

PROGRAMMING FUNDAMENTALS

SEMESTER PROJECT

DECEMBER 25, 2023

FACULTY OF COMPUTING
I-14 CAMPUS, ISLAMABAD

BANK MANAGEMENT SYSTEM

Name: ABDUL RAFAY
Section: SE 1-1
SAP-ID: 54689
Course: Programming Fundamentals
(LAB)
Submitted to: Sir Hassan Ali Shah

```

#include <iostream>
#include <time.h>
#include <string>
#include <unistd.h>

using namespace std;

const int max_accounts = 100;
int values=0;
string holdertype[max_accounts], accounttype[max_accounts],
accountno[max_accounts];
double balance[max_accounts];

void screen()
{
    system("cls");
    cout<<"\n\n\n\t\t\t\t\t***----BANK----***\n"<<endl;
    sleep(1);
    cout<<"\t\t\t\t\t\b\b\b***----MANAGEMENT----***\n"<<endl;
    sleep(1);
    cout<<"\t\t\t\t\t\b***----SYSTEM----***\n"<<endl;
    sleep(1);
    cout<<"\n\n\t\t\t\t\t\b\b\b----[ Project by: ]----"<<endl;
    cout<<"\t\t\t\t\t\b\b\b|| 54689-ABDUL RAFAY ||"<<endl;
    cout<<"\t\t\t\t\t\b\b\b|| \b54481-HASSAN ZAHID ||"<<endl;
    cout<<"\t\t\t\t\t\b\b\b|| 54791-MOIZ ASIF ||"<<endl;
    cout<<"\t\t\t\t\t\b\b\b|| \b55244-KAMRAN MAJEED||"<<endl;
    cout<<"\t\t\t\t\t\b\b\b|| 53143-AHSAN ABASS ||\a\n\t\t\t\t\t\b\b\b-----
-----"<<endl;
    sleep(3);
}

int start_screen()
{
    int choice=0;

```

[illegible]

```

if (execution!=1)
{
    cout<<"\n\t\tEnter Account Holder's Name: ";
    execution=0;
    getline(cin, holdername[values]);
    cout<<"\n\t\tEnter type of Account ('C' for Current/'S' for Saving): ";
    getline(cin, accounttype[values]);
    for(int i=0; i<1; i++)
        if(tolower(accounttype[values][i])=='c')
        {
            cout<<"\n\t\tEnter initial amount in current account: ";
            cin>>balance[values];
        }
        else if(tolower(accounttype[values][0])=='s')
        {
            cout<<"\n\t\tEnter initial amount in saving account: ";
            cin>>balance[values];
        }
        else
        {
            cout<<"\n\t\tInvalid account type selected"<<endl;
            execution=1;
            sleep(2);
            break;
        }
    if(execution!=1)
    {
        values++;
        cout<<"\n\t\tCreating Account....";
        sleep(2);
        cout<<"\n\n\t\tAccount Created Successfully ..... \n\n\t\t\t-----
-----"<<endl;

        sleep(2);
    }
}

```

```

}

int account_details()
{
    system("cls");
    int i=0, verified=-1;
    string savenumber = {0};
    do
    {
        cout<<"\n\n\t\tEnter account number: ";
        cin>>savenumber;
        for(i=0; i<values; i++)
            if(savenumber == accountno[i])
            {
                verified=i;
                cout<<"\n\t\t\tDETAILS:\n\t\t\tAccount Number:
"<<accountno[verified]<<"\n\t\t\tAccount Holder Name:
"<<holdername[verified]<<"\n\t\t\tYour Balance is "<<balance[verified]<<"
RS"<<"\n\t\t\tYour Account Type is "<<accounttype[verified];
            }
            if (verified==-1)
                cout<<"\n\t\t\t\tNo Record Found.";
            sleep(3);
        } while(verified==-1);
        return verified;
    }

void account_withdraw()
{
    int verify = account_details();
    double *total = &balance[verify], amount=0;
    cout<<"\n\n\t\tEnter amount to withdraw: ";
    cin>>amount;
    if(*total>=amount)
    {

```

```

        cout<<"\t\t\tProcessing....";
        sleep(2);
        *total-=amount;
        cout<<"\n\t\tAmount withdrawl succesfull....\n\n\t\t\t-----
-----"<<endl;
        sleep(2);
        account_details();
    }
    else
    {
        cout<<"\n\t\tYour entered amount is greater than your current amount that
is: "<<*total<<endl;
        cout<<"\t\t\t\tTry valid amount";
        sleep(4);
    }
}

void account_deposit()
{
    int verify = account_details();
    double *total = &balance[verify],amount=0;
    cout<<"\n\n\t\tEnter amount to deposit: ";
    cin>>amount;
    if(amount>0)
    {
        cout<<"\t\t\tProcessing....";
        sleep(2);
        *total+=amount;
        cout<<"\n\t\t\tAmount deposited succesfully....";
        sleep(2);
        account_details();
    }
}

void account_insurance()

