**Project : WBS Bank**

Submitted To:- **Submitted By:-**

Tribhuwan Kumar Tewari Sir Karan Dua(11103485)

CSE Department Niket Agarwal(11103512)

Rajat Bhardwaj(11103540)

Deepti Agarwal(11103466)

**Batch – B6**

**Project Code:- B607**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

#include<dos.h>

#include<stdlib.h>

#include<iostream.h>

#include<fstream.h>

#include<string.h>

class ATM

{

protected:

int atm\_num;

int password;

char accd[11];

public:

ATM()

{

}

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

{

fflush(stdin);

atm\_num=n;

password=p;

strcpy(accd,c);

}

int getatm()

{

return atm\_num;

}

char\* getatmid()

{

return accd;

}

int getpass()

{

return password;

}

};

class Account

{

protected:

int password ;

char acc\_id[21] ;

char name[21] ;

char phone[21] ;

float balance ;

char userid[21] ;

ATM \*a ;

float rate\_interest ;

char transaction\_date [10];

int number\_of\_trans ;

public :

virtual char\* getid()

{

}

virtual void create()

{

}

virtual void deduct(char[] , float)

{

}

virtual void credit(char[] , float)

{

}

virtual void atm\_exist()

{

}

virtual void details()

{

}

virtual void write(int)

{

}

virtual void init\_number\_of\_transac()

{

}

virtual void init\_rate\_of\_interest()

{

}

};

class Savings :virtual public Account

{

public:

char\* getid()

{

return acc\_id;

}

void init\_rate\_of\_interest()

{

cout<<"Enter rate of interest:";

cin>>rate\_interest;

}

void init\_number\_of\_transac()

{

number\_of\_trans=0;

}

void create()

{

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;

atm\_exist();

}

void deduct(char c[],float ded)

{

// file format- "accid"\t"date"\t"transactions"\n (without quotes)

char file\_accid[11];

char file\_date[11];

int file\_transactions;

char today\_date[11];

cout<<endl<<"Enter the today's date:";

gets(today\_date);

int flag=0;

// 10 is length of accid

fstream file;

fstream temp;

file.open("deduct.pro",ios::in);

temp.open("temp.pro",ios::out);

while(file)

{

file>>file\_accid;

file>>file\_date;

if(strcmp(file\_accid,c)==0)

{

if(strcmp(file\_date,today\_date)==0)

{

file>>file\_transactions;

if(file\_transactions==3)

{

cout<<endl<<"Cannot deduct.";

temp<<file\_accid<<"\t"<<file\_date<<"\t"<<file\_transactions<<"\n";

}

else

temp<<file\_accid<<"\t"<<today\_date<<"\t"<<(file\_transactions+1)<<"\n";

}

else

temp<<file\_accid<<"\t"<<today\_date<<"\t"<<(file\_transactions+1)<<"\n";

flag=1;

}

else

temp<<file\_accid<<"\t"<<file\_date<<"\t"<<file\_transactions<<"\n";

}

if(flag==0)

{

cout<<"\n\t\t\tAccount ID not found.";

}

file.close();

temp.close();

file.open("deduct.pro",ios::out|ios::trunc);

temp.open("temp.pro",ios::in);

//adding data back to deduct file.

while(temp)

{

temp>>file\_accid;

temp>>file\_date;

temp>>file\_transactions;

file<<file\_accid<<"\t"<<file\_date<<"\t"<<file\_transactions<<"\n";

}

file.close();

temp.close();

Account \*a\_temp;

a\_temp=new Savings;

flag=0;

file.open("save.pro",ios::binary|ios::in|ios::out);

temp.open("temp.pro",ios::binary|ios::out|ios::trunc);

while(file)

{

file.read((char\*)a\_temp,sizeof(Account));

if(strcmp(a\_temp->getid(),c)==0) //chk error for return type of getid() c first arg

{

a\_temp->write(ded); //second arg

flag=1;

}

else

continue;

temp.write((char\*)a\_temp,sizeof(Account));

}

file.close();

temp.close();

if(flag==0)

cout<<"\n\t\tCANNOT FIND ACCOUNT ID IN SAVE ENTRY FILE.";

file.open("save.pro",ios::out|ios::trunc);

temp.open("temp.pro",ios::in);

// move data back to original save\_entry file from tempfile.

while(temp)

{

temp.read((char\*)a\_temp,sizeof(Account));

file.write((char\*)a\_temp,sizeof(Account));

}

file.close();

temp.close();

// to delete all data from temp file.security purpose.

temp.open("temp.pro",ios::trunc);

temp.close();

}

void credit(char c[], float add)

{

fstream file;

fstream temp;

