//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// PROJECT BANKING

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// INCLUDED HEADER FILES

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

#include <iostream.h>

#include <fstream.h>

#include <process.h>

#include <string.h>

#include <stdlib.h>

#include <stdio.h>

#include <ctype.h>

#include <conio.h>

#include <dos.h>

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS CLASS CONTAINS FUNCTIONS RELATED TO DRAW BOX ETC.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class shape

{

public :

void line\_hor(int, int, int, char) ;

void line\_ver(int, int, int, char) ;

void box(int,int,int,int,char) ;

} ;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS CLASS CONTROL ALL THE FUNCTIONS IN THE MENU

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class control

{

public :

void main\_menu(void) ;

void help(void) ;

private :

void edit\_menu(void) ;

} ;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS CLASS CONTAINS FUNCTIONS RELATED TO INITIAL DEPOSIT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class initial

{

public :

void add\_to\_file(int, char t\_name[30], char t\_address[60], float) ;

void display\_list(void) ;

void delete\_account(int) ;

void update\_balance(int, float) ;

void modify(void) ;

int last\_accno(void) ;

int found\_account(int) ;

char \*return\_name(int) ;

char \*return\_address(int) ;

float give\_balance(int) ;

int recordno(int) ;

void display(int) ;

private :

void modify\_account(int, char t\_name[30], char t\_address[60]) ;

void box\_for\_list(void) ;

int accno ;

char name[30], address[60] ;

float balance ;

} ;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS CLASS CONTAINS FUNCTIONS RELATED TO TRANSACTIONS

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class account

{

public :

void new\_account(void) ;

void close\_account(void) ;

void display\_account(void) ;

void transaction(void) ;

void clear(int,int) ;

private :

void add\_to\_file(int, int, int, int, char, char t\_type[10], float, float, float) ;

void delete\_account(int) ;

int no\_of\_days(int, int, int, int, int, int) ;

float calculate\_interest(int, float) ;

void display(int) ;

void box\_for\_display(int) ;

int accno ;

char type[10] ; // Cheque or Cash //

int dd, mm, yy ; // Date //

char tran ; // Deposit or Withdraw //

float interest, amount, balance ;

} ;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// FUNCTION TO DRAW HORIZONTAL LINE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void shape :: line\_hor(int column1, int column2, int row, char c)

{

for ( column1; column1<=column2; column1++ )

{

gotoxy(column1,row) ;

cout <<c ;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// FUNCTION TO DRAW VERTICAL LINE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void shape :: line\_ver(int row1, int row2, int column, char c)

{

for ( row1; row1<=row2; row1++ )

{

gotoxy(column,row1) ;

cout <<c ;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// FUNCTION TO DRAW BOX LINE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void shape :: box(int column1, int row1, int column2, int row2, char c)

{

char ch=218 ;

char c1, c2, c3, c4 ;

char l1=196, l2=179 ;

if (c == ch)

{

c1=218 ;

c2=191 ;

c3=192 ;

c4=217 ;

l1 = 196 ;

l2 = 179 ;

}

else

{

c1=c ;

c2=c ;

c3=c ;

c4=c ;

l1 = c ;

l2 = c ;

}

gotoxy(column1,row1) ;

cout <<c1 ;

gotoxy(column2,row1) ;

cout <<c2 ;

gotoxy(column1,row2) ;

cout <<c3 ;

gotoxy(column2,row2) ;

cout <<c4 ;

column1++ ;

column2-- ;

line\_hor(column1,column2,row1,l1) ;

line\_hor(column1,column2,row2,l1) ;

column1-- ;

column2++ ;

row1++ ;

row2-- ;

line\_ver(row1,row2,column1,l2) ;

line\_ver(row1,row2,column2,l2) ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// FUNCTION TO DISPLAY MAIN MENU AND CALL OTHER FUNCTIONS

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void control :: main\_menu(void)

{

char ch ;

while (1)

{

clrscr() ;

shape s ;

s.box(10,5,71,21,219) ;

s.box(9,4,72,22,218) ;

textcolor(BLACK) ;

textbackground(WHITE) ;

gotoxy(32,7) ;

cout<<”BANKING”;

gotoxy(35,9) ;

cout<<”OPTIONS”;

textcolor(LIGHTGRAY) ;

textbackground(BLACK) ;

gotoxy(30,11) ;

cout <<"1: SEE ACCOUNT" ;

gotoxy(30,12) ;

cout <<"2: LIST OF ACCOUNTS" ;

gotoxy(30,13) ;

cout <<"3: TRANSECTIONS" ;

gotoxy(30,14) ;

cout <<"4: OPEN NEW ACCOUT" ;

gotoxy(30,15) ;

cout <<"5: EDIT ACCOUNTS" ;

gotoxy(30,16) ;

cout <<"6: HELP" ;

gotoxy(30,17) ;

cout <<"0: QUIT" ;

gotoxy(30,19) ;

cout <<"Enter your choice: " ;

ch = getche() ;

if (ch == 27)

break ;

else

if (ch == '1')

