Final Code

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

#include<dos.h>

#include<stdlib.h>

#include<iostream.h>

#include<string.h>

//#include<exception.h>

// template functions

template<class x>

void message(x a)

{

if(a==1)

cout<<"\n\t\t!!!!!!!!!!!Operation Successfully Done!!!!!!!!";

else

cout<<"\n\t\t!!!!!!!!!!!Operation Failed!!!!!!!!!!!!!!!";

}

class MyException //FOR HANDLING THE EXCEPTIONS .

{

public :

char message[80];

MyException(){\*message=0;}

MyException(char \*s){strcpy(message,s);}

};

class ATM

{

protected:

int atm\_number;

int atm\_password;

char\* accd;

public:

ATM(){}

ATM(int n,int p,char c[])

{

fflush(stdin);

atm\_number=n;

atm\_password=p;

strcpy(accd,c);

}

int getatm(){return atm\_number;}

char\* getatmid(){return accd;}

int getpass(){return atm\_password;}

~ATM(){}

};

class Account

{

protected:

int password ;

char acc\_id[21] ;

char name[21] ;

char phone[10] ;

float balance ;

char userid[21] ;

ATM \*a; //FOR COMPOSITION RELATIONSHIP .

public :

Account(){}

Account(int p,char aid[],char nm[],char ph[],float bl,char uid[])

{

fflush(stdin);

password=p;

strcpy(acc\_id,aid);

strcpy(name,nm);

strcpy(phone,ph);

balance=bl;

strcpy(userid,uid);

}

virtual char\* getid(){}

virtual void deduct(char[] , float){}

virtual void credit(char[] , float){}

virtual void atm\_exist(){}

virtual void details(){}

~Account(){delete a;} //COMPOSITION RELATIONSHIP

};

class Savings :virtual public Account

{

public:

Savings():Account(){}

char\* getid(){return acc\_id;}

Savings(int p,char aid[],char nm[],char ph[],float bl,char uid[]):Account(p,aid,nm,ph,bl,uid){}

void deduct(char c[],float ded)

{

if(strcmp(getid(),c)==0)

balance=balance-ded;

//EXCEPTIONAL HANDLING 1.

/\*try{

if(balance<0)

throw MyException("Sorry Insufficient Funds . ");

}

catch(MyException e)

{

cout<<"\n"<<e.message;

cout<<"\n";

}\*/

else

cout<<"\nSorry Invalid Account Number .";

}

void credit(char c[], float add)

{

if(strcmp(getid(),c)==0)

balance=balance+add;

else

cout<<"\nSorry Invalid Account Number .";

}

void details()

{

cout<<endl<<"\tUser ID : "<<userid;

cout<<endl<<"\tName : "<<name;

cout<<endl<<"\tPhone : "<<phone;

cout<<endl<<"\tBalance Left : "<<balance;

cout<<endl<<"\tPassword : "<<password;

}

~Savings(){}

};

class Current :virtual public Account

{

public:

Current():Account(){}

char\* getid(){return acc\_id;}

Current(int p,char aid[],char nm[],char ph[],float bl,char uid[]):Account(p,aid,nm,ph,bl,uid){}

void deduct(char c[],float ded)

{

if(strcmp(getid(),c)==0)

balance=balance-ded;

//EXCEPTIONAL HANDLING 2.

/\*try{

if(balance<0)

throw MyException("Sorry Insufficient Funds . ");

}

catch(MyException e)

{

cout<<"\n"<<e.message;

cout<<"\n";

}\*/

else

cout<<"\nSorry Invalid Account Number .";

}

void credit(char c[], float add)

{

if(strcmp(getid(),c)==0)

balance=balance+add;

else

cout<<"\nSorry Invalid Account Number .";

}

void details()

{

cout<<endl<<"\tUser ID : "<<userid;

cout<<endl<<"\tName : "<<name;

cout<<endl<<"\tPhone : "<<phone;

cout<<endl<<"\tBalance Left : "<<balance;

cout<<endl<<"\tPassword : "<<password;

}

~Current(){}

};

class Locker : virtual public Savings , virtual public Current

{

protected :

int locker\_num ;

char type[21] ;

float rent ;

char date[10] ;

public :

Locker():Savings(),Current(){}

int getlocknum(){return locker\_num;}

void create()

{

clrscr();

cout<<"\nEnter locker number:";

cin>>locker\_num;

cout<<"\nEnter type:";

gets(type);

cout<<"\nEnter Rent:";

cin>>rent;

cout<<"\nEnter Date:";

gets(date);

//EXCEPTIONAL HANDLING 3

/\*

try{

if(locker\_num<0 || strlen(type)>21 || rent<0 || strlen(date)>10)

throw MyException("Exception Caught . Try Again !!!!!!!!!!!!!!");

}

catch(...) //CAUGHT ALL EXCEPTIONS .