Account \*a\_temp;

a\_temp=new Savings;

float k=-add;

file.open("save.pro",ios::binary|ios::in);

temp.open("temp.pro",ios::binary|ios::out);

while(file)

{

file.read((char\*)a\_temp,sizeof(Account));

if(strcmp(a\_temp->getid(),c)==0)

a\_temp->write(k);

else

continue;

temp.write((char\*)a\_temp,sizeof(Account));

}

file.close();

temp.close();

file.open("save.pro",ios::out|ios::trunc);

temp.open("temp.pro",ios::in);

// move data back to original save\_entry file from tempfile.

while(temp)

{

temp.read((char\*)a\_temp,sizeof(Account));

file.write((char\*)a\_temp,sizeof(Account));

}

file.close();

temp.close();

// to delete all data from temp file.security purpose.

temp.open("temp.pro",ios::trunc);

temp.close();

}

void atm\_exist()

{

int choice;

int atmno;

int pin;

cout<<"\n\t\tEnter 1 if ATM card exist else enter 2:";

cin>>choice;

if(choice==1)

{

cout<<"\nEnter ATM number:";

cin>>atmno;

cout<<"\nEnter ATM pin";

cin>>pin ;

a=new ATM(atmno,pin,acc\_id);

fstream file;

file.open("ATM\_entry.pro",ios::binary|ios::app);

file.write((char\*)a,sizeof(ATM));

file.close();

}

else

cout<<"\n\t\tAccount not created.";

}

void details()

{

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

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

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

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

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

}

void write(int t)

{

balance=balance-t;

}

};

class Current :virtual public Account

{

public:

char\* getid()

{

return acc\_id;

}

void init\_rate\_of\_interest()

{

cout<<"Enter rate of interest:";

cin>>rate\_interest;

}

void init\_number\_of\_transac()

{

number\_of\_trans=0;

}

void create()

{

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;

atm\_exist();

}

void deduct(char c[] , float ded)

{

// file format- "accid"\t"date"\t"transactions"\n (without quotes)

char file\_accid[11];

char file\_date[11];

int file\_transactions;

char today\_date[11];

cout<<endl<<"Enter the today's date:";

gets(today\_date);

int flag=0;

// 10 is length of accid

fstream file;

fstream temp;

file.open("deduct.pro",ios::in);

temp.open("temp.pro",ios::out);

while(file)

{

file>>file\_accid;

file>>file\_date;

if(strcmp(file\_accid,c)==0)

{

if(strcmp(file\_date,today\_date)==0)

{

file>>file\_transactions;

if(file\_transactions==3)

{

cout<<endl<<"Cannot deduct.";

temp<<file\_accid<<"\t"<<file\_date<<"\t"<<file\_transactions<<"\n";

}

else

temp<<file\_accid<<"\t"<<today\_date<<"\t"<<(file\_transactions+1)<<"\n";

}

else

temp<<file\_accid<<"\t"<<today\_date<<"\t"<<(file\_transactions+1)<<"\n";

flag=1;

}

else

temp<<file\_accid<<"\t"<<file\_date<<"\t"<<file\_transactions<<"\n";

}

if(flag==0)

cout<<"\n\t\t CANNOT FIND ACCOUNT ID.";

file.close();

temp.close();

file.open("deduct.pro",ios::out|ios::trunc);

temp.open("temp.pro",ios::in);

//adding data back to deduct file.

while(temp)

{

temp>>file\_accid;

temp>>file\_date;

temp>>file\_transactions;

file<<file\_accid<<"\t"<<file\_date<<"\t"<<file\_transactions<<"\n";

}

file.close();

temp.close();

Account \*a\_temp;

a\_temp=new Current;

file.open("curr.pro",ios::binary|ios::in|ios::out);

temp.open("temp.pro",ios::binary|ios::out|ios::trunc);

while(file)

{

file.read((char\*)a\_temp,sizeof(Account));

if(strcmp(a\_temp->getid(),c)==0) //chk error for return type of getid()

a\_temp->write(ded); //second arg

else

continue;

temp.write((char\*)a\_temp,sizeof(Account));

}

file.close();

temp.close();

file.open("curr.pro",ios::out|ios::trunc);

temp.open("temp.pro",ios::in);

// move data back to original curr file from tempfile.

while(temp)

{

temp.read((char\*)a\_temp,sizeof(Account));

file.write((char\*)a\_temp,sizeof(Account));

}

file.close();

temp.close();

// to delete all data from temp file.security purpose.

temp.open("temp.pro",ios::trunc);

temp.close();

}

void credit(char c[], float add)

{

fstream file;

fstream temp;