{

account a ;

a.display\_account() ;

}

else

if (ch == '2')

{

initial ini ;

ini.display\_list() ;

}

else

if (ch == '3')

{

account a ;

a.transaction() ;

}

else

if (ch == '4')

{

account a ;

a.new\_account() ;

}

else

if (ch == '5')

edit\_menu() ;

else

if (ch == '6')

help() ;

else

if (ch == '0')

break ;

}

for (int i=25; i>=1; i--)

{

delay(20) ;

gotoxy(1,i) ; clreol() ;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// FUNCTION TO DISPLAY EDIT MENU AND CALL OTHER FUNCTIONS

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void control :: edit\_menu(void)

{

char ch ;

while (1)

{

clrscr() ;

shape s ;

s.box(10,5,71,21,219) ;

s.box(9,4,72,22,218) ;

textcolor(BLACK) ;

textbackground(WHITE) ;

gotoxy(34,10) ;

cout<<" EDIT MENU " ;

textcolor(LIGHTGRAY) ;

textbackground(BLACK) ;

gotoxy(31,12) ;

cout <<"1: MODIFY ACCOUNT" ;

gotoxy(31,13) ;

cout <<"2: CLOSE ACCOUNT" ;

gotoxy(31,14) ;

cout <<"0: QUIT" ;

gotoxy(31,16) ;

cout <<"Enter your choice: " ;

ch = getche() ;

if (ch == 27)

break ;

else

if (ch == '1')

{

initial ini ;

ini.modify() ;

break ;

}

else

if (ch == '2')

{

account a ;

a.close\_account() ;

break ;

}

else

if (ch == '0')

break ;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// FUNCTION TO DISPLAY HELP ABOUT PROJECT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void control :: help(void)

{

clrscr() ;

shape s ;

s.box(2,1,79,25,218) ;

s.box(25,2,54,4,219) ;

textcolor(LIGHTGRAY+BLINK) ;

gotoxy(27,3); cprintf("WELCOME TO PROJECT BANKING") ;

textcolor(LIGHTGRAY) ;

delay(10) ;

gotoxy(10,6); cout <<"In this Project you can keep record of daily banking" ;

delay(10) ;

gotoxy(10,7); cout <<"transactions. " ;

delay(10) ;

gotoxy(10,9); cout <<" This program is capable of holding any no. of account." ;

delay(10) ;

gotoxy(10,11); cout <<"- In first option you can see the account of a perticular" ;

delay(10) ;

gotoxy(10,12); cout <<" person by giving simply account no. of that person." ;

delay(10) ;

gotoxy(10,14); cout <<"- In second option you can see the list of all the accounts." ;

delay(10) ;

gotoxy(10,16); cout <<"- Through third option you can do banking transactions" ;

delay(10) ;

gotoxy(10,17); cout <<" (Deposit/Withdraw)." ;

delay(10) ;

gotoxy(10,19); cout <<"- In Fourth option you can open new account." ;

delay(10) ;

gotoxy(10,20); cout <<" (NOTE: Opening amount should not less than Rs.500/-" ;

delay(10) ;

gotoxy(10,22); cout <<"- In Fifth option you can modify or Delete any account." ;

delay(10) ;

gotoxy(10,24); cout <<"- And last option is Quit (Exit to Dos). " ;

delay(10) ;

textcolor(BLACK+BLINK) ; textbackground(WHITE) ;

gotoxy(26,25) ; cout<<" Press any key to continue " ;

textcolor(LIGHTGRAY) ; textbackground(BLACK) ;

gotoxy(25,2) ;

getch() ;

for (int i=25; i>=1; i--)

{

delay(20) ;

gotoxy(1,i) ; clreol() ;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION RETURN LAST ACCOUNT NO. IN THE FILE

// INITIAL.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int initial :: last\_accno(void)

{

fstream file ;

file.open("INITIAL.DAT", ios::in) ;

file.seekg(0,ios::beg) ;

int count=0 ;

while (file.read((char \*) this, sizeof(initial)))

count = accno ;

file.close() ;

return count ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION RETURN RECORD NO. OF THE GIVEN ACCOUNT NO.

// IN THE FILE INITIAL.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int initial :: recordno(int t\_accno)

{

fstream file ;

file.open("INITIAL.DAT", ios::in) ;

file.seekg(0,ios::beg) ;

int count=0 ;

while (file.read((char \*) this, sizeof(initial)))

{

count++ ;

if (t\_accno == accno)

break ;

}

file.close() ;

return count ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DISPLAY THE ACCOUNT FOR GIVEN ACCOUNT NO.

