OBJECT ORIENTED PROGRAMMING ASSIGNMENT # 1

CSC 210

By: Wakeel Furqan Ahmed

01-134212-192

BS(CS) - 2A

DEPARTMENT OF COMPUTER SCIENCE.
BAHRIA UNIVERSITY ISLAMABAD.
MARCH 31, 2022

Account.h:

```
#include<string>
#include<iostream>
#include<ctime>
#include<conio.h>
#include<string.h>
using namespace std;
#pragma once
class Account
{
private:
	string Account_Holder_Name;
	string Account_Num; //Account Number
	int Balance; //Total Balance in the account
	bool isActive; //True if the account is active else false
	int Minimum_Balance; //Minimum balance that must be maintained
	char Account_Type; //'S' for saving and 'C' for current
	int Account_Counter;
	char PIN[5];
public:
	Account();//Default Constructor
	Account(string, int, bool, int, char);//Parametrized Constructor
	void set();
	void set_PIN();
	void set_accountNo();
	void set_balance(int a);
	void update_balance(int a);
	void setisActive();
	void set_MinBalance();
	void set_accountType();

	void get();
	string get_AccountNo();
	string get_Account_Holder_Name();
	int get_balance();
	void get_minBalance();
	bool is_current_or_saving_accountType();
	void Deposit();
	char* getPIN();
	void Withdraw();
	bool is_Active();
	bool Suffient_Balance(int Amount);
	~Account();//Destructor
	void saving_dedution(int balance);
};	//End of class
```

```
bool transfer(Account& source, Account& destination, int amount);
void validate_Input(int& input, int accounts);
void validate_Input(int start, int& input, int accounts);
void validate_Input(int& input, int numberofaccount, Account  account[]);
```

Account.cpp:

```
#include "Account.h"

Account::Account() //Default Constructor
{
    Balance = 0; isActive = 0; Minimum_Balance = 1000; Account_Type = 'S'; Account_Counter = 0; strcpy_s(PIN, "0000");
}
Account::Account(string accountNo1, int balance1, bool isActive1, int minBalance1, char accountType1)//Parametrized Constructor
{
    Account_Num = accountNo1; Balance = balance1; isActive = isActive1; Minimum_Balance = minBalance1; Account_Type = accountType1;
strcpy_s(PIN, "0000");
}
void Account::set() {
    srand(time(0));
    cout << "Press \'C\' to Create Current Account or \'S\' to Create Saving Account " << endl
        << "Note:Current Account Minimum limit:Rs.5000/- Saving Account Minimum limit:Rs.1000/-" << endl;
    cin >> Account_Type;
    while (!(Account_Type == 's' || Account_Type == 'S' || Account_Type == 'C' || Account_Type == 'c'))
        cin >> Account_Type;
    if (is_current_or_saving_accountType())
        Minimum_Balance = 5000;
    else
        Minimum_Balance = 1000;

    cout << "AccountNo Assigned:";
    for (int i = 0; i < 10; i++)
    {
        int randomNum; char random;
        randomNum = rand() % 101;
        if (randomNum >= 50)
        {
            random = ((char)65 + rand() % 26);
            Account_Num = Account_Num + random;
            //Account_Num = Account_Num + ((char)65 + rand() % 26);
        }
        else {
            random = 48 + rand() % 10;
            Account_Num = Account_Num + random;
        }
    }
    cout << Account_Num;
    cout << "\nEnter Account Holder Name:";
    cin.ignore();
    getline(cin, Account_Holder_Name);
```

```
       cout << "Enter the Amount You Want To Add in account:";
       cin >> Balance;
       saving_dedution(Balance);
       set_PIN();
       cout << "PIN Assigned:" << PIN << endl;
}
void Account::set_PIN() {
       for (int i = 0; i < 4; i++)
              PIN[i] = 48 + rand() % 10;
       PIN[4] = '\0';
}
void Account::set_accountNo() {
       cout << "AccountNo:";
       cin >> Account_Num;
}
void Account::set_balance(int a) {
       Balance = a;
}
void Account::update_balance(int a) {
       Balance = Balance + a;
}
void Account::setisActive() {
       cout << "\nisActive:";
       cin >> isActive;
}
void Account::set_MinBalance() {
       cout << "\nMinimum Balance:";
       cin >> Minimum_Balance;
}
void Account::set_accountType() {
       cout << "\nAccountType:";
       cin >> Account_Type;
}

void Account::get()
{
       cout << "\nACCOUNT DETAILS" << endl
              << "AccountNo:" << Account_Num << endl
              << "Account Holder Name:" << Account_Holder_Name << endl
              << "Balance:Rs." << Balance << "/-" << endl;
       if (is_Active())
              cout << "Account Status:Active" << endl;
       else
              cout << "Account Status:Dormant" << endl << "Please go to Account Management to active your account" << endl;
       cout << "Minimum Balance:Rs." << Minimum_Balance << "/-" << endl;

       if (is_current_or_saving_accountType())
              cout << "Account Type:Current Account" << endl;
       else
              cout << "Account Type:Saving Account" << endl;
```

```
}
string Account::get_AccountNo() {
	return Account_Num;
}
string Account::get_Account_Holder_Name() {
	return Account_Holder_Name;
}
int Account::get_balance() {
	return Balance;
}
void Account::get_minBalance() {
	if (Account_Type == 'C' || Account_Type == 'c')
		Minimum_Balance = 5000;
	else if (Account_Type == 'S' || Account_Type == 's')
		Minimum_Balance = 1000;
	cout << "Minimum Balance:" << Minimum_Balance << endl;
}
bool Account::is_current_or_saving_accountType() {
	if (Account_Type == 'C' || Account_Type == 'c')
		return 1;
	else if (Account_Type == 'S' || Account_Type == 's')
		return 0;
}
void Account::Deposit() {
	int Deposit_Amount = 0, SUM = 0;
	if (Balance >= Minimum_Balance)
	{
		cout << "Enter Amount to Deposit:";
		cin >> Deposit_Amount;
		SUM = Deposit_Amount + get_balance();
		set_balance(SUM);
		cout << "Transaction Successful !!" << endl;
	}
	else
		cout << "Transaction Failed !!" << endl << "Account is Dormant" << endl;

}
char* Account::getPIN()
{
	return PIN;
}
void Account::Withdraw()
{
	int Withdraw_Amount;
	if (Balance >= Minimum_Balance)
	{
		cout << "Enter the Amount to Withdraw:";
		cin >> Withdraw_Amount;
		if (Balance >= Withdraw_Amount)
```

```cpp
        {
            Balance = Balance - Withdraw_Amount;
            cout << "Transaction Sucessful" << endl;
        }
        else
            cout << "Transaction failed!!" << endl << "Insuffient Balance" << endl;
    }
    else
        cout << "Transaction failed!!" << endl << "Account is Dormant" << endl;

}
bool Account::is_Active()
{
    if (Balance >= Minimum_Balance)
        return true;
    else
        return false;
}
bool Account::Suffient_Balance(int Amount)
{
    if (Balance >= Amount)
        return 1;
    else
        return 0;
}
void Account::saving_dedution(int balance) {
    if (!is_current_or_saving_accountType())
    {
        Balance = (Balance - (balance * 2.5 / 100));
        cout << "2.5% of the amount has been deducted\nNew Balance is:" << Balance << endl;
    }
}
Account::~Account() //Destructor
{
    isActive = 0; Balance = 0;
}

bool transfer(Account& source, Account& destination, int amount)
{
    if (source.is_Active())
    {
        if (source.Suffient_Balance(amount))
        {
            source.set_balance((source.get_balance() - amount));
            destination.set_balance((destination.get_balance() + amount));
            return 1;
        }
        else
            cout << "Insufficient Balance" << endl;             return 0;
    }
    else
        cout << "Inactive account" << endl;          return 0;
}
void validate_Input(int& input, int accounts) {
    while (input > accounts || input <= 0)
        cin >> input;
}
void validate_Input(int start, int& input, int accounts) {
    while (input > accounts || input < start)
        cin >> input;
}
void validate_Input(int& input, int numberofaccount, Account account[]) {
    while ((input > numberofaccount || input <= 0) || (account[input].get_AccountNo() == ""))
        cin >> input;
}
```