{

cout<<"\n"<<e.message;

create();

}

\*/

cout<<"\t\t\t ================================";

cout<<"\t\t\t|| Locker created Successfully ||";

cout<<"\t\t\t =================================";

}

void update()

{

float r;

char d[10];

cout<<"\n===========Welcome=============";

cout<<"\nEnter your Updated rent : ";

cin>>r;

cout<<"\nEnter today's date : ";

gets(d);

//EXCEPTIONAL HANDLING 4

/\*

try{

if(strlen(d)>10 || r<0)

throw myException("Exception Caught .Try Again!!!!!!!!!!!!!!!!!!!!!");

}

catch(...) //CAUGHT ALL EXCEPTIONS .

{

cout<<e.message();

update();

}

\*/

rent=r;

strcpy(date,d);

}

void details()

{

cout<<endl<<"Locker Number : "<<locker\_num;

cout<<endl<<"Locker Type : "<<type;

cout<<endl<<"Rent : "<<rent;

cout<<endl<<"Date of issue : "<<date;

}

friend int checks(char aid[],Savings &s);

friend int checkc(char aid[],Current &c);

~Locker(){}

};

class LOAN : virtual public Savings , virtual public Current

{

protected :

char loan\_id[21] ;

float rate\_interest;

int num\_of\_Installments;

int installments\_paid;

float principle ;

float amount\_paid;

float fixed\_amt;

char date[10] ;

public :

LOAN():Savings(),Current(){}

virtual char\* getlid(){}

virtual void create(){}

virtual void details(){}

virtual int check(char []){}

virtual int check\_marks(float){}

virtual int check\_salary(float){}

virtual void update(){}

~LOAN(){}

};

class CAR :virtual public LOAN

{

protected :

float salary ;

public :

CAR():LOAN(){}

char\* getlid(){return loan\_id;}

void create()

{

clrscr();

cout<<"\nEnter loan id:";

gets(loan\_id) ;

cout<<"\nEnter Loan amount:";

cin>>principle ;

cout<<"\nEnter date:";

gets(date) ;

cout<<"\nEnter fixed amount to be paid:";

cin>>fixed\_amt;

//EXCEPTIONAL HANDLING 5

/\*

try{

if(strlen(loan\_id)>21 || principle<0 || strlen(date)>10)

throw myException("Exception Caught .Try Again!!!!!!!!!!!!!!!!!!!!!");

}

catch(...) //CAUGHT ALL EXCEPTIONS .

{

cout<<e.message();

create();

}

\*/

rate\_interest=5;

num\_of\_Installments=12;

installments\_paid=0 ;

amount\_paid=0 ;

}

int check\_salary(float sal)

{

salary=sal;

if(salary>=200)

return 1;

else

return 0;

}

void update()

{

int ip;

float ap;

cout<<"\n Enter your Updated Installments Paid";

cin>>ip;

cout<<"\n Enter your Updated Amount Paid";

cin>>ap;

//EXCEPTIONAL HANDLING 6

/\*

try{

if(ip<0 || ap<0)

throw myException("Exception Caught .Try Again!!!!!!!!!!!!!!!!!!!!!");

}

catch(...) //CAUGHT ALL EXCEPTIONS .

{

cout<<e.message();

create();

}

\*/

installments\_paid=ip;

amount\_paid=ap;

}

void details()

{

cout<<endl<<"Loan ID : "<<loan\_id;

cout<<endl<<"Rate of Interest : "<<rate\_interest;

cout<<endl<<"Number of installments : "<<num\_of\_Installments;

cout<<endl<<"Installments paid : "<<installments\_paid;

cout<<endl<<"Principle : "<<principle;

cout<<endl<<"Amount Paid : "<<amount\_paid;

cout<<endl<<"Fixed amount : "<<fixed\_amt;

cout<<endl<<"Date : "<<date;

}

friend int checks(char aid[],Savings &s);

friend int checkc(char aid[],Current &c);

~CAR(){}

};

class EDUCATION :virtual public LOAN

{

protected :

float twelth\_marks ;

public :

EDUCATION():LOAN(){}

char\* getlid(){return loan\_id;}

void create()

{

clrscr();

cout<<"\nEnter loan id:";

gets(loan\_id) ;

cout<<"\nEnter Loan amount:";

cin>>principle ;

cout<<"\nEnter date:";

gets(date) ;

cout<<"\nEnter fixed amount to be paid:";

cin>>fixed\_amt;

//EXCEPTIONAL HANDLING 7

/\*

try{

if(strlen(loan\_id)>21 || principle<0 || strlen(date)>10 || fixed\_amt<0)

throw myException("Exception Caught .Try Again!!!!!!!!!!!!!!!!!!!!!");

}

catch(...) //CAUGHT ALL EXCEPTIONS .

{

cout<<e.message();

create();

}

\*/

rate\_interest=10;

num\_of\_Installments=12;

installments\_paid=0 ;

amount\_paid=0 ;

}

int check\_marks(float mar)

{

twelth\_marks=mar;

if(twelth\_marks>=80)

return 1;

else

return 0;

}

void update()

{

int ip;

float ap;

cout<<"\n Enter your Updated Installments Paid";

cin>>ip;

cout<<"\n Enter your Updated Amount Paid";

cin>>ap;

//EXCEPTIONAL HANDLING 8

/\*

try{

if(ip<0 || ap<0)

throw myException("Exception Caught .Try Again!!!!!!!!!!!!!!!!!!!!!");

}

catch(...) //CAUGHT ALL EXCEPTIONS .