// FROM THE FILE INITIAL.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void initial :: display(int t\_accno)

{

shape s ;

s.box(8,7,73,11,219) ;

fstream file ;

file.open("INITIAL.DAT", ios::in) ;

file.seekg(0,ios::beg) ;

while (file.read((char \*) this, sizeof(initial)))

{

if (t\_accno == accno)

{

gotoxy(8,5) ;

cout <<"ACCOUNT NO. " <<accno ;

gotoxy(10,8) ;

cout <<"Name : " <<name ;

gotoxy(10,9) ;

cout <<"Address : " <<address ;

gotoxy(10,10) ;

cout <<"Balance : " <<balance ;

break ;

}

}

file.close() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION RETURN NAME FOR THE GIVEN ACCOUNT NO.

// IN THE FILE INITIAL.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

char \*initial :: return\_name(int t\_accno)

{

fstream file ;

file.open("INITIAL.DAT", ios::in) ;

file.seekg(0,ios::beg) ;

char t\_name[30] ;

while (file.read((char \*) this, sizeof(initial)))

{

if (accno == t\_accno)

{

strcpy(t\_name,name) ;

break ;

}

}

file.close() ;

return t\_name ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION RETURN ADDRESS FOR THE GIVEN ACCOUNT NO.

// IN THE FILE INITIAL.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

char \*initial :: return\_address(int t\_accno)

{

fstream file ;

file.open("INITIAL.DAT", ios::in) ;

file.seekg(0,ios::beg) ;

char t\_address[60] ;

while (file.read((char \*) this, sizeof(initial)))

{

if (accno == t\_accno)

{

strcpy(t\_address,address) ;

break ;

}

}

file.close() ;

return t\_address ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION RETURN BALANCE FOR THE GIVEN ACCOUNT NO.

// IN THE FILE INITIAL.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

float initial :: give\_balance(int t\_accno)

{

fstream file ;

file.open("INITIAL.DAT", ios::in) ;

file.seekg(0,ios::beg) ;

float t\_balance ;

while (file.read((char \*) this, sizeof(initial)))

{

if (accno == t\_accno)

{

t\_balance = balance ;

break ;

}

}

file.close() ;

return t\_balance ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION RETURN 1 IF THE GIVEN ACCOUNT NO. FOUND

// IN THE FILE INITIAL.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int initial :: found\_account(int t\_accno)

{

fstream file ;

file.open("INITIAL.DAT", ios::in) ;

file.seekg(0,ios::beg) ;

int found=0 ;

while (file.read((char \*) this, sizeof(initial)))

{

if (accno == t\_accno)

{

found = 1 ;

break ;

}

}

file.close() ;

return found ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DRAWS THE BOX FOR THE LIST OF ACCOUNTS

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void initial :: box\_for\_list()

{

shape s ;

s.box(2,1,79,25,218) ;

s.line\_hor(3,78,3,196) ;

s.line\_hor(3,78,5,196) ;

s.line\_hor(3,78,23,196) ;

textbackground(WHITE) ;

gotoxy(3,4) ;

for (int i=1; i<=76; i++) cprintf(" ") ;

textbackground(BLACK) ;

textcolor(BLACK) ; textbackground(WHITE) ;

gotoxy(4,4) ;

cout<<"ACCOUNT NO. NAME OF PERSON BALANCE" ;

textcolor(LIGHTGRAY) ; textbackground(BLACK) ;

int d1, m1, y1 ;

struct date d;

getdate(&d);

d1 = d.da\_day ;

m1 = d.da\_mon ;

y1 = d.da\_year ;

gotoxy(4,2) ;

cout <<"Date: " <<d1 <<"/" <<m1 <<"/" <<y1 ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DISPLAYS THE LIST OF ACCOUNTS IN FILE

// INITIAL.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void initial :: display\_list(void)

{

clrscr() ;

box\_for\_list() ;

int row=6, flag ;

fstream file ;

file.open("INITIAL.DAT", ios::in) ;

while (file.read((char \*) this, sizeof(initial)))

{

flag = 0 ;

delay(10) ;

gotoxy(7,row) ;

cout <<accno ;

gotoxy(25,row) ;

cout <<name ;

gotoxy(57,row) ;

cout <<balance ;

row++ ;

if (row == 23)

{

flag = 1 ;

row = 6 ;

gotoxy(4,24) ;

cout <<"Press any key to continue..." ;

getch() ;

clrscr() ;

box\_for\_list() ;

}

}

file.close() ;

if (!flag)

{

gotoxy(4,24) ;

cout <<"Press any key to continue..." ;

getch() ;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION ADDS THE GIVEN DATA INTO THE FILE

// INITIAL.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void initial :: add\_to\_file(int t\_accno, char t\_name[30], char t\_address[60], float t\_balance)

{

accno = t\_accno ;

strcpy(name,t\_name) ;

strcpy(address,t\_address) ;

balance = t\_balance ;

fstream file ;

file.open("INITIAL.DAT", ios::out | ios::app) ;

file.write((char \*) this, sizeof(initial)) ;

file.close() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DELETES RECORD FOR THE GIVEN ACOUNT NO.