Driver.cpp:

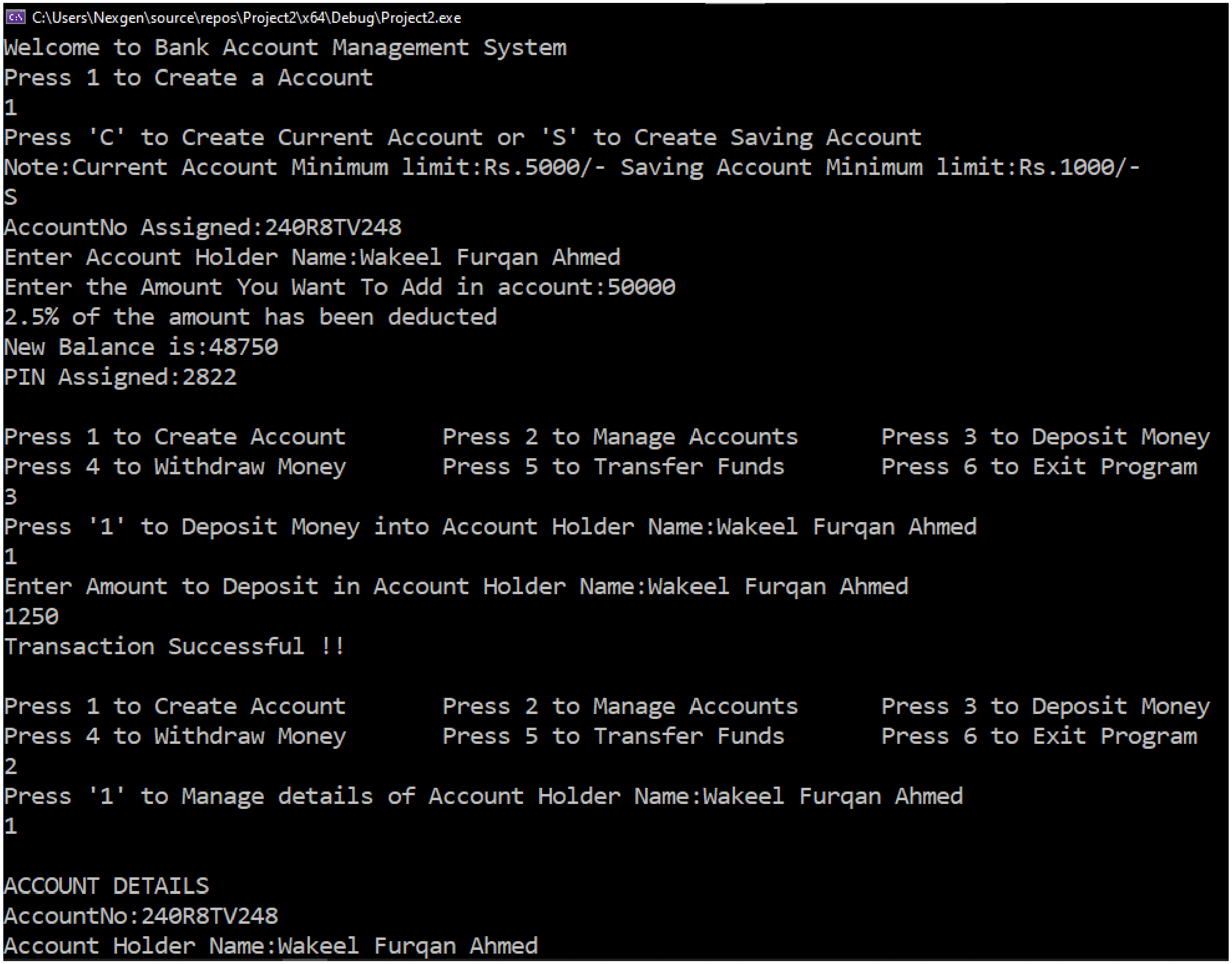
```
#include<iostream>
#include<ctime>
#include<conio.h>
#include<string>
#include<string.h>
#include"Account.h"
using namespace std;
Account A[50]; //Global (class)Account Object array.
int main()
{
    int No_of_AccountsCounter = 0, menu = 0, Input = 0, Index = 0, Amount;
    char sure;
    cout << "Welcome to Bank Account Management System" << endl;
    cout << "Press 1 to Create a Account" << endl;
    cin >> Input;
    validate_Input(Input, 1);//Input validation
    while (Input == 0 || Input == 1 || Input == 2 || Input == 3 || Input == 4 || Input == 5)//Input 6 terminates this loop.
    {
        switch (Input)
        {
        case 1: //Create Account
        {
            int k = 1;
            while (!(A[k].get_AccountNo() == ""))//Gets blank/empty account(array index) to fill & skips filled indexes.
            {
                k++;
            }
            No_of_AccountsCounter++;  //Number of Accounts counter increases.
            A[k].set();

            break;
        }//end of Case 1
```

```
case 2: //Manage Accounts
{
    for (int i = 1; i <= No_of_AccountsCounter; i++)
        if (!(A[i].get_AccountNo() == "")) //Doesn't display blank accounts(empty/no account).
            cout << "Press \'" << i << "\' to Manage details of " << A[i].get_Account_Holder_Name() <<endl;

    cin >> Index;
    validate_Input(Index,  No_of_AccountsCounter, A);   //Input validation
    A[Index].get();
    if (!(A[Index].is_Active()))
    {
        cout << "Press 0 to go to back Menu" << "\tPress 1 to active your Account";
        cin >> menu;
        validate_Input(0, menu, 1);     //Input validation
        if (menu == 1)
        {
            cout << "Desposit Amount to make Account Active:";
            cin >> Amount;
            A[Index].update_balance(Amount);
        }
    }
    else
    {
        cout << "Press 0 to go to back Menu" << "\tPress 1 to reset PIN" << "\tPress 2 to Close Account";
        cin >> menu;
        validate_Input(0, menu, 2);     //Input validation
        if (menu == 1)//Reset PIN
        {
            A[Index].set_PIN();
            cout << "New PIN is:" << A[Index].getPIN() << endl;
        }
        if (menu == 2) //Close Account
        {
            cout << "Are you sure you want to close your account: Press Y to proceed\\N to stop:";
            cin >> sure;
            if (sure == 'Y' || sure == 'y') //Confirmation to close Account
            {
                A[Index].~Account();     //To Delete an account.
            }break;
        }
    }
    break;
}//end of Case 2
case 3: //Deposit Money
{
    for (int j = 1; j <= No_of_AccountsCounter; j++)
        if (!(A[j].get_AccountNo() == ""))    //Doesn't display blank accounts(empty/no account).
            cout << "Press \'" << j << "\' to Deposit Money into " << A[j].get_Account_Holder_Name() << endl;

    cin >> Index;
```