```

```

{
    system("cls");
    int age;
    char health, location, sex;
    cout<<"\n\nINSURANCE POLICY: \n\n\t\tEnter your health P(poor) & E(excellent):
";
    cin>>health;
    cout<<"\n\t\tEnter your age but your age must be (b/w 25 to 35): ";
    cin>>age;
    cout<<"\n\t\tEnter your location location C(city) or V(villge): ";
    cin>>location;
    cout<<"\n\t\tEnter your gender M(male)& F(female): ";
    cin>>sex;
    if((health=='e')&&(age>=25&&age<=35)&&(location=='c')&&(sex=='m'))
        cout<<"\n\tYou will get premium Rs. 4 per thousand and policy amount
cannot exceed 2 lakhs";
    else if((health=='e')&&(age>=25&&age<=35)&&(location=='c')&&(sex=='f'))
        cout<<"\n\tYou will get premium Rs. 3 per thousand and policy amount
cannot exceed 1 lakhs";
    else if((health=='p')&&(age>=25&&age<=35)&&(location=='v')&&(sex=='m'))
        cout<<"\n\tYou will get premium Rs.6 per thousand and policy amount cannot
exceed 10,000 Rupees";
    else
        cout<<"\n\t\t\b\bYou will not get any insurance accoding to bank
policies.";
    sleep(4);
}

void account_moneytransfer()
{
    system("cls");
    string sender, receiver;
    double amount;
    int senderIndex = -1, receiverIndex = -1;
    cout<<"\n\n\t\tEnter sender account number: ";

```

```

cin>>sender;
cout<<"\n\t\tEnter the receiver's account number: ";
cin>>receiver;
cout<<"\n\t\tEnter the amount to transfer: ";
cin>>amount;
for(int i = 0; i < values; i++)
{
    if(accountno[i] == sender)
        senderIndex = i;
    if(accountno[i] == receiver)
        receiverIndex = i;
}
if(senderIndex != -1 && receiverIndex != -1)
{
    if(balance[senderIndex] >= amount) {
        balance[senderIndex] -= amount;
        balance[receiverIndex] += amount;
        cout<<"\t\t\tProcessing....";
        sleep(2);
        cout << "\n\t\t\tTransfer successful.";
        sleep(2);
        cout<<"\n\t\t\tSender's current new balance is: "
<<balance[senderIndex]<<"\n\t\t\tReceiver's current balance is
"<<balance[receiverIndex];
    }
    else
    {
        cout << "\n\t\t\tInsufficient balance.";
    }
}
else
    cout << "\n\t\t\tInvalid sender or receiver account number.";
sleep(3);
}

```



```

void account_modification()
{
    int verify = account_details();
    cout<<endl<<"\n\t\tEnter New Holders name: ";
    cin.ignore();
    getline(cin, holdername[verify]);
    cout<<"\t\tEnter New Account Type ('C' Current/'S' Saving): ";
    cin>> accounttype[verify];
    cout<<"\n\t\t\tProcessing....";
    sleep(2);
    cout<<"\n\t\t\tAccount Modified succesfully....";
    sleep(2);
}

void account_closure()
{
    system("cls");
    cout<<"\n\n\n\n\t\t\tNOTE: Kindly withdraw you balance.\n\t\t\t\tOtherwise, no
return is possible.";
    sleep(4);
    int verify = account_details();
    char choice = 0;
    cout<<"\n\n\t\t\tDo you want to delete your account (Y/N): ";
    cin>>choice;
    if(tolower(choice)=='y')
    {
        accountno[verify]="Account Removed";
        accounttype[verify]="Account Removed";
        holdername[verify]="Holder Data Removed";
        balance[verify]=0;
        cout << "\n\n\t\t\tProcessing....";
        sleep(2);
        cout << "\n\t\t\tAccount closed successfully....";
    }
    else

```

```

        cout<<"\n\t\t\t\bAccount not deleted...";
        sleep(2);
    }

void accounts_list()
{
    system("cls");
    for(int i=0; i<values; i++)
    {
        cout<<"\n\t\t\t\tDETAILS:\n\t\t\tAccount Number:
"<<accountno[i]<<"\n\t\t\tAccount Holder's Name: "<<holdername[i]<<"\n\t\t\tBalance is
"<<balance[i]<<" RS"<<"\n\t\t\tAccount's Type is "<<accounttype[i]<<endl;
        sleep(1);
    }
    sleep(6);
}

void account_exit()
{
    system("cls");
    cout<<"\n\n\n\n\n\t\t\t\t\t\b\b-----
\n\t\t\t\t\t\b\b\b||\t\t\tGOOD BYE!!\t ||\n\t\t\t\t\t\b\b\b||  HOPE TO SEE YOU
AGAIN!  ||\a\n\t\t\t\t\t\b\b-----\n\n\n\n"<<endl;
    sleep(3);
}

int main()
{
    screen();
    int choice = 0;
    do {
        choice = start_screen();
        switch(choice) {
            case 1:
                new_account();

```

```

        break;
    case 2:
        account_withdraw();
        break;
    case 3:
        account_deposit();
        break;
    case 4:
        account_details();
        break;
    case 5:
        account_insurance();
        break;
    case 6:
        account_moneytransfer();
        break;
    case 7:
        account_modification();
        break;
    case 8:
        account_closure();
        break;
    case 9:
        accounts_list();
        break;
    case 10:
        account_exit();
        break;
    default:
        cout<<"\n\t\t\tEnter valid option from Menu";
        sleep(2);
        break;
}
} while(choice!=10);
}
//--@THE END@--

```