// FROM THE FILE INITIAL.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void initial :: delete\_account(int t\_accno)

{

fstream file ;

file.open("INITIAL.DAT", ios::in) ;

fstream temp ;

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

file.seekg(0,ios::beg) ;

while ( !file.eof() )

{

file.read((char \*) this, sizeof(initial)) ;

if ( file.eof() )

break ;

if ( accno != t\_accno )

temp.write((char \*) this, sizeof(initial)) ;

}

file.close() ;

temp.close() ;

file.open("INITIAL.DAT", ios::out) ;

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

temp.seekg(0,ios::beg) ;

while ( !temp.eof() )

{

temp.read((char \*) this, sizeof(initial)) ;

if ( temp.eof() )

break ;

file.write((char \*) this, sizeof(initial)) ;

}

file.close() ;

temp.close() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION UPDATE BALANCE FOR THE GIVEN ACOUNT NO.

// IN THE FILE INITIAL.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void initial :: update\_balance(int t\_accno, float t\_balance)

{

int recno ;

recno = recordno(t\_accno) ;

fstream file ;

file.open("INITIAL.DAT", ios::out | ios::ate) ;

balance = t\_balance ;

int location ;

location = (recno-1) \* sizeof(initial) ;

file.seekp(location) ;

file.write((char \*) this, sizeof(initial)) ;

file.close() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION MODIFIES THE RECORD FOR THE GIVEN DATA

// IN THE FILE INITIAL.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void initial :: modify\_account(int t\_accno, char t\_name[30], char t\_address[60])

{

int recno ;

recno = recordno(t\_accno) ;

fstream file ;

file.open("INITIAL.DAT", ios::out | ios::ate) ;

strcpy(name,t\_name) ;

strcpy(address,t\_address) ;

int location ;

location = (recno-1) \* sizeof(initial) ;

file.seekp(location) ;

file.write((char \*) this, sizeof(initial)) ;

file.close() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION GIVE THE DATA TO MODIFY THE RECORD IN THE

// FILE INITIAL.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void initial :: modify(void)

{

clrscr() ;

char t\_acc[10] ;

int t, t\_accno ;

gotoxy(71,1) ;

cout <<"<0>=Exit" ;

gotoxy(5,5) ;

cout <<"Enter the account no. " ;

gets(t\_acc) ;

t = atoi(t\_acc) ;

t\_accno = t ;

if (t\_accno == 0)

return ;

clrscr() ;

if (!found\_account(t\_accno))

{

gotoxy(5,5) ;

cout <<"\7Account not found" ;

getch() ;

return ;

}

shape s ;

s.box(2,2,79,24,218) ;

s.line\_hor(3,78,4,196) ;

s.line\_hor(3,78,22,196) ;

gotoxy(71,1) ;

cout <<"<0>=Exit" ;

textbackground(WHITE) ;

gotoxy(3,3) ;

for (int i=1; i<=76; i++) cprintf(" ") ;

textbackground(BLACK) ;

textcolor(BLACK+BLINK) ; textbackground(WHITE) ;

gotoxy(30,3) ;

cout<<"MODIFY ACCOUNT SCREEN" ;

textcolor(LIGHTGRAY) ; textbackground(BLACK) ;

int d1, m1, y1 ;

struct date d;

getdate(&d);

d1 = d.da\_day ;

m1 = d.da\_mon ;

y1 = d.da\_year ;

gotoxy(62,5) ;

cout <<"Date: "<<d1 <<"/" <<m1 <<"/" <<y1 ;

char ch ;

display(t\_accno) ;

account a ;

do

{

a.clear(5,13) ;

gotoxy(5,13) ;

cout <<"Modify this account (y/n): " ;

ch = getche() ;

if (ch == '0')

return ;

ch = toupper(ch) ;

} while (ch != 'N' && ch != 'Y') ;

if (ch == 'N')

return ;

int modified=0, valid ;

char t\_name[30], t\_address[60] ;

gotoxy(5,15) ;

cout <<"Name : " ;

gotoxy(5,16) ;

cout <<"Address : " ;

do

{

a.clear(15,15) ;

a.clear(5,23) ;

gotoxy(5,23) ;

cout <<"ENTER NAME or PRESS <ENTER> FOR NO CHANGE" ;

valid = 1 ;

gotoxy(15,15) ;

gets(t\_name) ;

strupr(t\_name) ;

if (t\_name[0] == '0')

return ;

if (strlen(t\_name) > 25)

{

valid = 0 ;

gotoxy(5,23) ;

cout<<"\7NAME SHOULD NOT GREATER THAN 25" ;

getch() ;

}

} while (!valid) ;

if (strlen(t\_name) > 0)

modified = 1 ;

do

{

a.clear(15,16) ;

a.clear(5,23) ;

gotoxy(5,23) ;

cout <<"ENTER ADDRESS or PRESS <ENTER> FOR NO CHANGE" ;

valid = 1 ;

gotoxy(15,16) ;

gets(t\_address) ;

strupr(t\_address) ;

if (t\_address[0] == '0')

return ;

if (strlen(t\_address) > 55)