Account \*a\_temp;

a\_temp=new Current;

float k=-add;

file.open("curr.pro",ios::binary|ios::in);

temp.open("temp.pro",ios::binary|ios::out);

while(file)

{

file.read((char\*)a\_temp,sizeof(Account));

if(strcmp(a\_temp->getid(),c)==0)

a\_temp->write(k);

else

continue;

temp.write((char\*)a\_temp,sizeof(Account));

}

file.close();

temp.close();

file.open("curr.pro",ios::out|ios::trunc);

temp.open("temp.pro",ios::in);

// move data back to original curr file from tempfile.

while(temp)

{

temp.read((char\*)a\_temp,sizeof(Account));

file.write((char\*)a\_temp,sizeof(Account));

}

file.close();

temp.close();

// to delete all data from temp file.security purpose.

temp.open("temp.pro",ios::trunc);

temp.close();

}

void atm\_exist()

{

int choice;

int atmno;

int pin;

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

cin>>choice;

if(choice==1)

{

cout<<"\nEnter ATM number:";

cin>>atmno;

cout<<"\nEnter ATM pin";

cin>>pin ;

a=new ATM(atmno,pin,acc\_id);

fstream file;

file.open("ATM\_entry.pro",ios::binary|ios::app);

file.write((char\*)&a,sizeof(a));

file.close();

}

else

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

}

void details()

{

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

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

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

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

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

}

void write(int t)

{

balance=balance-t;

}

};

class Locker : virtual public Savings , virtual public Current

{

protected :

int locker\_num ;

char type[10] ;

char acc\_id[21] ;

float rent ;

char date[21] ;

public :

int getlocknum()

{

return locker\_num;

}

void create(char aid[])

{

if(check(aid)==1)

{

clrscr();

cout<<"\nEnter locker number:";

cin>>locker\_num;

cout<<"\nEnter type:";

gets(type);

cout<<"\nEnter Rent:";

cin>>rent;

strcpy(aid,acc\_id);

cout<<"\nEnter Date:";

gets(date);

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

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

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

}

else

cout<<endl<<"\n\t\t\tCannot create locker.Savings ID entered does not exist.";

}

int check(char aid[])

{

fstream file;

Account \*a\_temp;

a\_temp=new Savings;

file.open("save.pro",ios::binary|ios::in);

while(file)

{

file.read((char\*)a\_temp,sizeof(Account));

if(strcmp(a\_temp->getid(),aid)==0)

{

return 1;

}

}

return 0;

}

void details()

{

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

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

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

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

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

}

void update()

{

Account \*a\_temp;

a\_temp=new Savings;

fstream file;

fstream temp;

file.open("save.pro",ios::binary|ios::in|ios::out);

temp.open("temp.pro",ios::binary|ios::out);

while(file)

{

file.read((char\*)a\_temp,sizeof(Account));

if(strcmp(a\_temp->getid(),acc\_id)==0) //chk error for return type of getid() acc\_id var in class locker

{

a\_temp->write(rent); //rent var in class locker

}

else

continue;

temp.write((char\*)a\_temp,sizeof(Account));

}

file.close();

temp.close();

file.open("save.pro",ios::out|ios::trunc);

temp.open("temp.pro",ios::in);

// move data back to original save\_entry file from tempfile.

while(temp)

{

temp.read((char\*)a\_temp,sizeof(Account));

file.write((char\*)a\_temp,sizeof(Account));

}

file.close();

temp.close();

// to delete all data from temp file.security purpose.

temp.open("temp.pro",ios::trunc);

temp.close();

}

};

class LOAN : virtual public Savings , virtual public Current

{

protected :

char loan\_id[21] ;

char acc\_id[21] ;

float rate\_interest ;

int num\_of\_Installments ;

int installments\_paid ;

float principle ;

float amount\_paid ;

float fixed\_amt;

char date[21] ;

public :

char\* getlid()

{

return loan\_id;

}

virtual void update()

{

}

virtual void details()

{

}

virtual void create()

{

}

virtual int check(char [])

{

}

virtual int check\_marks(float)

{

}

virtual int check\_salary(float)

{

}

};

class CAR :virtual public LOAN

{

protected :

float salary ;

public :

void create()

{

clrscr();

cout<<"\nEnter loan id:";

gets(loan\_id) ;

cout<<"\nEnter accoutnt id:" ;

gets( acc\_id) ;

cout<<"\nEnter Loan amount:";

cin>>principle ;

cout<<"\nEnter Rate of interest(p/a):";

cin>>rate\_interest ;

cout<<"\nEnter number of installments:";

cin>>num\_of\_Installments ;

installments\_paid=0 ;

amount\_paid=0 ;

cout<<"\nEnter date:";

gets(date) ;

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

cin>>fixed\_amt;

}

int check\_salary(float sal)

{

salary=sal;

if(salary>=200)

return 1;

else

return 0;

}

void update()

{

//UPDATE SAVE ENTRY FILE SUBTRACT LOAN PRICE FROM ACCOUNT.