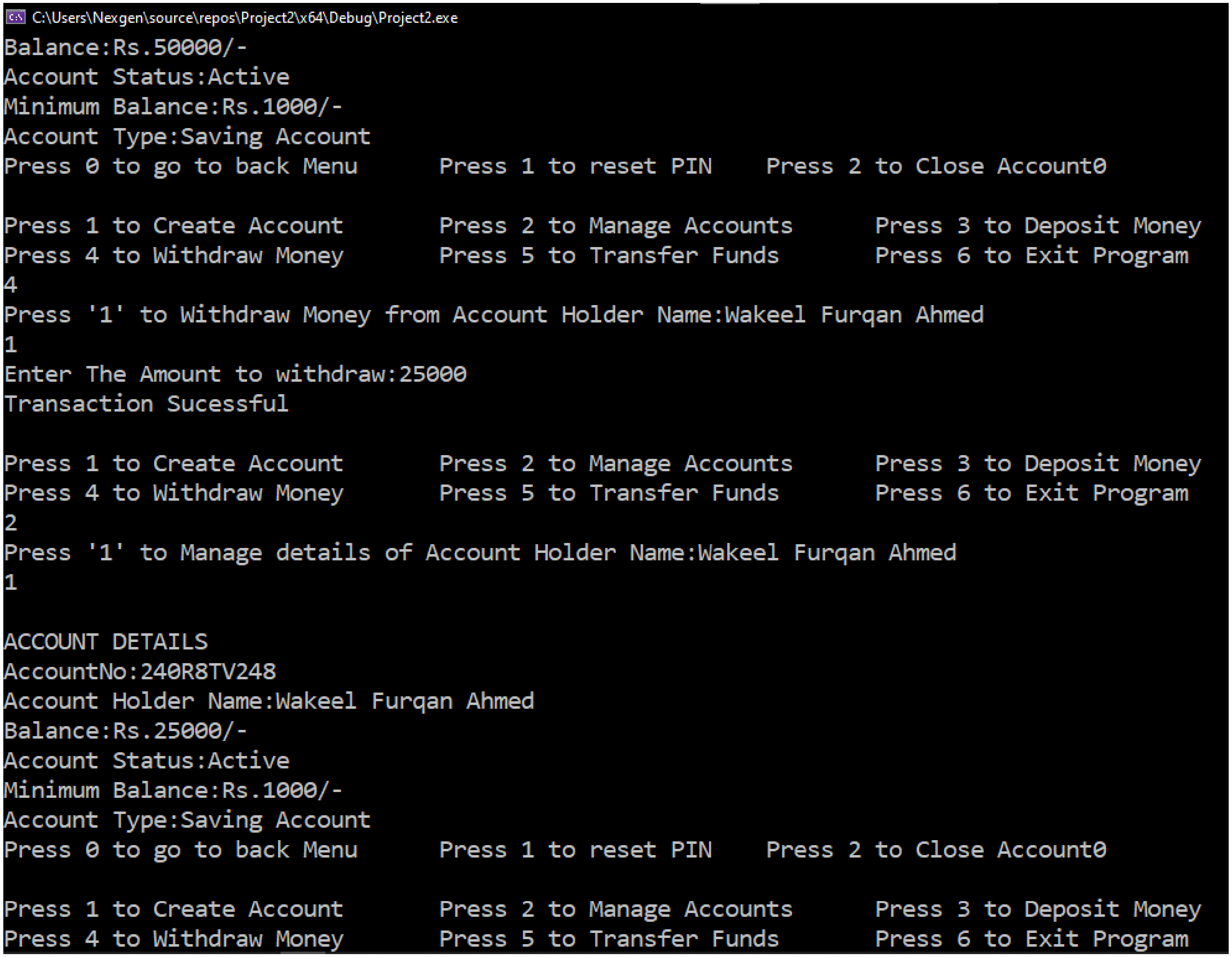
```
        validate_Input(Index, No_of_AccountsCounter, A);
        A[Index].Deposit();
        break;
}//end of Case 3
case 4:         //Withdraw Money
{
        for (int j = 1; j <= No_of_AccountsCounter; j++)
                if (!(A[j].get_AccountNo() == "")) //Doesn't display blank accounts(empty/no account).
                        cout << "Press \'" << j << "\' to Withdraw Money from " << A[j].get_Account_Holder_Name() << endl;

        cin >> Index;
        validate_Input(Index, No_of_AccountsCounter, A);    //Input validation
        A[Index].Withdraw();
        break;
}
case 5:         //Transfer Money
{       int Source_Index = 0, destination_Index = 0;
for (int j = 1; j <= No_of_AccountsCounter; j++)
        if (!(A[j].get_AccountNo() == ""))      //Doesn't display blank accounts(empty/no account).
                cout << "Press \'" << j << "\' to Select Account " << A[j].get_Account_Holder_Name() << endl;

cout << "Select Account to Transfer Funds FROM:";
cin >> Source_Index;
validate_Input(Source_Index, No_of_AccountsCounter, A);    //Input validation
cout << "Enter the 4-digit PIN to perform Transaction:";
char PIN[5];
for (int i = 0; i < 4; i++)
{
        PIN[i] = _getch();
        cout << "*";
}PIN[4] = '\0'; //User Entering PIN
if (!(strcmp(A[Source_Index].getPIN(), PIN)))//PIN Matching
{
        cout << "\nSelect Account to Transfer Amount TO:";
        cin >> destination_Index;
        validate_Input(destination_Index, No_of_AccountsCounter, A);//Input validation
        cout << "Enter Amount to Transfer:";
        cin >> Amount;
        if (transfer(A[Source_Index], A[destination_Index], Amount)) //transfer() function returns bool(1/0).
                cout << "Transaction is Sucessful";
        else
                cout << "Transaction is unSuccessful\n" << endl;
}
else
        cout << "\nInvalid PIN";

break;
}
}//End of switch Statement.
cout << "\nPress 1 to Create Account\t" << "Press 2 to Manage Accounts\t" << "Press 3 to Deposit Money" << endl
```

```
            << "Press 4 to Withdraw Money\t" << "Press 5 to Transfer Funds\t" << "Press 6 to Exit Program" << endl;
        cin >> Input;
        validate_Input(Input, 6);//Input validation
    }//End of While Loop controlling switch Statement.
    exit(0);    //Indicates successful termination of program.
}
```