{

valid = 0 ;

gotoxy(5,23) ;

cout<<"\7SHOULD NOT BLANK OR GREATER THAN 55" ;

getch() ;

}

} while (!valid) ;

if (strlen(t\_address) > 0)

modified = 1 ;

if (!modified)

return ;

a.clear(5,23) ;

do

{

a.clear(5,18) ;

gotoxy(5,18) ;

cout <<"Do you want to save changes (y/n): " ;

ch = getche() ;

if (ch == '0')

return ;

ch = toupper(ch) ;

} while (ch != 'N' && ch != 'Y') ;

if (ch == 'N')

return ;

modify\_account(t\_accno,t\_name,t\_address) ;

gotoxy(5,21) ;

cout <<"\7Record Modified" ;

gotoxy(5,23) ;

cout <<"Press any key to continue..." ;

getch() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION CLEAR THE GIVEN ROW AND COLUMNS

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void account :: clear(int col, int row)

{

for (int i=col; i<=78; i++)

{

gotoxy(i,row) ;

cout <<" " ;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION ADDS THE GIVEN DATA INTO THE FILE

// BANKING.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void account :: add\_to\_file(int t\_accno, int d1, int m1, int y1, char t\_tran, char t\_type[10], float t\_interest, float t\_amount, float t\_balance)

{

fstream file ;

file.open("BANKING.DAT", ios::app) ;

accno = t\_accno ;

dd = d1 ;

mm = m1 ;

yy = y1 ;

tran = t\_tran ;

strcpy(type,t\_type) ;

interest = t\_interest ;

amount = t\_amount ;

balance = t\_balance ;

file.write((char \*) this, sizeof(account)) ;

file.close() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DELETES THE RECORD FOR GIVIN ACCOUNT NO.

// FROM FILE BANKING.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void account :: delete\_account(int t\_accno)

{

fstream file ;

file.open("BANKING.DAT", ios::in) ;

fstream temp ;

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

file.seekg(0,ios::beg) ;

while ( !file.eof() )

{

file.read((char \*) this, sizeof(account)) ;

if ( file.eof() )

break ;

if ( accno != t\_accno )

temp.write((char \*) this, sizeof(account)) ;

}

file.close() ;

temp.close() ;

file.open("BANKING.DAT", ios::out) ;

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

temp.seekg(0,ios::beg) ;

while ( !temp.eof() )

{

temp.read((char \*) this, sizeof(account)) ;

if ( temp.eof() )

break ;

file.write((char \*) this, sizeof(account)) ;

}

file.close() ;

temp.close() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION ACCEPTS THE DATA TO ADD RECORDS IN THE

// FILE BANKING.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void account :: new\_account(void)

{

char ch ;

int i, valid ;

clrscr() ;

initial ini ;

shape s ;

s.box(2,2,79,24,218) ;

s.line\_hor(3,78,4,196) ;

s.line\_hor(3,78,22,196) ;

gotoxy(71,1) ;

cout <<"<0>=Exit" ;

textbackground(WHITE) ;

gotoxy(3,3) ;

for (i=1; i<=76; i++) cprintf(" ") ;

textbackground(BLACK) ;

textcolor(BLACK+BLINK) ; textbackground(WHITE) ;

gotoxy(32,3) ;

cout"OPEN NEW ACCOUNT" ;

textcolor(LIGHTGRAY) ; textbackground(BLACK) ;

int d1, m1, y1 ;

struct date d;

getdate(&d);

d1 = d.da\_day ;

m1 = d.da\_mon ;

y1 = d.da\_year ;

int t\_accno ;

t\_accno = ini.last\_accno() ;

t\_accno++ ;

if (t\_accno == 1)

{

ini.add\_to\_file(t\_accno,"abc","xyz",1.1) ;

ini.delete\_account(t\_accno) ;

add\_to\_file(t\_accno,1,1,1997,'D',"INITIAL",1.1,1.1,1.1) ;

delete\_account(t\_accno) ;

}

char t\_name[30], t[10], t\_address[60] ;

float t\_bal=0.0, t\_balance=0.0 ;

gotoxy(5,6) ;

cout <<"Date : "<<d1 <<"/" <<m1 <<"/" <<y1 ;

gotoxy(5,8) ;

cout <<"Account no. # " <<t\_accno ;

gotoxy(5,10) ;

cout <<"Name : " ;

gotoxy(5,11) ;

cout <<"Address : " ;

gotoxy(5,12) ;

cout <<"Name of Varifing person : " ;

gotoxy(5,14) ;

cout <<"Initial Deposit : " ;

do

{

clear(15,10) ;

clear(5,23) ;

gotoxy(5,23) ;

cout <<"ENTER NAME OF THE PERSON" ;

valid = 1 ;

gotoxy(15,10) ;

gets(t\_name) ;

strupr(t\_name) ;

if (t\_name[0] == '0')

return ;

if (strlen(t\_name) == 0 || strlen(t\_name) > 25)

