Bank management system

**Presented by**

***Arshiya Saleem***

***Izwa Afzal***

***Hala Ali khan***

***Submitted to:Ma’am Sobia Khalid***

// UHJMBB.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

#include<string>

using namespace std;

struct account

{

string name;

int CustAc;

double Balance;

};

account per1;

//Function Declaration

void deposit (void);

void withdrawl (void);

void F\_Balance (void);

int \_tmain(int argc, \_TCHAR\* argv[])

{

int i; //Selecting option

char A; //Permission

int o; //Account number

per1.name = "client";

per1.Balance = 3000;

per1.CustAc = 1011002;

cout << "Please enter your account number "<< endl;

L2: //goto label

cin >> o ;

if (o == per1.CustAc)

{

cout<<""<<endl;

cout << "\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*WELCOME TO BANK MANAGEMENT SYSTEM\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* " << endl;

L1: //goto label

cout<<"\tplease choose:\n\n \t1- Deposit. 2- Withdrawl. 3- Check your balance. : ";

cin>>i;

switch (i) // function calling

{

case(1):

deposit();

break;

case(2):

withdrawl();

break;

case(3):

F\_Balance();

break;

default:

cout<<" wrong entery, please try again\n";

goto L1;

}

}

else

{

cout << " please enter a valid account number "<< endl;

goto L2;

}

cout<<"Do you like to continue? [Y / N] ";

cin>>A;

if ((A=='Y') || (A=='y'))

{

goto L1;

}

cout<<"Thank you\n";

system("pause");

return 0;

}

//function definition

void deposit (void)

{

int k;

cout<< "please enter the value to deposit" << endl;

cin >> k;

cout << " your balance now is: " << per1.Balance + k << endl;

per1.Balance= per1.Balance+k; //updating previous value

}

void withdrawl (void)

{

int w;

cout<< "please enter the value to withdrawl" << endl;

cin >> w;

cout << " your balance now is: " << per1.Balance - w << endl;

per1.Balance= per1.Balance-w; //updating previous value

}void F\_Balance (void)

{

cout << " your balance is : " << per1.Balance << endl;

}

PPPPPPPPPPPPPRRRRRRRROOOOOOOOJJJJJJJJJJEEEEEEECTTTTTTTTTT::

// project\_bank.cpp : Defines the entry point for the console application.

//

#include "stdafx.h"

#include<iostream>

#include<string>

using namespace std;

class Bank // Base class

{ protected:

char bank\_name[100];

char branch\_code[100];

char location[1000];

string s1;

public:

Bank () // default constructor

{ strcpy\_s(bank\_name,"\0");

strcpy\_s(branch\_code,"\0");

strcpy\_s(location,"\0");

}

Bank(char bname[], char bcode[],char blocation[] ) // parametrized constructor

{ strcpy\_s(bank\_name,bname);

strcpy\_s(branch\_code,bcode);

strcpy\_s(location,blocation);

}

void get\_information ()

{ cout<<" Enter the name of the Bank :: ";

cin>>bank\_name;

cout<<" Enter the Branch code :: ";

cin>>branch\_code;

cout<<" Enter the location of the Bank :: ";

cin>>location;

}

void display ()

{ cout<<"\*\*\*\* NAME OF BANK \*\*\*\*"<<endl;

cout<<" "<<bank\_name<<" "<<endl;

cout<<"\*\*\*\* BRANCH CODE OF THE BANK \*\*\*\*"<<endl;

cout<<" "<<branch\_code<<" "<<endl;

cout<<"\*\*\*\* NAME OF BANK \*\*\*\*"<<endl;

cout<<" "<<location<<" "<<endl;

}

};

class bank\_employee : public Bank

{ private:

int ID;

protected:

char name[100];

char address[100];

long contact ;

int salary;

public:

bank\_employee():Bank(),ID(0), contact(0), salary(0)

{ strcpy\_s(name,"\0");

strcpy\_s(address,"\0");