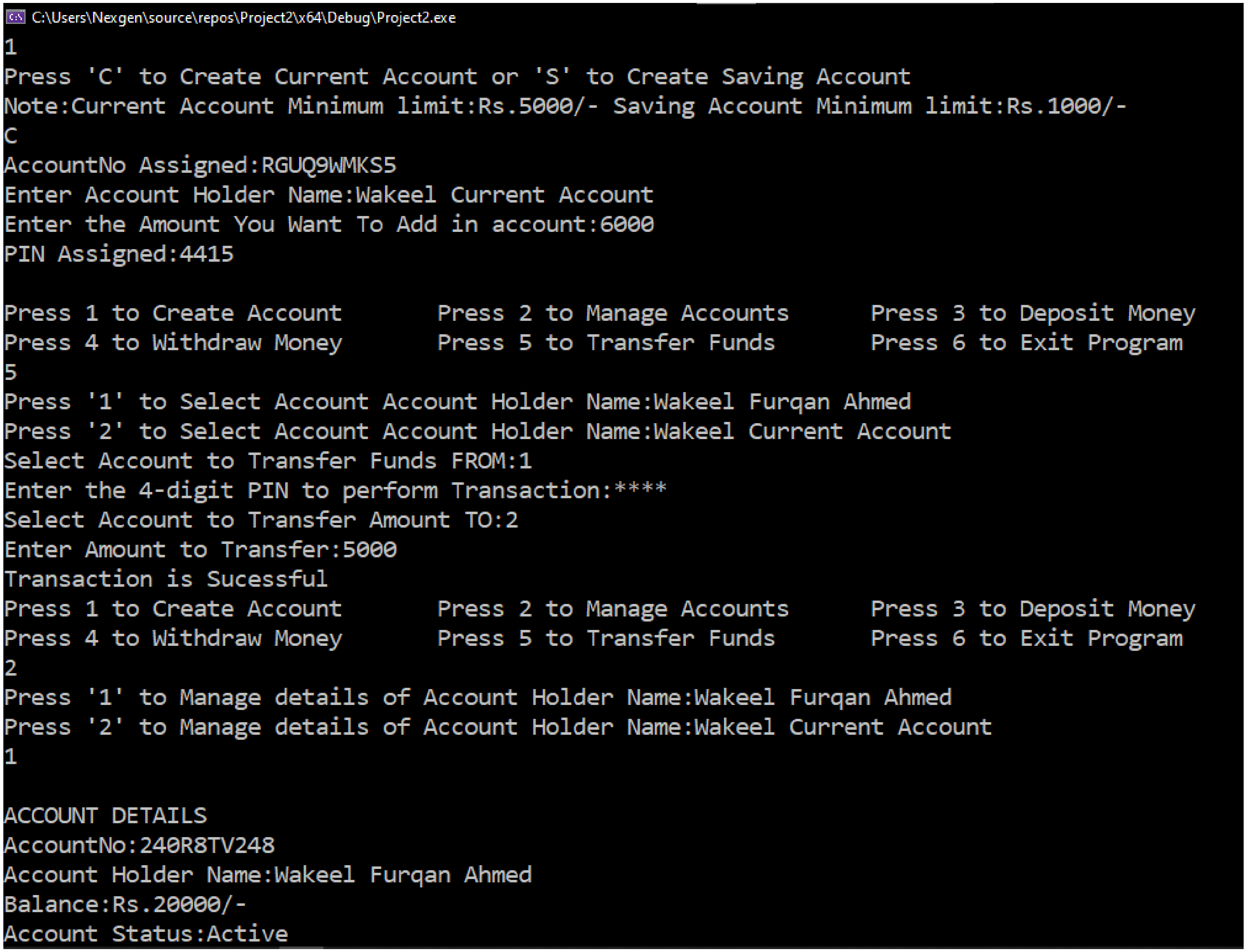
Screen Shots of Program Working:


C:\Users\Nexgen\source\repos\Project2\x64\Debug\Project2.exe

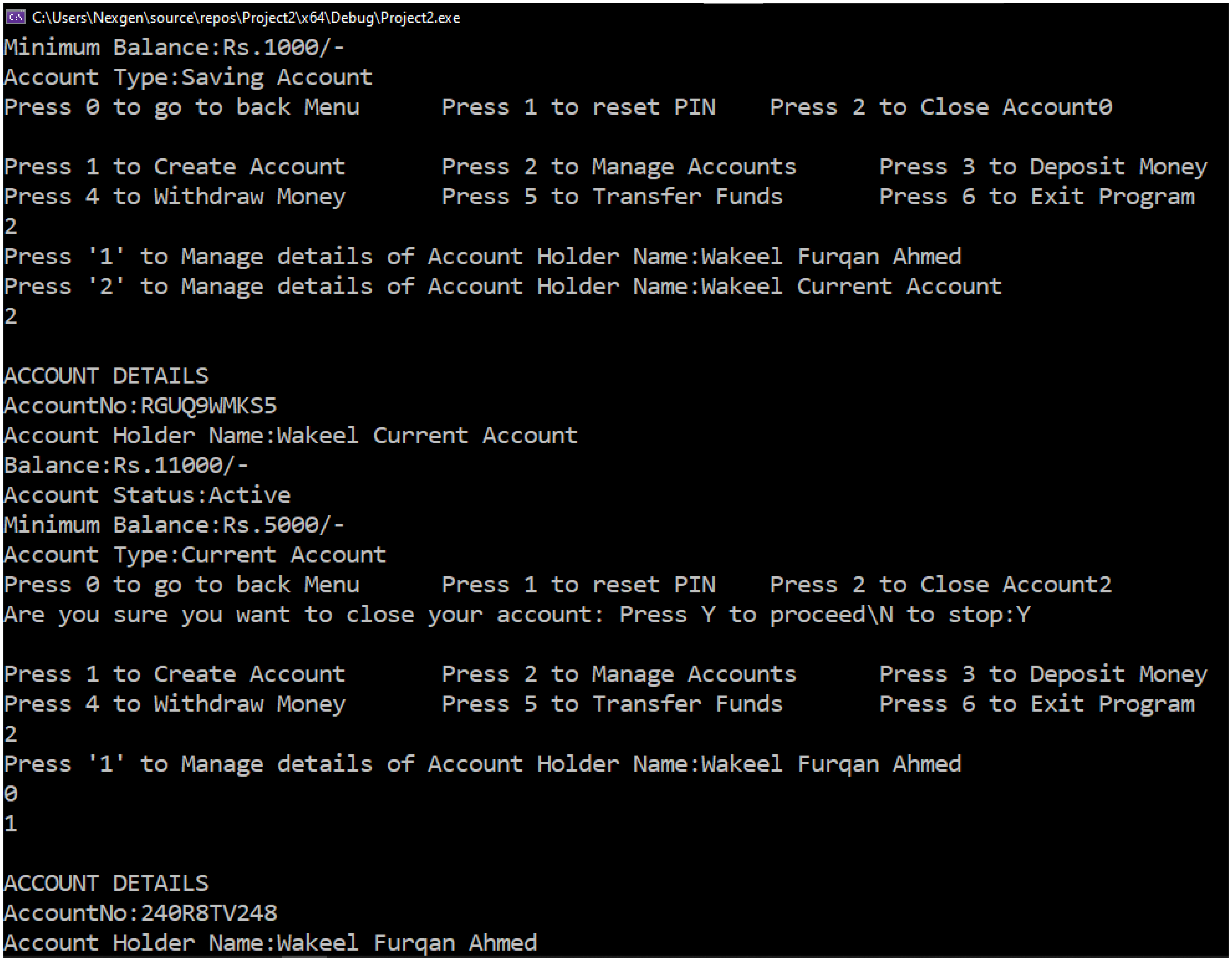
```
Welcome to Bank Account Management System
Press 1 to Create a Account
1
Press 'C' to Create Current Account or 'S' to Create Saving Account
Note:Current Account Minimum limit:Rs.5000/- Saving Account Minimum limit:Rs.1000/-
S
AccountNo Assigned:240R8TV248
Enter Account Holder Name:Wakeel Furqan Ahmed
Enter the Amount You Want To Add in account:50000
2.5% of the amount has been deducted
New Balance is:48750
PIN Assigned:2822

Press 1 to Create Account       Press 2 to Manage Accounts      Press 3 to Deposit Money
Press 4 to Withdraw Money       Press 5 to Transfer Funds       Press 6 to Exit Program
3
Press '1' to Deposit Money into Account Holder Name:Wakeel Furqan Ahmed
1
Enter Amount to Deposit in Account Holder Name:Wakeel Furqan Ahmed
1250
Transaction Successful !!

Press 1 to Create Account       Press 2 to Manage Accounts      Press 3 to Deposit Money
Press 4 to Withdraw Money       Press 5 to Transfer Funds       Press 6 to Exit Program
2
Press '1' to Manage details of Account Holder Name:Wakeel Furqan Ahmed
1

ACCOUNT DETAILS
AccountNo:240R8TV248
Account Holder Name:Wakeel Furqan Ahmed
```