{

valid = 0 ;

gotoxy(5,23) ;

cout<<"\7NAME SHOULD NOT BLANK OR GREATER THAN 25" ;

getch() ;

}

} while (!valid) ;

do

{

clear(15,11) ;

clear(5,23) ;

gotoxy(5,23) ;

cout <<"ENTER ADDRESS OF THE PERSON" ;

valid = 1 ;

gotoxy(15,11) ;

gets(t\_address) ;

strupr(t\_address) ;

if (t\_address[0] == '0')

return ;

if (strlen(t\_address) == 0 || strlen(t\_address) > 55)

{

valid = 0 ;

gotoxy(5,23) ;

cout<<"\7SHOULD NOT BLANK OR GREATER THAN 55" ;

getch() ;

}

} while (!valid) ;

do

{

char vari[30] ;

clear(31,12) ;

clear(5,23) ;

gotoxy(5,23) ;

cout <<"ENTER NAME OF THE VARIFING PERSON" ;

valid = 1 ;

gotoxy(31,12) ;

gets(vari) ;

strupr(vari) ;

if (vari[0] == '0')

return ;

if (strlen(vari) == 0 || strlen(vari) > 25)

{

valid = 0 ;

gotoxy(5,23) ;

cout<<"\7SHOULD NOT BLANK OR GREATER THAN 25";

getch() ;

}

} while (!valid) ;

do

{

clear(23,14) ;

clear(5,23) ;

gotoxy(5,23) ;

cout <<"ENTER INITIAL AMOUNT TO BE DEPOSIT" ;

valid = 1 ;

gotoxy(23,14) ;

gets(t) ;

t\_bal = atof(t) ;

t\_balance = t\_bal ;

if (t[0] == '0')

return ;

if (t\_balance < 500)

{

valid = 0 ;

gotoxy(5,23) ;

cout<<"\7SHOULD NOT LESS THAN 500 " ;

getch() ;

}

} while (!valid) ;

clear(5,23) ;

do

{

clear(5,17) ;

valid = 1 ;

gotoxy(5,17) ;

cout <<"Do you want to save the record (y/n) : " ;

ch = getche() ;

if (ch == '0')

return ;

ch = toupper(ch) ;

} while (ch != 'N' && ch != 'Y') ;

if (ch == 'N')

return ;

float t\_amount, t\_interest ;

t\_amount = t\_balance ;

t\_interest = 0.0 ;

char t\_tran, t\_type[10] ;

t\_tran = 'D' ;

strcpy(t\_type,"INITIAL") ;

ini.add\_to\_file(t\_accno,t\_name,t\_address,t\_balance) ;

add\_to\_file(t\_accno,d1,m1,y1,t\_tran,t\_type,t\_interest,t\_amount,t\_balance) ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DRAWS THE BOX FOR DISPLAYING RECORD FROM

// FILE BANKING.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void account :: box\_for\_display(int t\_accno)

{

shape s ;

s.box(2,1,79,25,218) ;

s.line\_hor(3,78,4,196) ;

s.line\_hor(3,78,6,196) ;

s.line\_hor(3,78,23,196) ;

textbackground(WHITE) ;

gotoxy(3,5) ;

for (int i=1; i<=76; i++) cprintf(" ") ;

textbackground(BLACK) ;

textcolor(BLACK) ; textbackground(WHITE) ;

gotoxy(4,5) ;

cout<<"DATE PARTICULAR DEPOSIT WITHDRAW INTEREST BALANCE" ;

textcolor(LIGHTGRAY) ; textbackground(BLACK) ;

int d1, m1, y1 ;

struct date d;

getdate(&d);

d1 = d.da\_day ;

m1 = d.da\_mon ;

y1 = d.da\_year ;

gotoxy(63,2) ;

cout <<"Date: " <<d1 <<"/" <<m1 <<"/" <<y1 ;

gotoxy(4,2) ;

cout <<"Account no. " <<t\_accno ;

initial ini ;

char t\_name[30] ;

strcpy(t\_name,ini.return\_name(t\_accno)) ;

char t\_address[60] ;

strcpy(t\_address,ini.return\_address(t\_accno)) ;

gotoxy(25,2) ;

cout <<t\_name ;

gotoxy(25,3) ;

cout <<t\_address ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION DISPLAY RECORD FROM THE FILE BANKING.DAT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void account :: display\_account(void)

{

clrscr() ;

char t\_acc[10] ;

int t, t\_accno ;

gotoxy(71,1) ;

cout <<"<0>=Exit" ;

gotoxy(5,5) ;

cout <<"Enter the account no. " ;

gets(t\_acc) ;

t = atoi(t\_acc) ;

t\_accno = t ;

if (t\_accno == 0)

return ;

clrscr() ;

initial ini ;

if (!ini.found\_account(t\_accno))

{

gotoxy(5,5) ;

cout <<"\7Account not found" ;

getch() ;

return ;