{

cout<<e.message();

create();

}

\*/

installments\_paid=ip;

amount\_paid=ap;

}

void details()

{

cout<<endl<<"Loan ID : "<<loan\_id;

cout<<endl<<"Rate of Interest : "<<rate\_interest;

cout<<endl<<"Account ID : "<<acc\_id;

cout<<endl<<"Number of installments : "<<num\_of\_Installments;

cout<<endl<<"Installments paid : "<<installments\_paid;

cout<<endl<<"Principle : "<<principle;

cout<<endl<<"Amount Paid : "<<amount\_paid;

cout<<endl<<"Fixed amount : "<<fixed\_amt;

cout<<endl<<"Date : "<<date;

}

friend int checks(char aid[],Savings &s);

friend int checkc(char aid[],Current &c);

~EDUCATION(){}

};

class DD : virtual public Savings , virtual public Current

{

protected :

//int dd\_num; In Login Class;

//float amount; In Login Class;

char date[10] ;

char from[21] ;

char to[21] ;

public :

DD():Savings(),Current(){}

void create()

{

cout<<"\nEnter Date:";

gets(date) ;

cout<<"\nEnter From:";

gets(from) ;

cout<<"\nEnter To:";

gets(to) ;

//EXCEPTIONAL HANDLING 9

/\*

try{

if(strlen(date)>10 || strlen(from)>21 || strlen(to)>21)

throw myException("Exception Caught .Try Again!!!!!!!!!!!!!!!!!!!!!");

}

catch(...) //CAUGHT ALL EXCEPTIONS .

{

cout<<e.message();

create();

}

\*/

}

/\*print function to be checked with output in tcc\*/

void print()

{

cout<<"\n\t\t\t A/c no.:"<<acc\_id;

cout<<"\n\t\t\t From :"<<from;

cout<<"\n\t\t\t To :"<<to;

cout<<"\n\t\t\t Date :"<<date;

cout<<"\n\t\t\t ==============================";

}

friend int checks(char aid[],Savings &s);

friend int checkc(char aid[],Current &c);

~DD(){cout<<"\nDestructing DD Class";}

};

class ATM\_transaction : public ATM

{

public :

ATM\_transaction():ATM(){}

ATM\_transaction(int atmno,int pin,char acc\_id[]):ATM(atmno,pin,acc\_id){}

int check\_details(int a\_num, int pwd)

{

if(atm\_number==a\_num && atm\_password==pwd)

return 1;

else

return 0;

}

~ATM\_transaction(){}

};

//========================FRIEND FUNCTION 1 DEFINATION ========================

int checks(char aid[],Savings &s)

{

Savings ss;

ss=s;

if(strcmp(ss.getid(),aid)==0)

return 1;

else

return 0;

}

//========================FRIEND FUNCTION 2 DEFINATION ========================

int checkc(char aid[],Current &c)

{

Current cc;

cc=c;

if(strcmp(cc.getid(),aid)==0)

return 1;

else

return 0;

}

class LOGIN : virtual public Savings , virtual public Current , virtual public ATM\_transaction , virtual public LOAN, public Locker , virtual public DD

{

Savings save;

Current curnt;

CAR car;

EDUCATION edu;

ATM\_transaction atm;

DD dd;

Locker lock;

public:

void assigns(Savings &ss){save=ss;}

void assignc(Current &cc){curnt=cc;}

void assigna(ATM\_transaction &aa){atm=aa;}

void cust\_screen()

{

clrscr();

cout<<"\n\n\n\n";

cout<<"\n\t\t\t =============================";

cout<<"\n\t\t\t|| Welcome To IND BANK ||";

cout<<"\n\t\t\t|| =================== ||";

cout<<"\n\t\t\t|| ||";

cout<<"\n\t\t\t|| 1.ATM ||";

cout<<"\n\t\t\t|| 2.Account Details ||";

cout<<"\n\t\t\t|| 3.Loan Details ||";

cout<<"\n\t\t\t|| 4.Locker Details ||";

cout<<"\n\t\t\t|| 5.Clerk\_Screen ||";

cout<<"\n\t\t\t|| 6.Exit ||";

cout<<"\n\t\t\t|| ||";

cout<<"\n\t\t\t =============================";

cout<<"\n\n\t\t\t Enter Choice:";

int a;

cin>>a;

switch (a)

{

case 1:

clrscr();

int atmn;

int pin;

cout<<"\n\n\n\n";

cout<<"\n\t\t\t =================================";

cout<<"\n\t\t\t|| ||";

cout<<"\n\t\t\t|| Welcome To IND BANK ATM ||";

cout<<"\n\t\t\t|| ======================= ||";

cout<<"\n\t\t\t|| ||";

cout<<"\n\t\t\t =================================";

cout<<"\n\t\t\t\t\tEnter ATM Number:";

cin>>atmn;

cout<<"\n\t\t\t\t\tEnter PIN Number:";

cin>>pin;

if(atm.check\_details(atmn,pin)==1)