C:\Users\Nexgen\source\repos\Project2\x64\Debug\Project2.exe

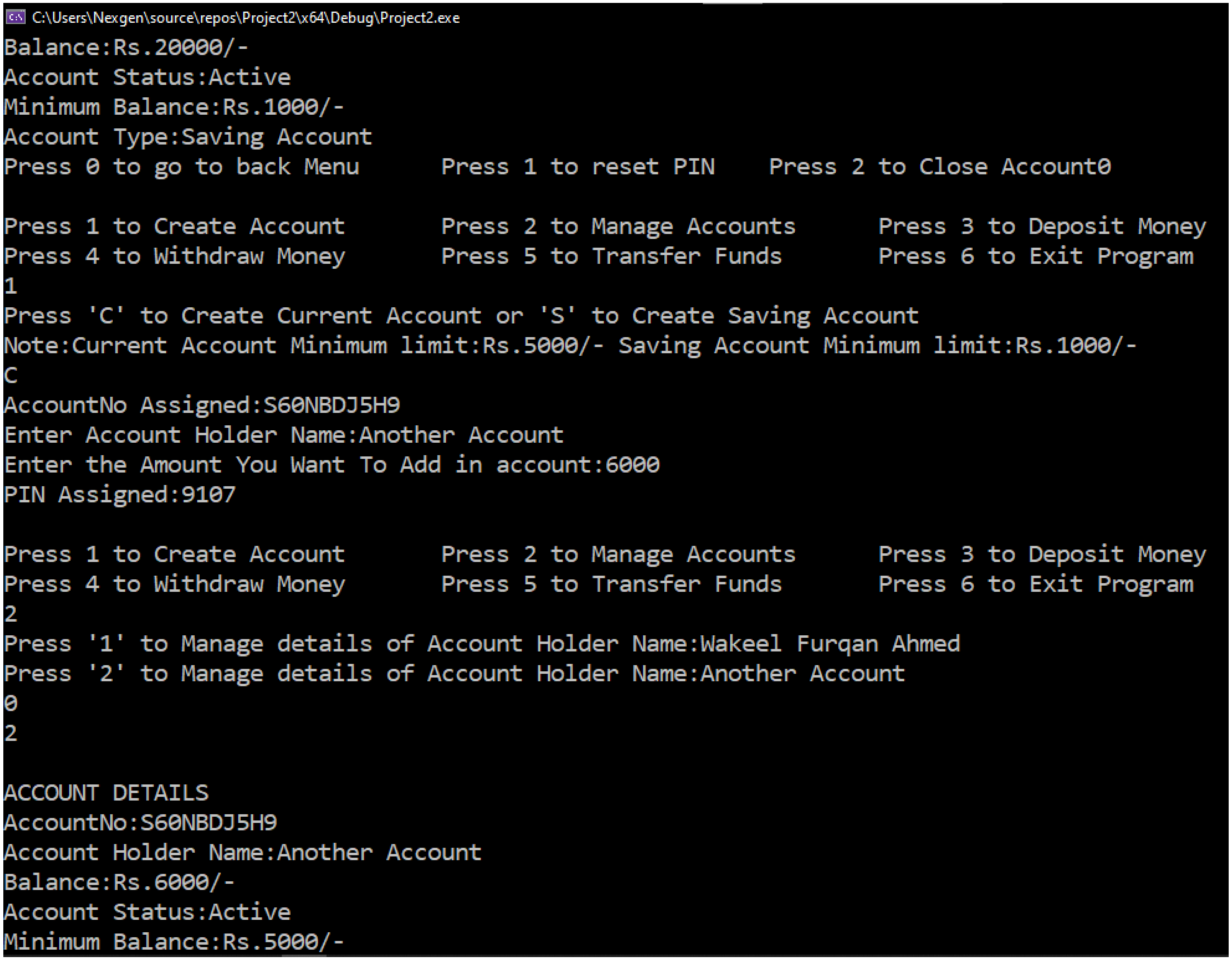
```
Balance:Rs.50000/-
Account Status:Active
Minimum Balance:Rs.1000/-
Account Type:Saving Account
Press 0 to go to back Menu      Press 1 to reset PIN    Press 2 to Close Account0

Press 1 to Create Account       Press 2 to Manage Accounts      Press 3 to Deposit Money
Press 4 to Withdraw Money       Press 5 to Transfer Funds       Press 6 to Exit Program
4
Press '1' to Withdraw Money from Account Holder Name:Wakeel Furqan Ahmed
1
Enter The Amount to withdraw:25000
Transaction Sucessful

Press 1 to Create Account       Press 2 to Manage Accounts      Press 3 to Deposit Money
Press 4 to Withdraw Money       Press 5 to Transfer Funds       Press 6 to Exit Program
2
Press '1' to Manage details of Account Holder Name:Wakeel Furqan Ahmed
1

ACCOUNT DETAILS
AccountNo:240R8TV248
Account Holder Name:Wakeel Furqan Ahmed
Balance:Rs.25000/-
Account Status:Active
Minimum Balance:Rs.1000/-
Account Type:Saving Account
Press 0 to go to back Menu      Press 1 to reset PIN    Press 2 to Close Account0

Press 1 to Create Account       Press 2 to Manage Accounts      Press 3 to Deposit Money
Press 4 to Withdraw Money       Press 5 to Transfer Funds       Press 6 to Exit Program
```