Account \*a\_temp;

a\_temp=new Savings;

fstream file2;

fstream temp;

float t;

t=fixed\_amt+(rate\_interest\*principle);

file2.open("save.pro",ios::binary|ios::in);

temp.open("temp.pro",ios::binary|ios::out|ios::trunc);

while(file2)

{

file2.read((char\*)a\_temp,sizeof(Account));

if(strcmp(a\_temp->getid(),acc\_id)==0) // acc\_id var in class locker

{

//cout<<"in update loan car";

//getch();

a\_temp->write(t); //rent var in class loan

temp.write((char\*)a\_temp,sizeof(Account));

//f=1;

break;

}

else

temp.write((char\*)a\_temp,sizeof(Account));

}

file2.close();

temp.close();

file2.open("save.pro",ios::binary|ios::out|ios::trunc);

temp.open("temp.pro",ios::binary|ios::in);

// move data back to original save\_entry file from tempfile.

while(temp)

{

temp.read((char\*)a\_temp,sizeof(Account));

file2.write((char\*)a\_temp,sizeof(Account));

}

file2.close();

temp.close();

//CHANGING LOAN FILE NUMBER OF INSTALLMENTS.

installments\_paid++;

// to delete all data from temp file.security purpose.

temp.open("temp.pro",ios::trunc);

temp.close();

}

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;

}

int check(char aid[])

{

fstream file;

Account \*a\_temp;

a\_temp=new Savings;

file.open("save.pro",ios::binary|ios::in);

while(file)

{

file.read((char\*)a\_temp,sizeof(Account));

if(strcmp(a\_temp->getid(),aid)==0)

{

return 1;

}

}

return 0;

}

};

class EDUCATION :virtual public LOAN

{

protected :

float twelth\_marks ;

public :

void create()

{

clrscr();

cout<<"\nEnter loan id:";

gets(loan\_id) ;

cout<<"\nEnter accoutnt id:";

gets( acc\_id) ;

cout<<"\nEnter Loan amount:";

cin>>principle ;

cout<<"\nEnter Rate of interest(p/a):";

cin>>rate\_interest ;

cout<<"\nEnter number of installments";

cin>>num\_of\_Installments ;

installments\_paid=0 ;

amount\_paid=0 ;

cout<<"\nEnter date:";

gets(date) ;

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

cin>>fixed\_amt;

}

int check\_marks(float mar)

{

twelth\_marks=mar;

if(twelth\_marks>=80)

return 1;

else

return 0;

}

void update()

{

//UPDATE SAVE ENTRY FILE . DEDUCTING LOAN PRICE.

Account \*a\_temp;

a\_temp=new Savings;

fstream file;

float t;

t=fixed\_amt+(rate\_interest\*principle);

fstream temp;

file.open("save.pro",ios::binary|ios::in);

temp.open("temp.pro",ios::binary|ios::out);

while(file)

{

file.read((char\*)a\_temp,sizeof(Account));

if(strcmp(a\_temp->getid(),acc\_id)==0) //chk error for return type of getid() acc\_id var in class locker

{

a\_temp->write(t); //rent var in class locker

}

else

continue;

temp.write((char\*)a\_temp,sizeof(Account));

}

file.close();

temp.close();

file.open("save.pro",ios::binary|ios::out|ios::trunc);

temp.open("temp.pro",ios::binary|ios::in);

// move data back to original save\_entry file from tempfile.

while(temp)

{

temp.read((char\*)a\_temp,sizeof(Account));

file.write((char\*)a\_temp,sizeof(Account));

}

file.close();

temp.close();

//CHANGING LOAN FILE NUMBER OF INSTALLMENTS.

installments\_paid++;

// to delete all data from temp file.security purpose.

temp.open("temp.pro",ios::trunc);

temp.close();

}

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;

}

int check(char aid[])

{

fstream file;

Account \*a\_temp;

a\_temp=new Savings;

file.open("save.pro",ios::binary|ios::in);

while(file)

{

file.read((char\*)a\_temp,sizeof(Account));

if(strcmp(a\_temp->getid(),aid)==0)

{

return 1;

}

}

return 0;

}

};

class DD : virtual public Savings , virtual public Current

{

protected :

int dd\_num ;

char date[21] ;

float amount ;

char from[21] ;

char to[21] ;

char acc\_id[21] ;

public :

void create()