}

bank\_employee(char bname[], char bcode[],char blocation[], int empID, char empname[], char empaddress[], long c, int s):Bank(bname,bcode,blocation),ID(empID), contact(c), salary(s)

{ strcpy\_s(name,empname);

strcpy\_s(address,empaddress);

}

void get\_information ()

{ cout<<" Enter the Name of the Employee :: ";

cin>>name;

cout<<" Enter the ID Number of the Employee :: ";

cin>>ID;

cout<<" Enter the Address of the Employee :: ";

cin>>address;

cout<<" Enter the Contact of the Employee :: ";

cin>>contact;

}

void calculate\_salary ()

{ int hours;

int extra\_hours;

int salary\_per\_day;

int bonus ;

for (int i= 0;i<5;i++)

{ bank\_employee::get\_information();

cout<<" Enter the working hours in a day ";

cin>>hours;

cout<<" Enter the Salary\_Per\_Day ";

cin>>salary\_per\_day;

cout<<" Enter the Extra\_hours of working in a month ";

cin>>extra\_hours;

if (extra\_hours==0)

{ bonus =0;

salary = (hours\*salary\_per\_day)+bonus;

cout<<"\*\*\*\* The Calculated Salary is :: ";

cout<<salary<<endl;

}

else if ((extra\_hours>0)&&(extra\_hours<=10))

{ bonus =5000;

salary = (hours\*salary\_per\_day)+bonus;

cout<<"\*\*\*\* The Calculated Salary is :: ";

cout<<salary<<endl;

}

else if ((extra\_hours>10)&&(extra\_hours<=15))

{ bonus =7000;

salary = (hours\*salary\_per\_day)+bonus;

cout<<"\*\*\*\* The Calculated Salary is :: ";

cout<<salary<<endl;

}

else if ((extra\_hours>16)&&(extra\_hours<=20))

{ bonus =10000;

salary = (hours\*salary\_per\_day)+bonus;

cout<<"\*\*\*\* The Calculated Salary is :: ";

cout<<salary<<endl;

}

else if (extra\_hours>21)

{ cout<<" GOOD JOB "<<endl;

cout<<" The bank will go through your progress report and will list your name in promotion file. "<<endl;

}

else

{ cout<<" INVALID ENTRY "<<endl<<" You have entered an invalid character please try again "<<endl;

cout<<" Regards "<<endl;

}

}

}

void display ()

{ cout<<"\*\*\*\* RECORD OF EMPLOYEE \*\*\*\*"<<endl;

cout<<"\*\*\*\* NAME \*\*\*\*"<<endl;

cout<<" "<<name<<" "<<endl;

cout<<"\*\*\*\* ID NUMBER \*\*\*\*"<<endl;

cout<<" "<<ID<<" "<<endl;

cout<<"\*\*\*\* ADDRESS \*\*\*\*"<<endl;

cout<<" "<<address<<" "<<endl;

cout<<"\*\*\*\* CONTACT NUMBER \*\*\*\*"<<endl;

cout<<" "<<contact<<" "<<endl;

cout<<"\*\*\*\* SALARY \*\*\*\*"<<endl;

cout<<" "<<salary<<" "<<endl;

}

};

class ATMs : public Bank

{ protected:

long pin; // pin number of the customer

char operator\_name[100];

long withdraw\_money ; // Money to be with drawn

long balance ;

public:

ATMs():Bank(),pin(0),withdraw\_money(0)

{strcpy\_s(operator\_name,"\0");}

ATMs(char bname[], char bcode[],char blocation[],long p, char opename[],long money, long b):Bank(bname,bcode,blocation),pin(p),withdraw\_money(money),balance(b)

{strcpy\_s(operator\_name,opename);}

void pin\_verification ()

{ cout<<" Please enter your pin number :: "<<endl;

cin>>pin;

cout<<endl;

if ((pin==5151)||(pin==6165)||(pin==7175)||(pin==8185)||(pin==9195))