C:\Users\Nexgen\source\repos\Project2\x64\Debug\Project2.exe

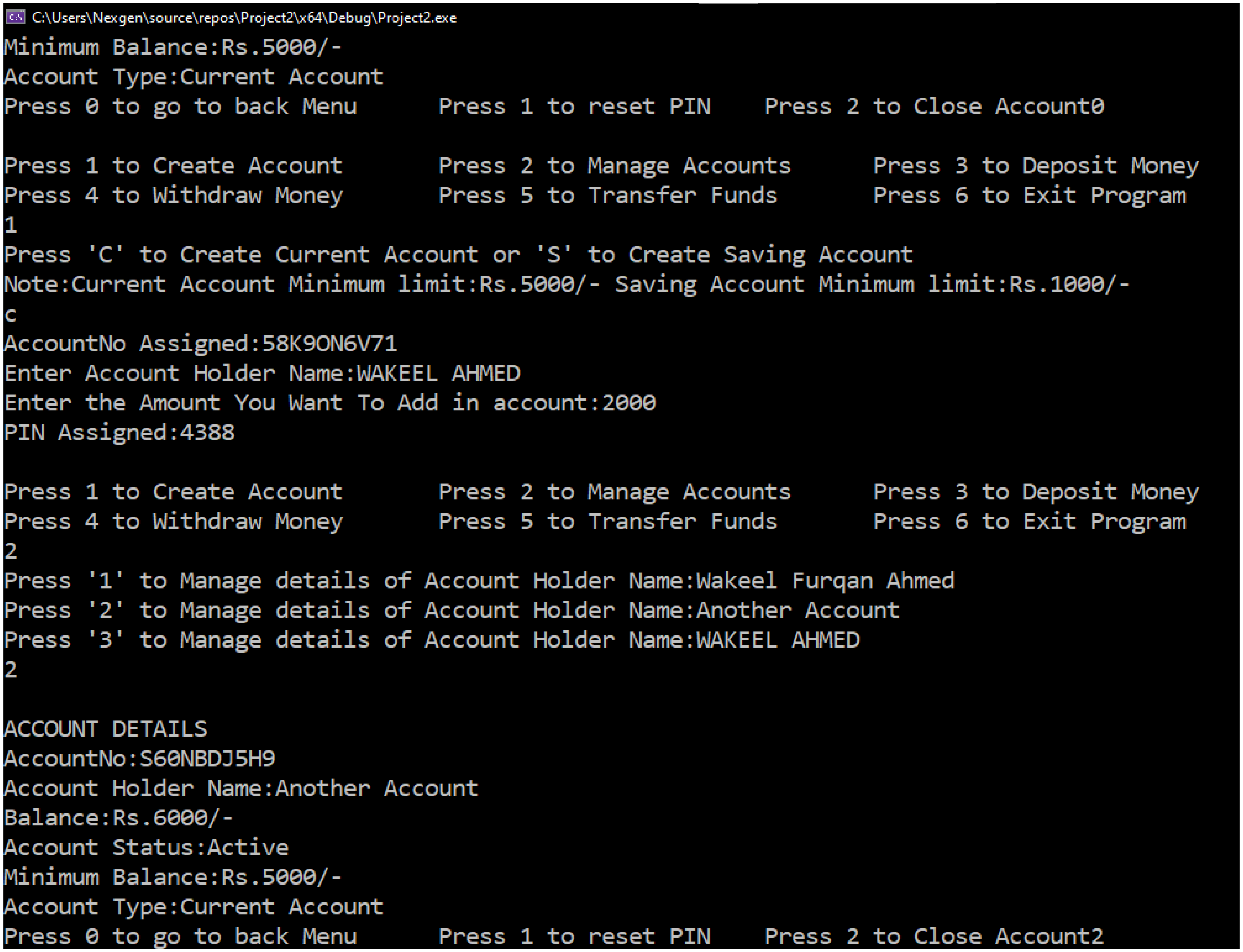
```
1
Press 'C' to Create Current Account or 'S' to Create Saving Account
Note:Current Account Minimum limit:Rs.5000/- Saving Account Minimum limit:Rs.1000/-
C
AccountNo Assigned:RGUQ9WMKS5
Enter Account Holder Name:Wakeel Current Account
Enter the Amount You Want To Add in account:6000
PIN Assigned:4415

Press 1 to Create Account       Press 2 to Manage Accounts      Press 3 to Deposit Money
Press 4 to Withdraw Money       Press 5 to Transfer Funds       Press 6 to Exit Program
5
Press '1' to Select Account Account Holder Name:Wakeel Furqan Ahmed
Press '2' to Select Account Account Holder Name:Wakeel Current Account
Select Account to Transfer Funds FROM:1
Enter the 4-digit PIN to perform Transaction:****
Select Account to Transfer Amount TO:2
Enter Amount to Transfer:5000
Transaction is Sucessful
Press 1 to Create Account       Press 2 to Manage Accounts      Press 3 to Deposit Money
Press 4 to Withdraw Money       Press 5 to Transfer Funds       Press 6 to Exit Program
2
Press '1' to Manage details of Account Holder Name:Wakeel Furqan Ahmed
Press '2' to Manage details of Account Holder Name:Wakeel Current Account
1

ACCOUNT DETAILS
AccountNo:240R8TV248
Account Holder Name:Wakeel Furqan Ahmed
Balance:Rs.20000/-
Account Status:Active
```


C:\Users\Nexgen\source\repos\Project2\x64\Debug\Project2.exe

```
Minimum Balance:Rs.1000/-
Account Type:Saving Account
Press 0 to go to back Menu      Press 1 to reset PIN    Press 2 to Close Account0

Press 1 to Create Account       Press 2 to Manage Accounts      Press 3 to Deposit Money
Press 4 to Withdraw Money       Press 5 to Transfer Funds       Press 6 to Exit Program
2
Press '1' to Manage details of Account Holder Name:Wakeel Furqan Ahmed
Press '2' to Manage details of Account Holder Name:Wakeel Current Account
2

ACCOUNT DETAILS
AccountNo:RGUQ9WMKS5
Account Holder Name:Wakeel Current Account
Balance:Rs.11000/-
Account Status:Active
Minimum Balance:Rs.5000/-
Account Type:Current Account
Press 0 to go to back Menu      Press 1 to reset PIN    Press 2 to Close Account2
Are you sure you want to close your account: Press Y to proceed\N to stop:Y

Press 1 to Create Account       Press 2 to Manage Accounts      Press 3 to Deposit Money
Press 4 to Withdraw Money       Press 5 to Transfer Funds       Press 6 to Exit Program
2
Press '1' to Manage details of Account Holder Name:Wakeel Furqan Ahmed
0
1

ACCOUNT DETAILS
AccountNo:240R8TV248
Account Holder Name:Wakeel Furqan Ahmed
```