{

clrscr();

cout<<"\nEnter DD number:";

cin>>dd\_num ;

cout<<"\nEnter Amount:";

cin>>amount ;

cout<<"\nEnter account id from where balance is to be deducted:";

gets(acc\_id) ;

cout<<"\nEnter Date:";

gets(date) ;

cout<<"\nEnter From:";

gets(from) ;

cout<<"\nEnter To:";

gets(to) ;

//CALLING DEDUCT FUNCTION.

acc\_sub(acc\_id,amount);

}

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

void print()

{

clrscr();

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

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

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

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

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

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

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 ==============================";

}

int check(char ch[])

{

fstream file;

Account \*a\_temp;

a\_temp=new Savings;

file.open("save.pro",ios::binary|ios::in);

while(file)

{

file.read((char\*)a\_temp,sizeof(Account));

if(strcmp(a\_temp->getid(),ch)==0)

{

return 1;

}

}

return 0;

}

void acc\_sub(char ch[],int amt)

{

Account \*s\_t;

s\_t=new Savings;

fstream file;

fstream temp;

if(check(ch)==1)

{

file.open("save.pro",ios::binary|ios::in);

temp.open("temp.pro",ios::binary|ios::out);

while(file)

{

file.read((char\*)s\_t,sizeof(Account));

if(strcmp(s\_t->getid(),ch)==0)

{

s\_t->write(amt);

temp.write((char\*)s\_t,sizeof(Account));

}

else

temp.write((char\*)s\_t,sizeof(Account));

}

file.close();

temp.close();

file.open("save.pro",ios::binary|ios::out|ios::trunc);

temp.open("temp.pro",ios::binary|ios::in);

while(temp)

{

temp.read((char\*)s\_t,sizeof(Account));

file.write((char\*)s\_t,sizeof(Account));

}

file.close();

temp.close();

temp.open("temp.pro",ios::trunc);

temp.close();

}

else

cout<<"\n\t\tACCOUNT ID NOT FOUND. PAY BY CASH.";

}

};

class ATM\_transaction : public ATM , virtual public Savings , virtual public Current

{

protected :

int check\_pass ;

char atmid[21];

public :

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

{

fstream file;

//atmnum=a\_num;

ATM \*a;

a=new ATM;

file.open("ATM\_entry.pro",ios::binary|ios::in);

while(file)

{

file.read((char\*)a,sizeof(ATM));

if(a->getatm()==a\_num && a->getpass()==pwd)

{

strcpy(atmid,a->getatmid());

return 1;

}

else

return 0;

}

file.close();

}

void deduct\_balance(float bal)

{

int a;

Account \*acc;

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

cin>>a;

switch(a)

{

case 1:

acc=new Savings();

acc->deduct(atmid,bal);

break;

case 2:

acc=new Current();

acc->deduct(atmid,bal);

break;

default:

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

getch();

deduct\_balance(bal);

break;

}

}

void details()

{

fstream file;

int flag=0;

Account \*s\_a;

Account \*c\_a;

s\_a=new Savings;

c\_a=new Current;

file.open("save.pro",ios::binary|ios::in);

while(file)

{

file.read((char\*)s\_a,sizeof(Account));

if(strcmp(s\_a->getid(),atmid)==0)

{

flag=1;

s\_a->details();

break;

}

else

continue;

}

file.close();

if(flag==0)

{

file.open("curr.pro",ios::binary|ios::in);

while(file)

{

file.read((char\*)c\_a,sizeof(Account));

if(strcmp(c\_a->getid(),atmid)==0)

{

flag=1;

c\_a->details();

break;

}

else

continue;

}

}

if(flag==0)

cout<<"\n\t\t\tAccount ID not found.";

}

};

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

{

protected :

char user\_id[21] ;

char password[21] ;

char acc\_id[21] ;

public :

void cust\_screen()

{

clrscr();

fstream file;

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

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

cout<<"\n\t\t\t|| Welcome To IOB 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.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 atm;

int pin;

ATM\_transaction a;

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

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 =================================";

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

cin>>atm;

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

cin>>pin;

if(a.check\_details(atm,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.Details ||";

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

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

int ds;

cin>>ds;

switch(ds)

{

case 1:

clrscr();

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

float ba;

cin>>ba;

a.deduct\_balance(ba);

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 ==========================================";

getch();

cust\_screen();

break;

case 2:

clrscr();

a.details();

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\t ==========================================";

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 ==========================================";

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);

// fstream file;

Account \*a\_temp1;

a\_temp1=new Savings;

file.open("save.pro",ios::binary|ios::in);

while(file)