}

box\_for\_display(t\_accno) ;

int row=7, flag ;

fstream file ;

file.open("BANKING.DAT", ios::in) ;

while (file.read((char \*) this, sizeof(account)))

{

if (accno == t\_accno)

{

flag = 0 ;

delay(10) ;

gotoxy(4,row) ;

cout <<dd <<"/" <<mm <<"/" <<yy ;

gotoxy(16,row) ;

cout <<type ;

if (tran == 'D')

gotoxy(30,row) ;

else

gotoxy(42,row) ;

cout <<amount ;

gotoxy(56,row) ;

cout <<interest ;

gotoxy(66,row) ;

cout <<balance ;

row++ ;

if (row == 23)

{

flag = 1 ;

row = 7 ;

gotoxy(4,24) ;

cout <<"Press any key to continue..." ;

getch() ;

clrscr() ;

box\_for\_display(t\_accno) ;

}

}

}

file.close() ;

if (!flag)

{

gotoxy(4,24) ;

cout <<"Press any key to continue..." ;

getch() ;

}

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION RETURNS THE DIFFERENCE BETWEEN 2 DATES.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

int account :: no\_of\_days(int d1, int m1, int y1, int d2, int m2, int y2)

{

static int month[] = {31,28,31,30,31,30,31,31,30,31,30,31} ;

int days = 0 ;

while (d1 != d2 || m1 != m2 || y1 != y2)

{

days++ ;

d1++ ;

if (d1 > month[m1-1])

{

d1 = 1 ;

m1++ ;

}

if (m1 > 12)

{

m1 = 1 ;

y1++ ;

}

}

return days ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION CALCULATES INTEREST.

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

float account :: calculate\_interest(int t\_accno, float t\_balance)

{

fstream file ;

file.open("BANKING.DAT", ios::in) ;

file.seekg(0,ios::beg) ;

int d1, m1, y1, days ;

while (file.read((char \*) this, sizeof(account)))

{

if (accno == t\_accno)

{

d1 = dd ;

m1 = mm ;

y1 = yy ;

break ;

}

}

int d2, m2, y2 ;

struct date d;

getdate(&d);

d2 = d.da\_day ;

m2 = d.da\_mon ;

y2 = d.da\_year ;

float t\_interest=0.0 ;

if ((y2<y1) || (y2==y1 && m2<m1) || (y2==y1 && m2==m1 && d2<d1))

return t\_interest ;

days = no\_of\_days(d1,m1,y1,d2,m2,y2) ;

int months=0 ;

if (days >= 30)

{

months = days/30 ;

t\_interest = ((t\_balance\*2)/100) \* months ;

}

file.close() ;

return t\_interest ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION MAKES TRANSACTIONS (DEPOSIT/WITHDRAW).

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void account :: transaction(void)

{

clrscr() ;

char t\_acc[10] ;

int t, t\_accno, valid ;

gotoxy(71,1) ;

cout <<"<0>=Exit" ;

gotoxy(5,5) ;

cout <<"Enter the account no. " ;

gets(t\_acc) ;

t = atoi(t\_acc) ;

t\_accno = t ;

if (t\_accno == 0)

return ;

clrscr() ;

initial ini ;

if (!ini.found\_account(t\_accno))

{

gotoxy(5,5) ;

cout <<"\7Account not found" ;

getch() ;

return ;

}

shape s ;

s.box(2,2,79,24,218) ;

s.line\_hor(3,78,4,196) ;

s.line\_hor(3,78,22,196) ;

gotoxy(71,1) ;

cout <<"<0>=Exit" ;

textbackground(WHITE) ;

gotoxy(3,3) ;

for (int i=1; i<=76; i++) cprintf(" ") ;

textbackground(BLACK) ;

textcolor(BLACK+BLINK) ; textbackground(WHITE) ;

gotoxy(29,3) ;

cout<<"TRANSACTION IN ACCOUNT";

textcolor(LIGHTGRAY) ; textbackground(BLACK) ;

int d1, m1, y1 ;

struct date d;

getdate(&d);

d1 = d.da\_day ;

m1 = d.da\_mon ;

y1 = d.da\_year ;

gotoxy(5,6) ;

cout <<"Date : "<<d1 <<"/" <<m1 <<"/" <<y1 ;

gotoxy(5,8) ;

cout <<"Account no. # " <<t\_accno ;

char t\_name[30] ;

char t\_address[60] ;

float t\_balance ;

strcpy(t\_name,ini.return\_name(t\_accno)) ;

strcpy(t\_address,ini.return\_address(t\_accno)) ;

t\_balance = ini.give\_balance(t\_accno) ;

s.box(25,10,75,13,218) ;

gotoxy(27,11) ;

cout <<"Name : " <<t\_name ;

gotoxy(27,12) ;

cout <<"Address: " <<t\_address ;

gotoxy(5,15) ;

cout <<"Last Balance : Rs." <<t\_balance ;

char t\_tran, t\_type[10], tm[10] ;

float t\_amount, t\_amt ;

do

{

clear(5,18) ;

valid = 1 ;

gotoxy(5,18) ;

cout <<"Deposit or Withdraw (D/W) : " ;

t\_tran = getche() ;

if (t\_tran == '0')

return ;

t\_tran = toupper(t\_tran) ;