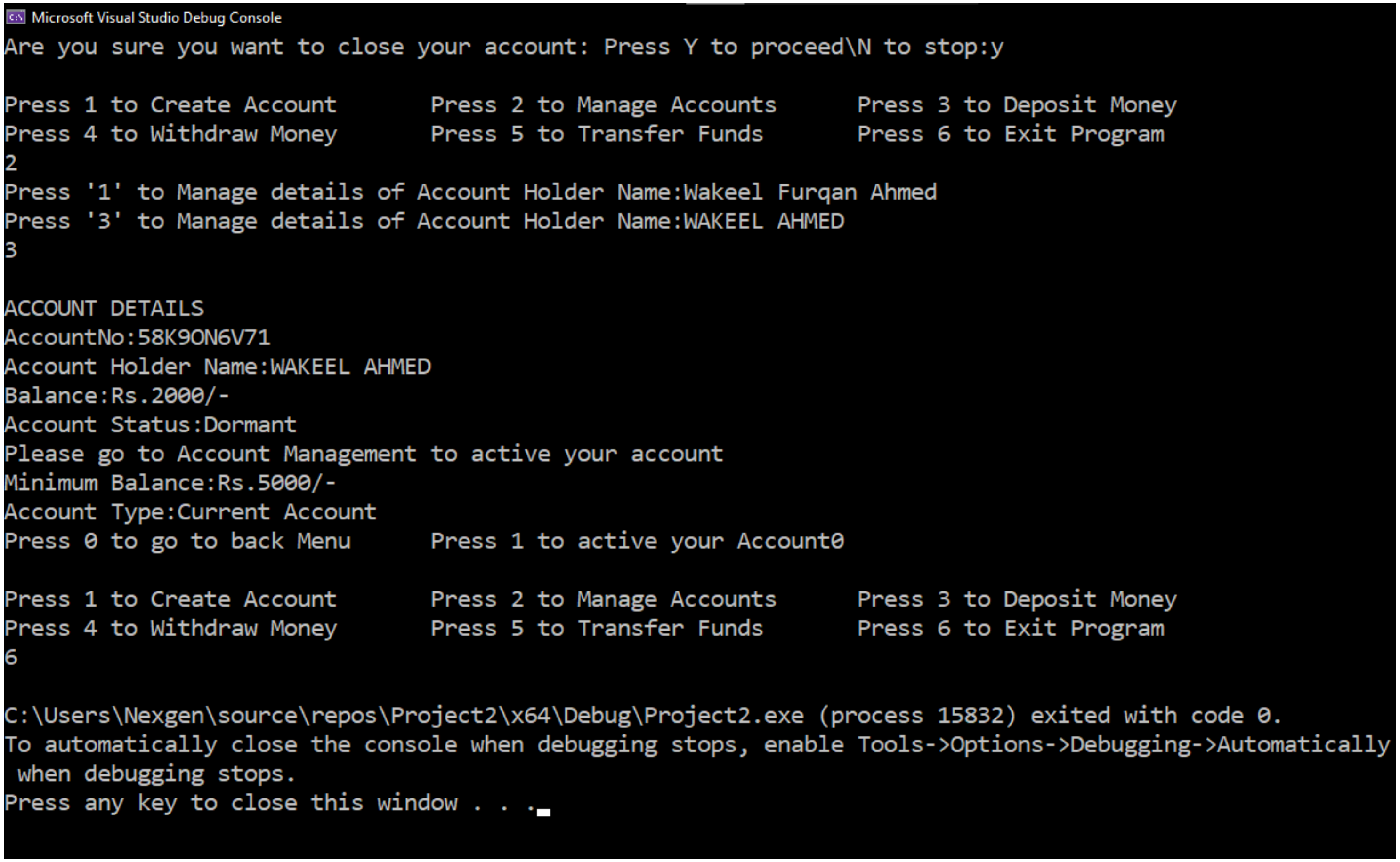

C:\Users\Nexgen\source\repos\Project2\x64\Debug\Project2.exe

```
Balance:Rs.20000/-
Account Status:Active
Minimum Balance:Rs.1000/-
Account Type:Saving Account
Press 0 to go to back Menu      Press 1 to reset PIN    Press 2 to Close Account0

Press 1 to Create Account       Press 2 to Manage Accounts      Press 3 to Deposit Money
Press 4 to Withdraw Money       Press 5 to Transfer Funds       Press 6 to Exit Program
1
Press 'C' to Create Current Account or 'S' to Create Saving Account
Note:Current Account Minimum limit:Rs.5000/- Saving Account Minimum limit:Rs.1000/-
C
AccountNo Assigned:S60NBDJ5H9
Enter Account Holder Name:Another Account
Enter the Amount You Want To Add in account:6000
PIN Assigned:9107

Press 1 to Create Account       Press 2 to Manage Accounts      Press 3 to Deposit Money
Press 4 to Withdraw Money       Press 5 to Transfer Funds       Press 6 to Exit Program
2
Press '1' to Manage details of Account Holder Name:Wakeel Furqan Ahmed
Press '2' to Manage details of Account Holder Name:Another Account
0
2

ACCOUNT DETAILS
AccountNo:S60NBDJ5H9
Account Holder Name:Another Account
Balance:Rs.6000/-
Account Status:Active
Minimum Balance:Rs.5000/-
```



C:\Users\Nexgen\source\repos\Project2\x64\Debug\Project2.exe

```
Minimum Balance:Rs.5000/-
Account Type:Current Account
Press 0 to go to back Menu      Press 1 to reset PIN    Press 2 to Close Account0

Press 1 to Create Account       Press 2 to Manage Accounts      Press 3 to Deposit Money
Press 4 to Withdraw Money       Press 5 to Transfer Funds       Press 6 to Exit Program
1
Press 'C' to Create Current Account or 'S' to Create Saving Account
Note:Current Account Minimum limit:Rs.5000/- Saving Account Minimum limit:Rs.1000/-
c
AccountNo Assigned:58K9ON6V71
Enter Account Holder Name:WAKEEL AHMED
Enter the Amount You Want To Add in account:2000
PIN Assigned:4388

Press 1 to Create Account       Press 2 to Manage Accounts      Press 3 to Deposit Money
Press 4 to Withdraw Money       Press 5 to Transfer Funds       Press 6 to Exit Program
2
Press '1' to Manage details of Account Holder Name:Wakeel Furqan Ahmed
Press '2' to Manage details of Account Holder Name:Another Account
Press '3' to Manage details of Account Holder Name:WAKEEL AHMED
2

ACCOUNT DETAILS
AccountNo:S60NBDJ5H9
Account Holder Name:Another Account
Balance:Rs.6000/-
Account Status:Active
Minimum Balance:Rs.5000/-
Account Type:Current Account
Press 0 to go to back Menu      Press 1 to reset PIN    Press 2 to Close Account2
```



```
Are you sure you want to close your account: Press Y to proceed\N to stop:y

Press 1 to Create Account       Press 2 to Manage Accounts      Press 3 to Deposit Money
Press 4 to Withdraw Money       Press 5 to Transfer Funds       Press 6 to Exit Program
2
Press '1' to Manage details of Account Holder Name:Wakeel Furqan Ahmed
Press '3' to Manage details of Account Holder Name:WAKEEL AHMED
3

ACCOUNT DETAILS
AccountNo:58K9ON6V71
Account Holder Name:WAKEEL AHMED
Balance:Rs.2000/-
Account Status:Dormant
Please go to Account Management to active your account
Minimum Balance:Rs.5000/-
Account Type:Current Account
Press 0 to go to back Menu      Press 1 to active your Account0

Press 1 to Create Account       Press 2 to Manage Accounts      Press 3 to Deposit Money
Press 4 to Withdraw Money       Press 5 to Transfer Funds       Press 6 to Exit Program
6

C:\Users\Nexgen\source\repos\Project2\x64\Debug\Project2.exe (process 15832) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically
 when debugging stops.
Press any key to close this window . . .
```