{ cout<<" PIN Number is verified "<<endl;}

else

{cout<<" You had entered invalid PIN Number "<<endl;

cout<<" Please try again and enter the correct PIN Number "<<endl;}

}

void transaction\_details () // function for withdrawing money

{ cout<<" Please enter your name ::";

cin>>operator\_name;

cout<<endl;

cout<<" Balance of your account ::";

cin>>balance;

cout<<" Please enter the amount of money you want to withdraw ::";

cin>>withdraw\_money;

if(withdraw\_money<balance)

{ balance = balance - withdraw\_money;

cout<<" Now your remaining balance is ::"<<balance<<endl;

}

else

{ cout<<" Your entered amount is greater than your balance "<<endl;

cout<<" Please try again "<<endl;

}

}

void change\_pin ()

{ int choice;

cout<<" Do you want to chnage your PIN Number "<<endl;

cout<<" If yes than Enter 1 otherwise enter 0 ::";

cin>>choice;

if (choice==1)

{ cout<<" Enter the new PIN Number ::";

cin>>pin;

cout<<endl;

}

else if (choice == 0)

{ }

else

{ cout<<" You had entered some wrong number "<<endl<<" Please try again "<<endl;

}

}

void end\_session ()

{ char date\_of\_issue[100];

char expiry\_date[100];

int validity\_years=5;

cout<<" Enter the year of issuing card :: ";

cin>>date\_of\_issue;

cout<<endl;

cout<<" Enter the expiry year of ATM card :: ";

cin>>expiry\_date;

cout<<endl;

if ((date\_of\_issue-expiry\_date)<=validity\_years)

{ cout<<" Your card is valid you can withdraw money "<<endl;

}

else

{ cout<<" Your card had expired, you can withdraw money "<<endl;}

}

};

class contact : public Bank

{ public:

void help ()

{ int choice =0;

cout<<" Well Come to our helpline "<<endl;

cout<<"Enter 1 for helpline of Rawalpindi branch "<<endl;

cout<<"Enter 2 for helpline of Karachi branch "<<endl;

cout<<"Enter 3 for helpline of Peshawar branch "<<endl;

cout<<"Enter 4 for helpline of Quetta branch "<<endl;

cout<<"Enter 5 for helpline of Islamabad branch "<<endl;

cin>>choice;

if(choice==1)

{ cout<<" Contact Number :: +92 335 7816543 "<<endl;

cout<<" Gmail :: helpline.pk@gmail.com "<<endl;

cout<<" Opening Time Of Bank :: 9:00 AM "<<endl;

cout<<" Break Time Of Bank :: 11:30AM - 1:00PM "<<endl;

cout<<" Closing Time Of Bank :: 8:00 PM "<<endl;

}

else if (choice==2)

{ cout<<" Contact Number :: +92 335 7817853 "<<endl;

cout<<" Gmail :: helpline.pk@gmail.com "<<endl;

cout<<" Opening Time Of Bank :: 9:00 AM "<<endl;

cout<<" Break Time Of Bank :: 11:00AM - 2:00PM "<<endl;

cout<<" Closing Time Of Bank :: 7:00 PM "<<endl;

}

else if (choice==3)

{cout<<" Contact Number :: +92 335 4562870 "<<endl;

cout<<" Gmail :: helpline.pk@gmail.com "<<endl;

cout<<" Opening Time Of Bank :: 9:00 AM "<<endl;

cout<<" Break Time Of Bank :: 11:00AM - 1:30PM "<<endl;

cout<<" Closing Time Of Bank :: 9:00 PM "<<endl;

}

else if (choice==4)

{cout<<" Contact Number :: +92 335 4567321 "<<endl;

cout<<" Gmail :: helpline.pk@gmail.com "<<endl;

cout<<" Opening Time Of Bank :: 8:00 AM "<<endl;

cout<<" Break Time Of Bank :: 11:00AM - 2:00PM "<<endl;

cout<<" Closing Time Of Bank :: 9:00 PM "<<endl;

}

else if (choice==5)