{

file.read((char\*)a\_temp1,sizeof(Account));

if(strcmp(a\_temp1->getid(),accid)==0)

{

a\_temp1->details();

flag2=1;

break;

}

else

continue;

}

file.close();

Account \*c\_temp1;

c\_temp1=new Current;

file.open("curr.pro",ios::binary|ios::in);

if(flag2==0)

{

while(file)

{

file.read((char\*)c\_temp1,sizeof(Account));

if(strcmp(c\_temp1->getid(),accid)==0)

{

c\_temp1->details();

flag2=1;

break;

}

else

continue;

}

}

if(flag2==0)

cout<<"Data not found.";

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);

LOAN \*c\_temp2;

c\_temp2=new CAR;

// fstream file;

file.open("car\_loan.pro",ios::binary|ios::in);

while(file)

{

file.read((char\*)c\_temp2,sizeof(LOAN));

if(strcmp(c\_temp2->getlid(),lid)==0)

{

c\_temp2->details();

flag3=1;

break;

}

else

continue;

}

file.close();

LOAN \*e\_temp;

e\_temp=new EDUCATION;

file.open("edu\_loan.pro",ios::binary|ios::in);

if(flag3==0)

{

while(file)

{

file.read((char\*)e\_temp,sizeof(LOAN));

if(strcmp(e\_temp->getlid(),lid)==0)

{

e\_temp->details();

flag3=1;

break;

}

else

continue;

}

}

if(flag3==0)

cout<<"Data not found.";

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;

Locker \*l\_temp;

// fstream file;

l\_temp=new Locker;

file.open("locker.pro",ios::binary|ios::in);

while(file)

{

file.read((char\*)l\_temp,sizeof(Locker));

if(l\_temp->getlocknum()==lnum)

{

l\_temp->details();

flag4=1;

break;

}

else

{

continue;

}

}

file.close();

if(flag4==0)

cout<<"Data not found.";

getch();

cust\_screen();

break;

case 5:

clrscr();

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

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

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

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

getch();

main();

break;

default:

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

getch();

cust\_screen();

}

getch();

}

void clerk\_screen()

{

clrscr();

fstream file;

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

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

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

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

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

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

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

cout<<"\n\t\t\t|| 4.Create DD ||";

cout<<"\n\t\t\t|| 5.Account Details ||";

cout<<"\n\t\t\t|| 6.Locker Details ||";

cout<<"\n\t\t\t|| 7.Loan Details ||";

cout<<"\n\t\t\t|| 8.Update Locker ||";

cout<<"\n\t\t\t|| 9.Update Loan ||";

cout<<"\n\t\t\t|| 10.Deduct Balance ||";

cout<<"\n\t\t\t|| 11.Credit Balance ||";

cout<<"\n\t\t\t|| 12.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();

Account \*acc;

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

int ch;

cin>>ch;

switch(ch)

{

case 1:

clrscr();

acc= new Savings;

acc->create();

acc->init\_rate\_of\_interest();

acc->init\_number\_of\_transac();

fstream file\_save;

fstream ded;

ded.open("deduct.pro",ios::binary|ios::app);

file\_save.open("save.pro",ios::binary|ios::app);

file\_save.write((char\*)acc,sizeof(Account));

ded<<acc->getid()<<"\t"<<"00/00/0000"<<"\t"<<"0"<<"\n";

file\_save.close();

ded.close();

clrscr();

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

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

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

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

getch();

clerk\_screen();

break;

case 2:

clrscr();

acc= new Current();

acc->create();

acc->init\_rate\_of\_interest();

acc->init\_number\_of\_transac();

fstream ded1;

ded1.open("deduct.pro",ios::binary|ios::app);

file.open("curr.pro",ios::binary|ios::app);

file.write((char\*)acc,sizeof(Account));

ded1<<acc->getid()<<"\t"<<"00/00/0000"<<"\t"<<"0"<<"\n";

file.close();

ded1.close();

clrscr();

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

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

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

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

getch();

clerk\_screen();

break;

default:

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

getch();

clerk\_screen();

break;

}

break;

case 2:

clrscr();

//LOAN \*ln;

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

int ch1;

cin>>ch1;

switch(ch1)

{

case 1:

clrscr();

LOAN \*ln;

ln=new EDUCATION();

char ac[11];

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

float mark;

cin>>mark;

if(ln->check\_marks(mark))

{

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

gets(ac);

if(ln->check(ac))

{

ln->create();

// fstream file;

file.open("edu\_loan.pro",ios::binary|ios::app);

file.write((char\*)ln,sizeof(LOAN));

file.close(); 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 =================================";

}

else

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

getch();

clerk\_screen();

}

else

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

getch();

clerk\_screen();

break;

case 2:

clrscr();

