#include "stdafx.h"

#include<iostream>

using namespace std;

class Bank // Base class

{ protected:

char bank\_name[100];

char branch\_code[100];

char location[1000];

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 () // Information getting function

{ 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 () // Display function

{ 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 // derive class inherited from Bank

{ private:

int ID; // id number

protected:

char name[100];

char address[100];

long contact ;

int salary;

public:

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

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

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

}

// parametrized constructor

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 () // function for calculating salary of employees

{ 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<<endl;

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

cin>>salary\_per\_day;

cout<<endl;

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;

cout<<endl;

cout<<endl;

cout<<endl;

}

};

class ATMs : public Bank // Derive class inherited from base class(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) // Default constructor

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

// parametrized Constructor

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 () // function for validity of PIN number

{ cout<<endl;

cout<<endl;

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;

}

cout<<endl;

cout<<endl;

cout<<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 ()

{ cout<<endl;

cout<<endl;

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;

}

cout<<endl;

cout<<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");}

// parametrized function

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);}

// pure virtual functions

virtual void get\_data ()=0;

virtual void display ()=0;

virtual void calculation()=0;

void put\_data ()

{ cout<<endl;

cout<<endl;

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 () // function for adding money in account

{ 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 () // function for withdrawing money from account

{ 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

{ protected:

char time[30];

char date[30];

public:

statement():Accounts()

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

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

}

statement(char bname[], char bcode[],char blocation[], char acc[], char branchloc[],char namee[], float b, long account\_num, char t[], char d[]):Accounts( bname, bcode, blocation, acc, branchloc, namee, b , account\_num)

{ strcpy\_s(time,t);

strcpy\_s(date,d);

}

void display ()

{ Accounts::show\_data();

cout<<" Your Remaining balance is :: "<<acc\_balance<<endl;

cout<<" Time of Transaction is :: "<<time<<endl;

cout<<" Date of Transaction is :: "<<date<<endl;

cout<<endl;

cout<<endl;

}

void get\_data()

{ Accounts::put\_data();

cout<<" Time of Transaction is :: ";

cin>>time;

cout<<endl;

cout<<" Date of Transaction is :: ";

cin>>date;

cout<<endl;

}

void calculation()

{ }

};

class current\_account : public Accounts

{ protected:

int fixed\_fee;

int PIN;

public:

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

{}

// parametrized constructor

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 your Account Number :: ";

cin>>acc\_numb;

cout<<" Enter your 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;

cout<<endl;

cout<<endl;

cout<<endl;

}

};

class saving\_account : public Accounts

{ protected:

int pin;

public:

saving\_account():Accounts(),pin(0) // Default constructor

{}

// parametrized constructor

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 ()

{ cout<<" \*\*\*\* CREDIT FUNCTION \*\*\*\* "<<endl;

Accounts::credit();

cout<<" \*\*\*\* DEBIT FUNCTION \*\*\*\* "<<endl;

Accounts::debit();

cout<<" Your minimum balance should be 500 in your account. "<<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;

}

cout<<endl;

cout<<endl;

cout<<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;}

void calculation ()

{ Accounts::calculation();

cout<<" Your application will be considered by the Bank "<<endl;

cout<<" You will be informed later whether you can avail loan or not "<<endl;

cout<<" Thanks "<<endl;

cout<<endl;

cout<<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 calculation ()

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

Accounts::debit();

cout<<endl;

cout<<endl;

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<<" Oopssss, you had entered wrong password. "<<endl<<" Please try later. "<<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;

char cust\_contact\_numb[10];

char gmail[100];

char password[50];

char address[100];

char user\_name[50];

public:

customer():cust\_id(0)

{ strcpy\_s( 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,char contact[], char g[], char p[], char a[], char usernamee[] ):cust\_id(id)

{ strcpy\_s( cust\_contact\_numb,contact);

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 ()

{ for (int i=0;i<3;i++)