{cout<<" Contact Number :: +92 334 4567330 "<<endl;

cout<<" Gmail :: helpline.pk@gmail.com "<<endl;

cout<<" Opening Time Of Bank :: 9:00 AM "<<endl;

cout<<" Break Time Of Bank :: 11:30AM - 1:30PM "<<endl;

cout<<" Closing Time Of Bank :: 10:00 PM "<<endl;

}

else

{ cout<<" You had entered an invalid number "<<endl;

cout<<" Please try again "<<endl;

}

}

};

class Accounts : public Bank

{ protected:

long acc\_numb;

float acc\_balance;

char name[100];

char acc\_type [100];

char branch\_location [100];

public:

Accounts():Bank() // default constructor

{ strcpy\_s(name,"\0");

strcpy\_s(acc\_type,"\0");

strcpy\_s(branch\_location,"\0");}

Accounts (char bname[], char bcode[],char blocation[], char acc[], char branchloc[],char namee[], float b, long account\_num):Bank(bname,bcode,blocation),acc\_balance(b),acc\_numb(account\_num)

{ strcpy\_s(name,namee);

strcpy\_s(acc\_type,acc);

strcpy\_s(branch\_location,branchloc);}

virtual void get\_data ()=0;

virtual void display ()=0;

virtual void calculation()=0;

void put\_data ()

{ Bank::get\_information();

cout<<"Enter your name : ";

cin>>name;

cout<<" Enter the type of account :: ";

cin>>acc\_type;

cout<<endl;

cout<<" Enter the location of the branch :: ";

cin>>branch\_location;

cout<<endl;

cout<<"Enter the account balance : ";

cin>>acc\_balance;

cout<<endl;

}

void show\_data ()

{ Bank::display();

cout<<"\*\* NAME \*\*"<<endl;

cout<<" "<<name<<endl;

cout<<"\*\* ACCOUNT BALANCE \*\*"<<endl;

cout<<" "<<acc\_balance<<endl;

cout<<"\*\* ACCOUNT TYPE \*\*"<<endl;

cout<<" "<<acc\_type<<endl;

cout<<"\*\* BRANCH LOCATION \*\*"<<endl;

cout<<" "<<acc\_type<<endl;

}

void credit ()

{ float balance=0;

cout<<"Enter the amount of balance you wanted to add in your account : ";

cin>>balance;

acc\_balance=acc\_balance+balance;

cout<<"Now your Account Balance is : "<<acc\_balance;

cout<<endl;

}

void debit ()

{ float c;

cout<<"Enter the amount of balance you wanted to Withdraw from your account : ";

cin>>c;

cout<<endl;

if (c>acc\_balance)

{

cout<<" Your Withdraw amount is greater than the account balance "<<endl;

cout<<" You cannot withdraw this amount "<<endl;

cout<<"Please enter the amount again"<<endl;

}

else

{ acc\_balance= acc\_balance-c;

cout<<"Now your Account Balance is : "<<acc\_balance;

cout<<endl;

}

}

};

class statement : public Accounts

{

};

class current\_account : public Accounts

{ protected:

int fixed\_fee;

int PIN;

public:

current\_account(): Accounts(), fixed\_fee(0),PIN(0)

{}

current\_account(char bname[], char bcode[],char blocation[], char acc[], char branchloc[],char namee[], float b, long account\_num ,int f, int p):Accounts( bname, bcode, blocation, acc, branchloc, namee, b, account\_num ), fixed\_fee(f), PIN(p)

{}

void get\_data()

{ cout<<"\*\* Current Account \*\*"<<endl;

Accounts::put\_data();

cout<<endl;

cout<<" Enter you Account Number :: ";

cin>>acc\_numb;

cout<<" Enter you PIN :: ";

cin>>PIN;

cout<<endl;

cout<<" Bank will enter the fixed fee amount :: ";

cin>>fixed\_fee;

cout<<endl;

}

void display ()

{ cout<<" \*\* Displaying the details of Current Account \*\*"<<endl;