{

clrscr();

cout<<"\n\n\n\n\n";

cout<<"\n\t\t\t =======================================";

cout<<"\n\t\t\t|| 1.Balance deduction ||";

cout<<"\n\t\t\t|| 2.Balance Addition ||";

cout<<"\n\t\t\t =======================================";

cout<<"\n\t\t\t Enter your choice:";

int ds;

cin>>ds;

switch(ds)

{

case 1:

clrscr();

int a;

float ded;

cout<<"\n\t\t\tEnter 1 for savings account and 2 for current account : ";

cin>>a;

switch(a)

{

case 1:

cout<<"\n\t\t\tEnter Balance:";

cin>>ded;

save.deduct(save.getid(),ded);

break;

case 2:

cout<<"\n\t\t\tEnter Balance:";

cin>>ded;

curnt.deduct(curnt.getid(),ded);

break;

default:

cout<<"\n\t\t\tEntered wrong choice.";

message(0); //Calling the template functions .

getch();

break;

}

clrscr();

cout<<"\n\n\n\n\n";

cout<<"\t\t ==========================================";

cout<<"\n\t\t|| Thanks for your visit. Have a nice day ||";

cout<<"\n\t ==========================================";

message(1); //Calling the template functions .

getch();

cust\_screen();

break;

case 2:

clrscr();

int a1;

float add;

cout<<"\n\t\t\tEnter 1 for savings account and 2 for current account : ";

cin>>a1;

switch(a1)

{

case 1:

cout<<"\n\t\t\tEnter Balance:";

cin>>add;

save.credit(save.getid(),add);

break;

case 2:

cout<<"\n\t\t\tEnter Balance:";

cin>>add;

curnt.credit(curnt.getid(),add);

break;

default:

cout<<"\n\t\t\tEntered wrong choice.";

getch();

break;

}

clrscr();

cout<<"\n\n\n\n\n";

cout<<"\t\t ==========================================";

cout<<"\n\t\t|| Thanks for your visit. Have a nice day ||";

cout<<"\n\t ==========================================";

message(1); //Calling the template functions .

getch();

cust\_screen();

break;

default:

cout<<"\n\t\t\tEntered choice is wrong. Please try again.";

cout<<"\n\n\n\n\n";

cout<<"\t\t ==========================================";

cout<<"\n\t\t|| Thanks for your visit. Have a nice day ||";

cout<<"\n\t\t ==============================================";

message(0); //Calling the template functions .

getch();

cust\_screen();

break;

}

}

else

{

clrscr();

cout<<"\n\t\t\tEntered Details are wrong";

cout<<"\n\n\n\n\n";

cout<<"\t\t ==========================================";

cout<<"\n\t\t|| Thanks for your visit. Have a nice day ||";

cout<<"\n\t\t\t\t\t =========================================";

getch();

cust\_screen();

}

break;

case 2:

clrscr();

int flag2=0;

cout<<"\n\n\n";

cout<<"\n\t\t\tEnter your Account number:";

char accid[11];

gets(accid);

if(strcmp(save.getid(),accid)==0)

{

save.details();

flag2=1;

message(1); //Calling the template functions .

getch();

cust\_screen();

break;

}

if(flag2==0)

{

if(strcmp(curnt.getid(),accid)==0)

{

curnt.details();

flag2=1;

message(1); //Calling the template functions .

getch();

cust\_screen();

break;

}

}

if(flag2==0)

cout<<"Data not found.";

message(0); //Calling the template functions .

getch();

cust\_screen();

break;

case 3:

clrscr();

int flag3=0;

cout<<"\n\n\n";

cout<<"\n\t\t\tEnter your loan number:";

char lid[21];

gets(lid);

if(strcmp(car.getlid(),lid)==0)

{

car.details();

flag3=1;

message(1); //Calling the template functions .

getch();

cust\_screen();

break;

}

if(flag3==0)

{

if(strcmp(edu.getlid(),lid)==0)

{

edu.details();

flag3=1;

message(1); //Calling the template functions .

getch();

cust\_screen();

break;

}

}

if(flag3==0)

cout<<"Data not found.";

cout<<"\n";

getch();

cust\_screen();

break;

case 4:

clrscr();

cout<<"\n\n\n";

cout<<"\n\t\t\tEnter your locker number:";

int lnum;

int flag4=0;

cin>>lnum;

if(lock.getlocknum()==lnum)

{

lock.details();

getch();

message(1); //Calling the template functions .

cust\_screen();

flag4=1;

break;

}

if(flag4==0)

cout<<"Data not found.";

getch();

cust\_screen();

break;

case 5:

getch();

clerk\_screen();

break;

case 6:

exit(0);

break;

default:

cout<<"\n\t\t\tEntered choice is wrong. Please try again.";

message(0); //Calling the template functions .

cust\_screen();

getch();

main();

}

getch();

}

void clerk\_screen()

{

int password ;

char acc\_id[21] ;

char name[21] ;

char phone[21] ;

float balance ;

char userid[21] ;

clrscr();

cout<<"\n\t\t\t =============================";

cout<<"\n\t\t\t|| Welcome To YOU Bank ||";

cout<<"\n\t\t\t|| =================== ||";

cout<<"\n\t\t\t|| ||" ;

cout<<"\n\t\t\t|| 1.Create Loan ||";

cout<<"\n\t\t\t|| 2.Create Locker ||";

cout<<"\n\t\t\t|| 3.Create DD ||";

cout<<"\n\t\t\t|| 4.Update Loan ||";

cout<<"\n\t\t\t|| 5.Update Locker ||";

cout<<"\n\t\t\t|| 6.Deduct Balance ||";

cout<<"\n\t\t\t|| 7.Credit Balance ||";

cout<<"\n\t\t\t|| 8.Delete ATM ||";

cout<<"\n\t\t\t|| 9.Delete Account ||";

cout<<"\n\t\t\t|| 10.Cust\_Screen ||";

cout<<"\n\t\t\t|| 11.Exit ||";

cout<<"\n\t\t\t|| ||";

cout<<"\n\t\t\t =============================";

cout<<"\n\n\t\t\t Enter Choice:";

int a;

cin>>a;

switch (a)

{

case 1:

clrscr();

cout<<"Enter 1 for Education Loan OR 2 for Car Loan:";

int ch1;

cin>>ch1;

switch(ch1)

{

case 1:

clrscr();

char ac[11];

cout<<"\n\t\t\tEnter Twelth marks:";

float mark;

cin>>mark;

if(edu.check\_marks(mark))

{

cout<<"\n\t\t\tEnter Account Number:";

gets(ac);

if(checks(ac,save)==1 || checkc(ac,curnt)==1)

{

clrscr();

edu.create();

cout<<"\n\n\n\n\n";

cout<<"\n\t\t\t ================================";

cout<<"\n\t\t\t|| Loan created Successfully ||";

cout<<"\n\t\t\t =================================";

message(1); //Calling the template functions .

}

else

cout<<"\n\t\t\tInvalid Account number";

getch();

clerk\_screen();

}

else

cout<<"\n\t\t\tYour marks are low. Can't give loan";

cout<<"\n";

getch();

clerk\_screen();

break;

case 2:

clrscr();

char ac1[11];

cout<<"\n\t\t\tEnter Salary:";

float sal;

cin>>sal;

if(car.check\_salary(sal))

{

cout<<"\n\t\t\tEnter Account Number:";

gets(ac1);

if(checks(ac1,save)==1 || checkc(ac1,curnt)==1)

{

car.create();

clrscr();

cout<<"\n\n\n\n\n";

cout<<"\n\t\t\t ================================";

cout<<"\n\t\t\t|| Loan created Successfully ||";

cout<<"\n\t\t\t =================================";

message(1); //Calling the template functions .

}

else

cout<<"\nInvalid Account number";

getch();

clerk\_screen();

}

else

cout<<"Your salary is low. Can't give loan";

getch();

clerk\_screen();

break;

default:

cout<<"\n\t\t\tWrong choice";

getch();

clerk\_screen();

break;

}

break;

case 2:

clrscr();

cout<<"\n\n\t\t\t\tCREATING LOCKER :::\n\n\n";

cout<<"Enter Account number:";

char aacc[11];

gets(aacc);

if(checks(aacc,save)==1 || checkc(aacc,curnt)==1)

{

lock.create();

message(1); //Calling the template functions .

getch();

clerk\_screen();

}

else

cout<<"\n\t\t\tEntered Account number is incorrect";

getch();

clerk\_screen();

break;

case 3:

clrscr();

char accc[21];

int dd\_num;

float amount;

cout<<"\n\t\t\tEnter account number:";

gets(accc);

if(checks(accc,save)==1 || checkc(accc,curnt)==1)

{

cout<<"\nEnter DD number:";

cin>>dd\_num ;

cout<<"\nEnter Amount:";

cin>>amount;

dd.create();

save.deduct(accc,amount);

cout<<"\n\t\t\tEnter 1 to print DD:";

int n;

cin>>n;

cout<<"\n\t\t\tPrinting......";

delay(3000);

clrscr();

cout<<"\n\t\t\t ==============================";

cout<<"\n\t\t\t INDIAN BANK";

cout<<"\n\t\t\t ==================";

cout<<"\n\t\t\t DD no. :"<<dd\_num;

cout<<"\n\t\t\t Amount :"<<amount;

dd.print();

getch();

}

else

cout<<"\n\t\t\tEntered Account number is incorrect";

getch();

clerk\_screen();

break;

case 4:

clrscr();

cout<<"\n\t\t\tEnter 1 for Education Loan OR 2 for Car Loan:";

int cc;

cin>>cc;

switch(cc)

{

case 1:

clrscr();

edu.update();

clrscr();

cout<<"\n\n\n\n\n";

cout<<"\t\t\t ===============================================";

cout<<"\t\t\t|| Education Loan details updated Successfully ||";

cout<<"\t\t\t ===============================================";

message(1); //Calling the template functions .

getch();

clerk\_screen();

break;

case 2:

clrscr();

car.update();

clrscr();

cout<<"\n\n\n\n\n";

cout<<"\t\t\t ===============================================";

cout<<"\t\t\t|| Car Loan details updated Successfully ||";

cout<<"\t\t\t ===============================================";

message(1); //Calling the template functions .