{

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

cin>>cust\_name;

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

cin>>cust\_id;

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

cin>>cust\_contact\_numb;

cout<<" Enter the Gmail :: ";

cin>>gmail;

cout<<" Enter the Password :: ";

cin>>password;

cout<<" Enter the Address :: ";

cin>>address;

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

cin>>user\_name;

display();

}

}

void display ()

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

cout<<cust\_name<<endl;

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

cout<<cust\_id<<endl;

cout<<" \*\*\*\* Customer Contact Number \*\*\*\* "<<endl;

cout<<cust\_contact\_numb<<endl;

cout<<" \*\*\*\* Customer Gmail \*\*\*\* "<<endl;

cout<<gmail<<endl;

cout<<" \*\*\*\* Customer Password \*\*\*\* "<<endl;

cout<<password<<endl;

cout<<" \*\*\*\* Customer Address \*\*\*\* "<<endl;

cout<<address<<endl;

cout<<" \*\*\*\* Customer User Name \*\*\*\* "<<endl;

cout<<user\_name<<endl;

}

void edit\_customer ()

{ cout<<endl;

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

if (cust\_name=="Mahrukh")

{add\_customer();}

else if (cust\_name=="Izwa\_Afzal")

{add\_customer();}

else if (cust\_name=="Sobia\_Khalid")

{add\_customer();}

else

cout<<" Customer not found "<<endl;

}

void search\_customer ()

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

cin>>cust\_name;

if (cust\_name=="Mahrukh")

{display();}

else if (cust\_name=="Izwa\_Afzal")

{display();}

else if (cust\_name=="Sobia\_Khalid")

{display();}

cout<<" Customer not found "<<endl;

}

};

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

{ cout<<endl;

cout<<" ||:) WELLCOME TO BANK MANAGEMENT SYSTEM ||:) ";

cout<<endl;

int choice;

cout<<"Enter 1 for the Bank Employee Information "<<endl;

cout<<"ENter 2 for the ATM Machine "<<endl;

cout<<"Enter 3 for the Helpline "<<endl;

cout<<"ENter 4 for the Accounts "<<endl;

cin>>choice;

if (choice == 1)

{ bank\_employee obj1;

obj1.get\_information();

obj1.calculate\_salary();

obj1.display();

}

else if (choice == 2)

{ ATMs obj2;

obj2.end\_session();

obj2.pin\_verification();

obj2.transaction\_details();

obj2.change\_pin();

}

else if (choice == 3)

{ contact obj3;

obj3.help();

}

else if (choice == 4)

{

}

else

{ cout<<" You had enter wrong number "<<endl;

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

}

}

// ya mai nai upar else if(choice == 4) mai dalna hai ya sara.

Accounts \*p[10];

int choice;

for (int i=0; i<10;i++)

{

cout<<"Enter 1 for the Saving account "<<endl;

cout<<"ENter 2 for the Current account "<<endl;

cout<<"Enter 3 for the Loan account "<<endl;

cout<<"ENter 4 for the Online Transaction "<<endl;

cout<<"Enter 5 for the Statement Print "<<endl;

cin>>choice;

if (choice == 1)

{ p[1] = new saving\_account;

p[1]->get\_data();

p[1]->display();

p[1]->calculation();

}

else if (choice == 2)

{ p[2] = new current\_account;

p[2]->get\_data();

p[2]->display();

p[2]->calculation();

}

else if (choice == 3)

{ p[3] = new loan\_account;

p[3]->get\_data();

p[3]->display();

p[3]->calculation();

}

else if (choice == 4)

{ p[4] = new online\_transaction;

p[4]->get\_data();

p[4]->display();

p[4]->calculation();

}

else if (choice == 5)

{ p[5] = new statement;

p[5]->get\_data();

p[5]->display();

}

else

{ cout<<" You had enter wrong number "<<endl;

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

}

system("pause");

return 0;

}