Accounts::show\_data();

cout<<endl;

cout<<" \*\* ACCOUNT NUMBER \*\* "<<endl;

cout<<" "<<acc\_numb<<endl;

cout<<" \*\* PIN \*\* "<<endl;

cout<<" "<<PIN<<endl;

cout<<" \*\* FIXED FEE AMOUNT \*\* "<<endl;

cout<<fixed\_fee<<endl;

}

void calculation ()

{ cout<<" Credit Function "<<endl;

Accounts::credit();

cout<<" Dedit Function "<<endl;

Accounts::debit();

acc\_balance = acc\_balance- fixed\_fee;

cout<<"Now your current balance after deducting fixed\_fee is :: "<<acc\_balance<<endl;

}

};

class saving\_account : public Accounts

{ protected:

int pin;

public:

saving\_account():Accounts(),pin(0)

{}

saving\_account(char bname[], char bcode[],char blocation[], char acc[], char branchloc[],char namee[], float b,long account\_num,int p):Accounts(bname, bcode, blocation, acc, branchloc, namee, b, account\_num ),pin(0)

{}

void get\_data()

{ cout<<"\*\* Saving Account \*\*"<<endl;

Accounts::put\_data();

cout<<endl;

cout<<" Enter your PIN :: ";

cin>>pin;

cout<<endl;

}

void display ()

{ cout<<" \*\* Displaying the details of Saving Account \*\*"<<endl;

Accounts::show\_data();

cout<<" \*\* PIN NUMBER \*\* "<<endl;

cout<<pin;

cout<<endl;

}

void calculation ()

{ Accounts::credit();

Accounts::debit();

cout<<" Your minimum balance should be 500 "<<endl;

cout<<" Deduction of interest every year "<<endl;

if (acc\_balance>500)

{ acc\_balance = acc\_balance - 0.025;

cout<<" Now your current balance is :: "<<acc\_balance;

cout<<endl;

}

else

{ cout<<" Your account should contain a minimum balance of 500 "<<endl;

cout<<" Add the minimum balance in the account or your account will be blocked "<<endl;

}

}

};

class loan\_account : public Accounts

{ protected:

long loan\_numb;

char loan\_type[100];

public:

loan\_account():Accounts(),loan\_numb(0)

{strcpy(loan\_type,"\0");

}

loan\_account(char bname[], char bcode[],char blocation[], char acc[], char branchloc[],char namee[], float b,long account\_num,long lnumb,char ltype[]):Accounts(bname, bcode, blocation, acc, branchloc, namee, b, account\_num ),loan\_numb(lnumb)

{strcpy(loan\_type,ltype);

}

void get\_data()

{ cout<<"\*\* Loan Account \*\*"<<endl;

Accounts::put\_data();

cout<<" Enter the loan number issued to you ::";

cin>>loan\_numb;

cout<<" Enter the loan type ::";

cin>>loan\_type;}

void display ()

{ cout<<" \*\* Displaying the details of Loan Account \*\*"<<endl;

Accounts::show\_data();

cout<<" \*\* LOAN NUMBER \*\* "<<endl;

cout<<loan\_numb<<endl;

cout<<" \*\* LOAN\_TYPE \*\*"<<endl;

cout<<loan\_type<<endl;}

};

class online\_transaction :public Accounts

{ private:

int cust\_pin; // pin number

long CNIC\_numb;

long card\_numb;

char password[100];

public:

online\_transaction():Accounts(),cust\_pin(0),CNIC\_numb(0),card\_numb(0)

{ strcpy\_s(password,"\0");

}

online\_transaction(char bname[], char bcode[],char blocation[], char acc[], char branchloc[],char namee[], float b,long account\_num,int p,long cnic, long cardnumb, char passwordd[]):Accounts(),cust\_pin(p),CNIC\_numb(cnic),card\_numb(cardnumb)

{ strcpy\_s(password,passwordd);

}

void transaction ()

{ cout<<"\*\* ONLINE TRANSACTION \*\*";

Accounts::debit();

cout<<endl;

}

void get\_data()

{ cout<<"\*\* Online Transaction \*\*"<<endl;