getch();

clerk\_screen();

break;

default:

cout<<"\n\t\t\tWrong choice";

message(0); //Calling the template functions .

getch();

clerk\_screen();

break;

}

break;

case 5:

clrscr();

cout<<"\n\t\t\tEnter Locker Id:";

int lid;

cin>>lid;

if(lock.getlocknum()==lid)

{

lock.update();

clrscr();

cout<<"\n\n\n\n\n";

cout<<"\n\t\t\t ========================================";

cout<<"\n\t\t\t|| Locker details updated Successfully ||";

cout<<"\n\t\t\t ========================================";

message(1); //Calling the template functions .

getch();

clerk\_screen();

}

else

{

cout<<"\nSorry Invalid Locker Number";

getch();

clerk\_screen();

}

break;

case 6:

clrscr();

cout<<"\n\t\t\tEnter 1 for Savings Account OR 2 for Current Account:";

int chc;

cin>>chc;

switch(chc)

{

case 1:

clrscr();

cout<<"\n\t\t\tEnter Account Id:";

char acid3[11];

gets(acid3);

cout<<"\n\t\t\tEnter Balance to be deducted:";

float bal1;

cin>>bal1;

save.deduct(acid3,bal1);

clrscr();

cout<<"\n\n\n\n\n";

cout<<"\n\t\t\t =========================================";

cout<<"\n\t\t\t|| Account details updated Successfully ||";

cout<<"\n\t\t\t =========================================";

message(1); //Calling the template functions .

getch();

clerk\_screen();

break;

case 2:

clrscr();

cout<<"\n\t\t\tEnter Account Id:";

char acid4[11];

gets(acid4);

cout<<"\n\t\t\tEnter Balance to be deducted:";

float bal2;

cin>>bal2;

curnt.deduct(acid4,bal2);

clrscr();

cout<<"\n\n\n\n\n";

cout<<"\n\t\t\t =========================================";

cout<<"\n\t\t\t|| Account details updated Successfully ||";

cout<<"\n\t\t\t =========================================";

message(1); //Calling the template functions .

getch();

clerk\_screen();

break;

default:

cout<<"\n\t\t\tWrong choice";

message(0); //Calling the template functions .

getch();

clerk\_screen();

break;

}

break;

case 7:

clrscr();

cout<<"\n\t\t\tEnter 1 for Savings Account OR 2 for Current Account:";

int ccc;

cin>>ccc;

switch(ccc)

{

case 1:

clrscr();

cout<<"\n\t\t\tEnter Account Id:";

char acid5[11];

gets(acid5);

cout<<"\n\t\t\tEnter Balance to be credited:";

float bal3;

cin>>bal3;

save.credit(acid5,bal3);

clrscr();

cout<<"\n\n\n\n\n";

cout<<"\n\t\t\t =========================================";

cout<<"\n\t\t\t|| Account details updated Successfully ||";

cout<<"\n\t\t\t =========================================";

message(1); //Calling the template functions .

getch();

clerk\_screen();

break;

case 2:

clrscr();

cout<<"\n\t\t\tEnter Account Id:";

char acid6[11];

gets(acid6);

cout<<"\n\t\t\tEnter Balance to be credited:";

float bal4;

cin>>bal4;

curnt.credit(acid6,bal4);

clrscr();

cout<<"\n\n\n\n\n";

cout<<"\n\t\t\t =========================================";

cout<<"\n\t\t\t|| Account details updated Successfully ||";

cout<<"\n\t\t\t =========================================";

message(1); //Calling the template functions .

getch();

clerk\_screen();

break;

default:

cout<<"\n\t\t\tWrong choice";

message(0); //Calling the template functions .

getch();

clerk\_screen();

break;

}

break;

case 8:

clrscr();

int atmn;

int pin;

cout<<"\n\n\n\n";

cout<<"\n\t\t\t =================================";

cout<<"\n\t\t\t|| ||";

cout<<"\n\t\t\t|| Welcome To IND BANK ATM ||";

cout<<"\n\t\t\t|| ======================= ||";

cout<<"\n\t\t\t|| ||";

cout<<"\n\t\t\t =================================";

cout<<"\n\t\t\t\t\tEnter ATM Number:";

cin>>atmn;

cout<<"\n\t\t\t\t\tEnter PIN Number:";

cin>>pin;

if(atm.check\_details(atmn,pin)==1)

{

clrscr();

cout<<"\n\n\n\n\n";

cout<<"\n\t\t\t ============================================";

cout<<"\n\t\t\t|| 1.Are You Sure ? press 1 ||";

cout<<"\n\t\t\t|| 1.Exit press 2 . ||";

cout<<"\n\t\t\t ============================================";

cout<<"\n\t\t\t Enter your choice:";

int ds;

cin>>ds;

switch(ds)

{

case 1:

clrscr();

ATM\_transaction am;

atm=am;

cout<<"\n===========ATM deleted but Acount still Exists . Check Yourself .============ ";

cout<<"\n Thank You for using Indian Bank . ";

message(1); //Calling the template functions .

break;

case 2:

getch();

cust\_screen();

break;

default:

cout<<"\n Sorry wrong choice . ";

message(0); //Calling the template functions .

