```
//*****************************************
*************** // HEADER FILE USED IN PROJECT
//*****************************************************************

#include<iostream>
#include<fstream>
#include<cctype>
#include<iomanip>
using namespace std;

//*****************************************
*************** // CLASS USED IN PROJECT
//*****************************************


*************** class account


{
        int acno;
        char name[50];
        int deposit;
        char type;
public:
        void create_account(); //function to get data from user
        void show_account() const; //function to show data
        on screen void modify(); //function to add new
        data
        void dep(int); //function to accept amount and add to
balance amount void draw(int); //function to accept amount and
subtract from balance  amount
        void report() const; //function to show data in
        tabular format int retacno() const; //function to
        return account number
        int retdeposit() const; //function to return balance amount
        char rettype() const; //function to return type of account
}; //class ends here

void account::create_account()
{
        cout<<"\nEnter The account No. :";
        cin>>acno;
        cout<<"\n\nEnter The Name of The account Holder : ";
        cin.ignore();
        cin.getline(name,50);
        cout<<"\nEnter Type of The account (C/S) : ";
        cin>>type;
        type=toupper(type);
        cout<<"\nEnter The Initial amount(>=500 for Saving and
>=1000 for current  ) : ";
        cin>>deposit;
        cout<<"\n\n\nAccount Created..";
}
```

```
void account::show_account() const
{
        cout<<"\nAccount No. : "<<acno;
        cout<<"\nAccount Holder Name : ";
        cout<<name;
        cout<<"\nType of Account : "<<type;
        cout<<"\nBalance amount : "<<deposit;
}


void account::modify()
{
        cout<<"\nAccount No. : "<<acno;
        cout<<"\nModify Account Holder Name : ";
        cin.ignore();
        cin.getline(name,50);
        cout<<"\nModify Type of Account : ";
        cin>>type;
        type=toupper(type);
        cout<<"\nModify Balance amount : ";
        cin>>deposit;
}


void account::dep(int x)
{
        deposit+=x;
}

void account::draw(int x)
{
        deposit-=x;
}

void account::report() const
{
        cout<<acno<<setw(10)<<" "<<name<<setw(10)<<"
"<<type<<setw(6)<<deposit<<endl;
}


int account::retacno() const
{
        return acno;
}

int account::retdeposit() const
{
        return deposit;
}

char account::rettype() const
{
        return type;
}
```

```
//*****************************************
*************** // function declaration
//*************************************************************
void write_account(); //function to write record in binary file
void display_sp(int); //function to display account details given
by user void modify_account(int); //function to modify record of
file void delete_account(int); //function to delete record of file
void display_all(); //function to display all account details void
deposit_withdraw(int, int); // function to desposit/withdraw amount
for given  account
void intro(); //introductory screen function

//*****************************************
*************** // THE MAIN FUNCTION OF PROGRAM
//*************************************************************


int main()
{
        char ch;
        int num;
        intro();
        do
        {
                system("cls");
                cout<<"\n\n\n\tMAIN MENU";
                cout<<"\n\n\t01. NEW ACCOUNT";
                cout<<"\n\n\t02. DEPOSIT AMOUNT";
                cout<<"\n\n\t03. WITHDRAW AMOUNT";
                cout<<"\n\n\t04. BALANCE ENQUIRY";
                cout<<"\n\n\t05. ALL ACCOUNT HOLDER LIST";
                cout<<"\n\n\t06. CLOSE AN ACCOUNT";
                cout<<"\n\n\t07. MODIFY AN ACCOUNT";
                cout<<"\n\n\t08. EXIT";
                cout<<"\n\n\tSelect Your Option (1-8) ";
                cin>>ch;
                system("cls");
                switch(ch)
                {
                case '1':
                        write_account();
                        break;
                case '2':
        cout<<"\n\n\tEnter The account No. : "; cin>>num;
                        deposit_withdraw(num, 1);
                        break;
                case '3':
        cout<<"\n\n\tEnter The account No. : "; cin>>num;
                        deposit_withdraw(num, 2);
                        break;
                case '4':
        cout<<"\n\n\tEnter The account No. : "; cin>>num;
                        display_sp(num);
                        break;
```

```
                case '5':
                        display_all();
                        break;
                case '6':
        cout<<"\n\n\tEnter The account No. : "; cin>>num;
                        delete_account(num);
                        break;
                 case '7':
        cout<<"\n\n\tEnter The account No. : "; cin>>num;
                        modify_account(num);
                        break;
                 case '8':
        cout<<"\n\n\tThanks for using bank managemnt system";
                        break;
                 default :cout<<"\a";
                }
                cin.ignore();
                cin.get();
        }while(ch!='8');
        return 0;
}


//***************************************************
*************** // function to write in file
//****************************************************************

void write_account()
{
        account ac;
        ofstream outFile;
        outFile.open("account.dat",ios::binary|ios::app);
        ac.create_account();
        outFile.write(reinterpret_cast<char *> (&ac),
        sizeof(account)); outFile.close();
}

//***************************************************
*************** // function to read specific
record from file
//****************************************************************

void display_sp(int n)
{
        account ac;
        bool flag=false;
        ifstream inFile;
        inFile.open("account.dat",ios::binary);
        if(!inFile)
        {
        cout<<"File could not be open !! Press any Key...";