LOAN \*ln1;

ln1=new CAR();

char ac1[11];

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

float sal;

cin>>sal;

if(ln1->check\_salary(sal))

{

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

gets(ac1);

if(ln1->check(ac1))

{

ln1->create();

// fstream file;

file.open("car\_loan.pro",ios::binary|ios::app);

file.write((char\*)ln1,sizeof(LOAN));

file.close();

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 =================================";

}

else

cout<<"Invalid 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 3:

clrscr();

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

cout<<"Enter Account number:";

char aacc[11];

gets(aacc);

Locker \*l;

l=new Locker;

l->create(aacc);

file.open("Locker.pro",ios::binary|ios::app);

file.write((char\*)l,sizeof(Locker));

file.close();

getch();

clerk\_screen();

break;

case 4:

clrscr();

DD d;

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

char accc[11];

gets(accc);

if(d.check(accc)==1)

{

d.create();

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

int n;

cin>>n;

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

delay(3000);

d.print();

getch();

}

else

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

getch();

clerk\_screen();

break;

case 5:

clrscr();

int flag2=0;

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

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

char accid[11];

gets(accid);

fstream file12;

Account \*a\_temp;

a\_temp=new Savings;

file12.open("save.pro",ios::binary|ios::in);

while(file12)

{

//cout<<"\nflag 1";

file12.read((char\*)a\_temp,sizeof(Account));

//cout<<a\_temp.getid();

if(strcmp(a\_temp->getid(),accid)==0)

{ //cout<<"\nflag 2";

a\_temp->details();

flag2=1;

break;

}

else

continue;

}

file12.close();

Account \*c\_temp2;

c\_temp2=new Current;

file12.open("curr.pro",ios::binary|ios::in);

if(flag2==0)

{ //cout<<"\nflag 3";

while(file12)

{ // cout<<"\nflag 4";

file12.read((char\*)c\_temp2,sizeof(Account));

if(strcmp(c\_temp2->getid(),accid)==0)

{

c\_temp2->details();

flag2=1;

break;

}

else

continue;

}

}

file12.close();

if(flag2==0)

cout<<"Data not found.";

getch();

clerk\_screen();

break;

case 7:

clrscr();

int flag3=0;

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

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

char lid[21];

gets(lid);

LOAN \*c\_temp;

c\_temp=new CAR;

// fstream file;

file.open("car\_loan.pro",ios::binary|ios::in);

while(file)

{

file.read((char\*)c\_temp,sizeof(LOAN));

if(strcmp(c\_temp->getlid(),lid)==0)

{

c\_temp->details();

flag3=1;

break;

}

else

continue;

}

file.close();

LOAN \*e\_temp;

e\_temp=new EDUCATION;

file.open("edu\_loan.pro",ios::binary|ios::in);

if(flag3==0)

{

while(file)

{

file.read((char\*)e\_temp,sizeof(LOAN));

if(strcmp(e\_temp->getlid(),lid)==0)

{

e\_temp->details();

flag3=1;

break;

}

else

continue;

}

}

if(flag3==0)

cout<<"Data not found.";

getch();

clerk\_screen();

break;

case 6:

clrscr();

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

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

int lnum;

int flag4=0;

cin>>lnum;

Locker l\_temp;

// fstream file;

file.open("locker.pro",ios::binary|ios::in);

while(file)

{

file.read((char\*)&l\_temp,sizeof(Locker));

if(l\_temp.getlocknum()==lnum)

{

l\_temp.details();

flag4=1;

break;

}

else

{

continue;

}

}

file.close();

if(flag4==0)

cout<<"Data not found.";

getch();

clerk\_screen();

break;

case 8:

clrscr();

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

int lx;

cin>>lx;

Locker \*l1;

l1=new Locker;

file.open("Locker.pro",ios::binary|ios::in);

while(file)

{

file.read((char\*)l1,sizeof(Locker));

if(l1->getlocknum()==lx)

l1->update();

}

file.close();

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 ========================================";

getch();

clerk\_screen();

break;

case 9:

clrscr();

LOAN \*ln1;

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

int cc;

cin>>cc;

switch(cc)

{

case 1:

clrscr();

ln1=new EDUCATION();

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

char lz[11];

fstream tempx;

gets(lz);

file.open("edu\_loan.pro",ios::binary|ios::in);

tempx.open("tempx.pro",ios::binary|ios::out);

while(file)

{

file.read((char\*)ln1,sizeof(LOAN));

if(strcmp(ln1->getlid(),lz)==0)

ln1->update();

tempx.write((char\*)ln1,sizeof(LOAN));