getch();

cust\_screen();

}

}

else

{

clrscr();

cout<<"\nData Not Found . ";

cout<<"\nTry Again!!!!!!!!!!!!!";

getch();

cust\_screen();

}

getch();

cust\_screen();

break;

case 9:

clrscr();

char aid[21];

cout<<"\n\n\n\n";

cout<<"\n\t\t\t ==========================================";

cout<<"\n\t\t\t|| ||";

cout<<"\n\t\t\t|| Welcome To IND BANK SERVICES ||";

cout<<"\n\t\t\t|| ======================= ||";

cout<<"\n\t\t\t|| ||";

cout<<"\n\t\t\t ==========================================";

cout<<"\n\t\t\t\t\tEnter Account Number:";

gets(aid);

if(strcmp(save.getid(),aid)==0)

{

clrscr();

cout<<"\n\n\n\n\n";

cout<<"\n\t\t\t ============================================";

cout<<"\n\t\t\t|| 1.Are You Sure ? press 1 ||";

cout<<"\n\t\t\t|| 1.Exit press 2 . ||";

cout<<"\n\t\t\t ============================================";

cout<<"\n\t\t\t Enter your choice:";

int ds;

cin>>ds;

switch(ds)

{

case 1:

clrscr();

Savings ss;

ATM\_transaction at;

atm=at;

save=ss;

cout<<"\n=========Account Deleted===========. ";

cout<<"\n Thank You for using Indian Bank . ";

message(1); //Calling the template functions .

cust\_screen();

break;

case 2:

getch();

cust\_screen();

break;

default:

cout<<"\n Sorry wrong choice . ";

message(0); //Calling the template functions .

getch();

cust\_screen();

}

}

else if(strcmp(curnt.getid(),aid)==0)

{

clrscr();

cout<<"\n\n\n\n\n";

cout<<"\n\t\t\t ============================================";

cout<<"\n\t\t\t|| 1.Are You Sure ? press 1 ||";

cout<<"\n\t\t\t|| 1.Exit press 2 . ||";

cout<<"\n\t\t\t ============================================";

cout<<"\n\t\t\t Enter your choice:";

int ds;

cin>>ds;

switch(ds)

{

case 1:

clrscr();

Current cc;

ATM\_transaction at;

atm=at;

curnt=cc;

cout<<"\n=========Account Deleted===========. ";

cout<<"\n Thank You for using Indian Bank . ";

message(1); //Calling the template functions .

cust\_screen();

break;

case 2:

getch();

cust\_screen();

break;

default:

cout<<"\n Sorry wrong choice . ";

message(0); //Calling the template functions .

getch();

cust\_screen();

}

}

else

{

clrscr();

cout<<"\nEntered Details are wrong. ";

cout<<"\nTry Again!!!!!!!!!!";

getch();

cust\_screen();

}

getch();

cust\_screen();

break;

case 10:

getch();

cust\_screen();

break;

case 11:

exit(0);

break;

default:

cout<<"\n\t\t\tEntered choice is wrong. Please try again.";

message(0); //Calling the template functions .

getch();

clerk\_screen();

}

}

~LOGIN(){}

};

// MAIN STARTS HERE

int ka=0;

void welcome();

void welcome\_screen();

void main()

{

clrscr();

if(ka==0)

{

welcome();

ka=1;

}

clrscr();

LOGIN login;

cout<<"\n\n\n\n";

clrscr();

cout<<"============Welcome to Our Bank ================";

cout<<"============Please First Create Your Account==========";

cout<<"\n\t\t\tEnter 1 for Savings OR 2 for Current:";

int ch;

char name[21];

char phone[10];

char userid[21];

char acc\_id[21];

int password;

float balance;

int choice;

int atmno;

int pin;

cin>>ch;

switch(ch)

{

case 1:

clrscr();

cout<<"\n Enter name:";

gets(name);

cout<<"\nEnter phone:";

gets(phone);

cout<<"\nEnter user id:";

gets(userid);

cout<<"\n Enter account id:";

gets(acc\_id);

cout<<"\n Enter password:";

cin>>password;

cout<<"\n Enter balance:";

cin>>balance;

//EXCEPTIONAL HANDLING 10

/\*

try{

if(strlen(name)>21 || strlen(phone)>10 || strlen(phone)<10 || strlen(userid)>21 || strlen(acc\_id)>21 || password<0 || balance<0 )

throw myException("Exception Caught .Try Again!!!!!!!!!!!!!!!!!!!!!");

}

catch(...) //CAUGHT ALL EXCEPTIONS .

{

cout<<e.message();

exit 0;

}

\*/

Savings ss(password,acc\_id,name,phone,balance,userid);

login.assigns(ss);

cout<<"\nEnter 1 if ATM card exist else enter 2:";

cin>>choice;

while(choice==1)

{

cout<<"\nEnter ATM number:";

cin>>atmno;

cout<<"\nEnter ATM pin:";

cin>>pin ;

choice=2; //for out of the loop ;

//EXCEPTIONAL HANDLING 11

/\*

try{

if(atmno<0 || pin<0)

throw myException("Exception Caught .Try Again!!!!!!!!!!!!!!!!!!!!!");

}

catch(...) //CAUGHT ALL EXCEPTIONS .