                return;
        }
        cout<<"\nBALANCE DETAILS\n";
```

```
while(inFile.read(reinterpret_cast<char *> (&ac), sizeof(account)))
        {
                        if(ac.retacno()==n)
                        {
                                ac.show_account();
                                flag=true;
                        }
                }
                inFile.close();
                if(flag==false)
                        cout<<"\n\nAccount number does not exist";
        }


        //***********************************************
        *************** // function to modify record of
        file
        //*************************************************************

        void modify_account(int n)
        {
                bool found=false;
                account ac;
                fstream File;
                File.open("account.dat",ios::binary|ios::in|ios::out);
                if(!File)
                {
                cout<<"File could not be open !! Press any Key...";
                        return;
                }
                while(!File.eof() && found==false)
                {
                        File.read(reinterpret_cast<char *> (&ac),
                        sizeof(account)); if(ac.retacno()==n)
                        {
                                ac.show_account();
                cout<<"\n\nEnter The New Details of account"<<endl;
                                ac.modify();
                int pos=(-1)*static_cast<int>(sizeof(account));
                                File.seekp(pos,ios::cur);
                File.write(reinterpret_cast<char *> (&ac),
        sizeof(account));
                                cout<<"\n\n\t Record Updated";
                                found=true;
                         }
                }
                File.close();
                if(found==false)
                        cout<<"\n\n Record Not Found ";
        }

        //***********************************************
        *************** // function to delete record of
        file
```

```
//***************************************************************


void delete_account(int n)
{
        account ac;
        ifstream inFile;
        ofstream outFile;
        inFile.open("account.dat",ios::binary);
        if(!inFile)
        {
        cout<<"File could not be open !! Press any Key...";
                return;
        }
        outFile.open("Temp.dat",ios::binary);
        inFile.seekg(0,ios::beg);
        while(inFile.read(reinterpret_cast<char *> (&ac),
        sizeof(account))) {
                if(ac.retacno()!=n)
                {
        outFile.write(reinterpret_cast<char *> (&ac),
sizeof(account));
                }
        }
        inFile.close();
        outFile.close();
        remove("account.dat");
        rename("Temp.dat","account.dat");
        cout<<"\n\n\tRecord Deleted ..";
}

//***********************************************
*************** // function to display all
accounts deposit list
//***************************************************************

void display_all()
{
        account ac;
        ifstream inFile;
        inFile.open("account.dat",ios::binary);
        if(!inFile)
        {
        cout<<"File could not be open !! Press any Key...";
                return;
        }
        cout<<"\n\n\t\tACCOUNT HOLDER LIST\n\n";
        cout<<"====================================
        ============\n"; cout<<"A/c no. NAME Type
        Balance\n";
        cout<<"==========================================
        ========\n";
        while(inFile.read(reinterpret_cast<char *> (&ac),
        sizeof(account))) {
```

```
                ac.report();
        }
        inFile.close();
}

//***********************************************
*************** // function to deposit and
withdraw amounts
//***************************************************************

void deposit_withdraw(int n, int option)
{
        int amt;
        bool found=false;
        account ac;
        fstream File;
        File.open("account.dat", ios::binary|ios::in|ios::out);
        if(!File)
        {
        cout<<"File could not be open !! Press any Key...";
                return;
        }
        while(!File.eof() && found==false)
        {
File.read(reinterpret_cast<char *> (&ac), sizeof(account));
                if(ac.retacno()==n)
                {
                        ac.show_account();
                        if(option==1)
                        {
                cout<<"\n\n\tTO DEPOSITE AMOUNT ";
                cout<<"\n\nEnter The amount to be deposited";
                                cin>>amt;
                                ac.dep(amt);
                        }
                        if(option==2)
                        {
                cout<<"\n\n\tTO WITHDRAW AMOUNT ";
                cout<<"\n\nEnter The amount to be withdraw";
                                cin>>amt;
                int bal=ac.retdeposit()-amt;
                if((bal<500 && ac.rettype()=='S') || (bal<1000
&& ac.rettype()=='C'))
                        cout<<"Insufficience balance";
                                else
                        ac.draw(amt);
                        }
        int pos=(-1)*static_cast<int>(sizeof(ac));
                        File.seekp(pos,ios::cur);
        File.write(reinterpret_cast<char *> (&ac),
sizeof(account));
                        cout<<"\n\n\t Record Updated";
                        found=true;
         }
```

```
 }
        File.close();
        if(found==false)
                cout<<"\n\n Record Not Found ";
}


//***********************************************
*************** // INTRODUCTION FUNCTION
//***************************************************************


void intro()
{
        cout<<"\n\n\n\t BANK";
        cout<<"\n\n\tMANAGEMENT";
        cout<<"\n\n\t SYSTEM";
        cout<<"\n\n\n\nMADE BY : your name";
        cout<<"\n\nSCHOOL : your school name";
        cin.get();
}

//***********************************************
*************** // END OF PROJECT
//***************************************************************
```