}

tempx.close();

file.close();

file.open("car\_loan.pro",ios::binary|ios::out);

tempx.open("tempx.pro",ios::binary|ios::in);

while(tempx)

{

tempx.read((char\*)ln1,sizeof(LOAN));

file.write((char\*)ln1,sizeof(LOAN));

}

file.close();

tempx.close();

clrscr();

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

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

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

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

getch();

//delete(ln);

clerk\_screen();

break;

case 2:

clrscr();

ln1=new CAR();

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

char ly[11];

fstream tempy;

gets(ly);

file.open("car\_loan.pro",ios::binary|ios::in);

tempy.open("tempx.pro",ios::binary|ios::out);

while(file)

{

file.read((char\*)ln1,sizeof(LOAN));

if(strcmp(ln1->getlid(),ly)==0)

ln1->update();

tempy.write((char\*)ln1,sizeof(LOAN));

}

tempy.close();

file.close();

file.open("car\_loan.pro",ios::binary|ios::out);

tempy.open("tempx.pro",ios::binary|ios::in);

while(tempy)

{

tempy.read((char\*)ln1,sizeof(LOAN));

file.write((char\*)ln1,sizeof(LOAN));

}

file.close();

tempy.close();

clrscr();

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

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

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

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

getch();

//delete(ln);

clerk\_screen();

break;

default:

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

getch();

clerk\_screen();

break;

}

break;

case 10:

clrscr();

Account \*an;

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

int chc;

cin>>chc;

switch(chc)

{

case 1:

clrscr();

Account \*ac;

ac=new Savings();

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;

ac->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 =========================================";

getch();

clerk\_screen();

break;

case 2:

clrscr();

Account \*ac1;

ac1=new Current();

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;

ac1->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 =========================================";

getch();

clerk\_screen();

break;

default:

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

getch();

clerk\_screen();

break;

}

break;

case 11:

clrscr();

Account \*an1;

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

int ccc;

cin>>ccc;

Account \*ac;

switch(ccc)

{

case 1:

clrscr();

ac=new Savings();

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;

ac->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 =========================================";

getch();

clerk\_screen();

break;

case 2:

clrscr();

ac=new Current();

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;

ac->deduct(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 =========================================";

getch();

clerk\_screen();

break;

default:

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

getch();

clerk\_screen();

break;

}

break;

case 12:

clrscr();

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

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

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

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

getch();

main();

break;

default:

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

getch();

clerk\_screen();

}

}

};

int ka=0;

void welcome();

void welcome\_screen();

void main()

{

clrscr();

/\*

Account \*a,\*b;

a=new Savings;

b=new Savings;

a->create();

fstream y;

y.open("test.txt",ios::binary|ios::app);

y.write((char\*)a,sizeof(Savings));

y.close();

fstream x;

x.open("test.txt",ios::binary|ios::in);

while(x)

{ cout<<"bka";

x.read((char\*)b,sizeof(Savings));

b->details();

getch();

}

x.close();

\*/

if(ka==0)

{

welcome();

ka=1;

}

clrscr();

LOGIN l;

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

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.Employee ||";

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 ch;

cin>>ch;

switch(ch)

{

case 1:

l.cust\_screen();

break;

case 2:

l.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 welcome\_screen()

{

int a,c,v,h,g,count1,b;

count1=380;

for(h=1;h<=10;h++)

{

count1-=20;

for(g=220;g<=count1;g++)

{

setcolor(h);

circle(320,240,g);

setcolor(15);

circle(320,240,219);

setfillstyle(1,h+1);

floodfill(320,240,15);

setcolor(0);

settextstyle(1,0,7);

outtextxy(150,150,"IOB BANK");

}

}

delay(2000);

for(a=0,c=479,b=0,v=639;a<639,c>0,b<479,v>0;a++,c--,b++,v--)

{

setlinestyle(0,0,0);

setcolor(0);

delay(10);

line(a,0,639,c);

line(0,b,v,479);

}

delay(200);

setcolor(GREEN);

settextstyle(7,0,6);

outtextxy(30,100,"A Project By:-");

outtextxy(30,140,"KARAN DUA");

outtextxy(30,180,"NIKET AGARWAL");

outtextxy(30,260,"DEEPTI AGARWAL");

outtextxy(30,220,"RAJAT BHARDWAJ");

settextstyle(0,0,0);

setlinestyle(0,0,0);

getch();

delay(1000);

setcolor(RED);

for(a=0,c=640;a<640,c>=0;a++,c--)

{

delay(2);

line(a,0,c,480);

}

for(a=0,c=480;a<480,c>=0;a++,c--)

{

delay(2);

line(0,a,640,c);

}

delay(500);

closegraph();

}