{

cout<<e.message();

choice=1;

}

\*/

ATM\_transaction at(atmno,pin,acc\_id);

login.assigna(at);

}

if(choice==2)

cout<<"\n\t\t ATM created.";

else

cout<<"\n\t\t ATM not created";

break;

case 2:

clrscr();

cout<<"\n Enter name:";

gets(name);

cout<<"\nEnter phone:";

gets(phone);

cout<<"\nEnter user id:";

gets(userid);

cout<<"\n Enter account id:";

gets(acc\_id);

cout<<"\n Enter password:";

cin>>password;

cout<<"\n Enter balance:";

cin>>balance;

//EXCEPTIONAL HANDLING 12

/\*

try{

if(strlen(name)>21 || strlen(phone)>10 || strlen(phone)<10 || strlen(userid)>21 || strlen(acc\_id)>21 || password<0 || balance<0 )

throw myException("Exception Caught .Try Again!!!!!!!!!!!!!!!!!!!!!");

}

catch(...) //CAUGHT ALL EXCEPTIONS .

{

cout<<e.message();

exit 0;

}

\*/

Current cc(password,acc\_id,name,phone,balance,userid);

login.assignc(cc);

cout<<"\nEnter 1 if ATM card exist else enter 2:";

cin>>choice;

while(choice==1)

{

cout<<"\nEnter ATM number:";

cin>>atmno;

cout<<"\nEnter ATM pin:";

cin>>pin ;

choice=2; //for out of the loop

//EXCEPTIONAL HANDLING 13

/\*

try{

if(atmno<0 || pin<0)

throw myException("Exception Caught .Try Again!!!!!!!!!!!!!!!!!!!!!");

}

catch(...) //CAUGHT ALL EXCEPTIONS .

{

cout<<e.message();

choice=1;

}

\*/

ATM\_transaction at(atmno,pin,acc\_id);

login.assigna(at);

}

if(choice==2)

cout<<"\n\t\tATM created";

else

cout<<"\n\t\tATM not created";

break;

default:

cout<<"\n\t\t\tWrong Choice.";

break;

}

cout<<"\n\t\t\t =================================";

cout<<"\n\t\t\t|| ||";

cout<<"\n\t\t\t|| Welcome To IOB Bank ATM ||";

cout<<"\n\t\t\t|| ======================= ||";

cout<<"\n\t\t\t|| ||";

cout<<"\n\t\t\t|| 1.Customer ||";

cout<<"\n\t\t\t|| 2.Clerk ||";

cout<<"\n\t\t\t|| 3.Exit ||";

cout<<"\n\t\t\t|| ||";

cout<<"\n\t\t\t =================================";

cout<<"\n\n\t\t\t Enter Choice:";

int ch1;

cin>>ch1;

switch(ch1)

{

case 1:

login.cust\_screen();

break;

case 2:

login.clerk\_screen();

break;

case 3:

exit(0);

default :

cout<<"\n\t\t\tEntered choice is not valid.";

getch();

main();

break;

}

getch();

}

void welcome()

{

int gd=DETECT,gm;

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

welcome\_screen();

}

void intro(void){

void \*image;

int size;

int dr=9,mode=2;

initgraph(&dr,&mode,"..\\bgi");

size=imagesize(140,140,500,250);

image=malloc(size);

setfillstyle(SOLID\_FILL,GREEN);

circle(200,200,50);

floodfill(200,200,WHITE);

setcolor(LIGHTGRAY);

circle(200,200,19);

setcolor(WHITE);

circle(200,200,18);

circle(200,200,49);

circle(200,200,53);

circle(200,200,20);

arc(215,215,350,90,30);

arc(210,182,90,194,30);

arc(180,195,180,300,30);

settextstyle(1,HORIZ\_DIR,1);

setcolor(LIGHTGRAY);

outtextxy(270,180,"National Bank");

outtextxy(270,185,"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_");

outtextxy(270,210,"Of India Pvt Ltd.");

settextstyle(0,HORIZ\_DIR,0);

setcolor(DARKGRAY);

outtextxy(270,240,"COPYRIGHT 2014");

getimage(141,141,499,259,image);

cleardevice();

for(int count=1;count<300;count+=2)

putimage(1+count,100,image,COPY\_PUT);

for(int down=0;down<640;++down)

{

delay(5);

line(1,220,1+down,220);

}

/\*setcolor(BLUE);

for(int bottom=0;bottom<300;bottom+=4) //BOTTOM LINES ARE CANCELLED .

line(1,220+bottom,640,220+bottom); \*/

free(image);

// closegraph(); //dont here otherwise next graph also not executed .

}

void welcome\_screen()

{

intro();

delay(500);

setcolor(GREEN);

settextstyle(7,0,6);

outtextxy(30,220,"A Project By:-");

outtextxy(30,260,"BURHANUDDIN");

outtextxy(30,300,"BHOPAL WALA");

outtextxy(30,340,"AND HIS");

outtextxy(30,380,"COLLEAGUES.");

delay(5000);

closegraph();

}