Accounts::put\_data();

cout<<" Enter the PIN Number ::";

cin>>cust\_pin;

cout<<" Enter CNIC Number ::";

cin>>CNIC\_numb;

cout<<" Enter the CARD Number ::";

cin>>card\_numb;

cout<<" Enter password ::";

cin>>password;

if (password=="mano7175")

{ cout<<" You had entered correct password "<<endl;

cout<<" You can do online transaction "<<endl;

}

else

{ cout<<" You had entered wrong password "<<endl<<" Please try again "<<endl; }

}

void display ()

{ cout<<" \*\* Displaying the details of Online transaction \*\*"<<endl;

Accounts::show\_data();

cout<<" \*\* PIN NUMBER \*\* "<<endl;

cout<<cust\_pin<<endl;

cout<<" \*\* CNIC NUMBER \*\*"<<endl;

cout<<CNIC\_numb<<endl;

cout<<" \*\* CARD NUMBER \*\* "<<endl;

cout<<cust\_pin<<endl;

cout<<" \*\* REMAINING BALANCE \*\* "<<endl;

cout<<acc\_balance<<endl;

}

};

class customer

{ protected:

char cust\_name[100];

int cust\_id;

long cust\_contact\_numb;

char gmail[100];

char password[50];

char address[100];

char user\_name[50];

public:

customer():cust\_id(0), cust\_contact\_numb(0)

{ strcpy\_s( cust\_name,"\0");

strcpy\_s( gmail,"\0");

strcpy\_s( password,"\0");

strcpy\_s( address,"\0");

strcpy\_s( user\_name,"\0");

}

customer(char name[], int id,long contactnumb, char g[], char p[], char a[], char usernamee[] ):cust\_id(id), cust\_contact\_numb(contactnumb)

{ strcpy\_s( cust\_name,name);

strcpy\_s( gmail,g);

strcpy\_s( password,p);

strcpy\_s( address,a);

strcpy\_s( user\_name,usernamee);

}

void add\_customer ()

{ cout<<" Enter the Customer Name ::";

cin>>cust\_name;

cout<<endl;

cout<<" Enter the Customer ID ::";

cin>>cust\_id;

cout<<endl;

cout<<" Enter the Customer Contact Number :: ";

cin>>cust\_contact\_numb;

cout<<endl;

cout<<" Enter the Gmail :: ";

cin>>gmail;

cout<<endl;

cout<<" Enter the Password :: ";

cin>>cust\_name;

cout<<endl;

cout<<" Enter the Address :: ";

cin>>gmail;

cout<<endl;

cout<<" Enter the User Name :: ";

cin>>cust\_name;

cout<<endl;

}

void edit\_customer ()

{ cout<<" \*\* Editing Customer Details \*\* "<<endl;

add\_customer();

}

void search\_customer ()

{ cout<<" \*\* Searching Customer \*\* "<<endl;

cin>>cust\_name;

add\_customer();

}

};

int \_tmain(int argc, \_TCHAR\* argv[])

{/\* Bank b1;

b1.get\_information();

b1.display();

bank\_employee o;

o.calculate\_salary();

o.display();

ATMs a;

a.end\_session();

a.pin\_verification();

a.transaction\_details();

a.change\_pin();\*/

contact c;

c.help();

system("pause");

return 0;

}

***REPORT***

This report is about the ***“bank management system”.*** This will help you to check your total amount in your bank account. the transaction amount as well as the deposit amount.

***Procedure:***

First of all we use structures to enter the client information.we use char datatype to enter name,and int data type to enter account number,and double data type for balance.

Secondly we declare functions.we use functions for deposit, transaction and total amount in bank account.

Thirdly,we declare some variables that we use in program.and we initialize the name,account number,and total balance in account.we use if-else statement and switch statement to apply some conditions.we also goto label so that user may perform many operations at the same time.this wil be very help for user.if user wants to exit the program he/she can also exit.we also added some statement for this process also.we call the function in switch statement.and after all these procedure we write the function definition.

**Flow chart**

Module-1 for structures:





