

    break;
```

```
case 2:
    cout<<"Enter Account Number:";
    cin>>accno;
    cout<<endl;
    cout<<"Your Accoount
        Details"<<endl;
    acc=b.balanceenquiry(accno);
    cout<<acc;
    break;
```

```
case 3:
    cout<<"Enter Account Number:";
    cin>>accno;
    cout<<"Enter Balance:";
    cin>>balance;
    cout<<endl;
    cout<<endl<<"Amount is
        Deposited"<<endl;
    acc=b.deposit(accno, balance);
    cout<<acc<<endl;
    break;
```

```
case 4:
    cout<<"Enter Account Number:";
    cin>>accno;
    cout<<"Enter Balance:";
    cin>>balance;
```

```
cout<<endl<<"Amount is  
Withdrawn"<<endl;  
acc=b.withdraw(accno, balance);  
cout<<acc<<endl;  
break;
```

```
case 5:  
    cout<<"Enter Account Number:";  
    cin>>accno;  
    cout<<endl;  
    b.CloseAccount(accno);  
    break;
```

```
case 6:  
    //show all accounts case  
    cout<<endl;  
    b.ShowAllAccounts();  
    break;
```

```
case 7:  
    //case used for quitting;  
    cout<<endl;  
    return 0;  
    break;
```

```
default:  
    break;
```

```
}
```

```
cout<<endl;
```

```
cout<<"                Select one option  
below"<<endl;  
cout<<"                1.Open an  
Account"<<endl;
```

```

        cout<<"                2.Balance
        Enquiry"<<endl;
        cout<<"                3.Deposit"<<endl;
        cout<<"                4.Withdrawl"<<endl;
        cout<<"                5.Close an
        Account"<<endl;
        cout<<"                6.Show all
        Accounts"<<endl;
        cout<<"                7.Quit"<<endl;

        cout<<"Enter Your Choice:";

        cin>>choice;
    }

    return 0;
}

ofstream & operator<<(ofstream &oft,account
&acc){
    oft<<acc.accountno<<endl<<acc
    .firstname<<endl<<acc.lastname<<endl<<acc
    .balance<<endl;
    return oft;
}

ifstream & operator>>(ifstream &it,account
&acc){
    it>>acc.accountno;
    it>>acc.firstname;
    it>>acc.lastname;
    it>>acc.balance;
    return it;
}

ostream & operator<<(ostream &ot,account &acc){

```

```

        ot<<"First Name:"<<acc.firstname<<endl;
        ot<<"Last Name:"<<acc.lastname<<endl;
        ot<<"Account Number:"<<acc.accountno<<endl;
        ot<<"Balance:"<<acc.balance<<endl;

        return ot;
    }

account::account() {

account::account(string fname, string
lname, float balance) {
    nextaccno=getlastaccno();
    ++nextaccno;
    accountno=nextaccno;
//    accountno=++(getlastaccno());
    firstname=fname;
    lastname=lname;
    this->balance=balance;

}

void account::deposit(float amount) {
    this->balance+=amount;

}

void account::withdraw(float amount) {
    this->balance-=amount;
}

void account::setlastaccno(long accno) {
    nextaccno=accno;
}

```

```

    long account::getlastaccno(){
        return nextaccno;
    }

Bank::Bank(){
    account acc;
    //    map<long,account> accounts;
    //    map<long,account>::iterator itr;
    ifstream input("accountdetails.txt");

    while(input>>acc){
        //    while(input.eof()){
        //    input>>acc;
        //        input>>itr->second;
        //        acc=itr->second;
        accounts.insert(pair<long,account>(acc
            .getaccno(),acc));
        //        itr->first = acc.getaccno();
        //        itr++;
        //        account acc;
    }

    account acc1;
    ifstream input1("accountdetails.txt");
    if(input1>>acc1){
        map<long,account>::iterator itr5;
        itr5=accounts.end();
        itr5--;
        account::setlastaccno(itr5->second
            .getaccno());}

}

account Bank::openaccount(string fname,string
    lname,float balance){

```



```

        account acc(fname,lname,balance);
        accounts.insert(pair<long,account>(acc
            .getaccno(),acc));
        map<long,account>::iterator itr;
        ofstream
            out("accountdetails.txt",ios::app);
//
        for(itr=accounts.begin();itr!=accounts
            .end();itr++){
            out<<acc;
//        }
        out.close();
        return acc;
    }
    account Bank::balanceenquiry(long accountno){
        account acc;
        map<long,account>::iterator itr1;
        itr1=accounts.find(accountno);

        return itr1->second;
    }
    account Bank::deposit(long accountno,float
        amount){
        account acc;
        map<long,account>::iterator itr;
        itr=accounts.find(accountno);
        itr->second.deposit(amount);
        return itr->second;
    }
    account Bank::withdraw(long accountno,float
        amount){
        account acc;
        map<long,account>::iterator itr;
        itr=accounts.find(accountno);
        itr->second.withdraw(amount);
    }

```

```

        return itr->second;
    }
    void Bank::CloseAccount(long accountno){
        map<long,account>::iterator itr;
        itr=accounts.find(accountno);
        //    itr.erase(accountno);
        cout<<"Account Deleted"<<endl;
        cout<<itr->second<<endl;
        accounts.erase(accountno);
    }
    void Bank::ShowAllAccounts(){

        int i=0;
        map<long,account>::iterator itr2;
        for(itr2=accounts.begin();itr2!=accounts
            .end();itr2++){
            cout<<"Account: "<<++i<<endl;
            cout<<itr2->second<<endl;
        }

    }

    Bank::~~Bank(){
        map<long,account>::iterator itr;
        ofstream out("accountdetails.txt");
        for(itr=accounts.begin();itr!=accounts
            .end();itr++){
            out<<itr->second;
        }
    }
}

```