} while (t\_tran != 'D' && t\_tran != 'W') ;

do

{

clear(5,19) ;

clear(5,23) ;

gotoxy(5,23) ;

cout <<"ENTER TRANSACTION BY CASH OR CHEQUE" ;

valid = 1 ;

gotoxy(5,19) ;

cout <<"(Cash/Cheque) : " ;

gets(t\_type) ;

strupr(t\_type) ;

if (t\_type[0] == '0')

return ;

if (strcmp(t\_type,"CASH") && strcmp(t\_type,"CHEQUE"))

{

valid = 0 ;

gotoxy(5,23) ;

cout<<"\7ENTER CORRECTLY " ;

getch() ;

}

} while (!valid) ;

do

{

clear(5,21) ;

clear(5,23) ;

gotoxy(5,23) ;

cout <<"ENTER AMOUNT FOR TRANSACTION" ;

valid = 1 ;

gotoxy(5,21) ;

cout <<"Amount : Rs." ;

gets(tm) ;

t\_amt = atof(tm) ;

t\_amount = t\_amt ;

if (tm[0] == '0')

return ;

if ((t\_tran == 'W' && t\_amount > t\_balance) || (t\_amount < 1))

{

valid = 0 ;

gotoxy(5,23) ;

cout<<"\7INVALID DATA ENTERED " ;

getch() ;

}

} while (!valid) ;

char ch ;

clear(5,23) ;

do

{

clear(40,20) ;

valid = 1 ;

gotoxy(40,20) ;

cout <<"Save transaction (y/n): " ;

ch = getche() ;

if (ch == '0')

return ;

ch = toupper(ch) ;

} while (ch != 'N' && ch != 'Y') ;

if (ch == 'N')

return ;

float t\_interest ;

t\_interest = calculate\_interest(t\_accno,t\_balance) ;

if (t\_tran == 'D')

t\_balance = t\_balance + t\_amount + t\_interest ;

else

t\_balance = (t\_balance - t\_amount) + t\_interest ;

ini.update\_balance(t\_accno,t\_balance) ;

add\_to\_file(t\_accno,d1,m1,y1,t\_tran,t\_type,t\_interest,t\_amount,t\_balance) ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS FUNCTION CLOSE THE ACCOUNT (DELETE ACCOUNT).

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void account :: close\_account(void)

{

clrscr() ;

char t\_acc[10] ;

int t, t\_accno ;

gotoxy(71,1) ;

cout <<"<0>=Exit" ;

gotoxy(5,5) ;

cout <<"Enter the account no. " ;

gets(t\_acc) ;

t = atoi(t\_acc) ;

t\_accno = t ;

if (t\_accno == 0)

return ;

clrscr() ;

initial ini ;

if (!ini.found\_account(t\_accno))

{

gotoxy(5,5) ;

cout <<"\7Account not found" ;

getch() ;

return ;

}

shape s ;

s.box(2,2,79,24,218) ;

s.line\_hor(3,78,4,196) ;

s.line\_hor(3,78,22,196) ;

gotoxy(71,1) ;

cout <<"<0>=Exit" ;

textbackground(WHITE) ;

gotoxy(3,3) ;

for (int i=1; i<=76; i++) cout<<" " ;

textbackground(BLACK) ;

textcolor(BLACK+BLINK) ; textbackground(WHITE) ;

gotoxy(30,3) ;

cout<<"CLOSE ACCOUNT SCREEN" ;

textcolor(LIGHTGRAY) ; textbackground(BLACK) ;

int d1, m1, y1 ;

struct date d;

getdate(&d);

d1 = d.da\_day ;

m1 = d.da\_mon ;

y1 = d.da\_year ;

gotoxy(62,5) ;

cout <<"Date: "<<d1 <<"/" <<m1 <<"/" <<y1 ;

char ch ;

ini.display(t\_accno) ;

do

{

clear(5,15) ;

gotoxy(5,15) ;

cout <<"Close this account (y/n): " ;

ch = getche() ;

if (ch == '0')

return ;

ch = toupper(ch) ;

} while (ch != 'N' && ch != 'Y') ;

if (ch == 'N')

return ;

ini.delete\_account(t\_accno) ;

delete\_account(t\_accno) ;

gotoxy(5,20) ;

cout <<"\7Record Deleted" ;

gotoxy(5,23) ;

cout <<"Press any key to continue..." ;

getch() ;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// THIS IS MAIN FUNCTION CALLING HELP AND MAIN MENU FUNCTIONS

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void main(void)

{

control c ;

c.help() ;

c.main\_menu() ;

}