**Certificate**

Certified that this Projects is the bonafide work of **Parmjeet Singh Kainth** of class **12th** of **Ganga International School, Dhabi Gujran(Patiala).**

He performed this project during the academic year 2016-17, as part of practical examination of **Computer Science** to be submitted for:-

‘**ALL INDIA SENIOR SECONDARY CERTIFICATE EXAMINATION-2016-17(A.I.S.S.C.E,-2016-17)**‘

* The student’s initiative, Cooperation during the performance of the project was Excellent/Good/Average/Below Average.
* His/her aesthetic presentation, visual appeal, expression and neatness is Excellent/Good/Average/Below Average.
* His/her content accuracy, originality and analysis of different perception is excellent/Good/Average/Below Average.

\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

EXAMINER’S SIGN Principal’s sign Teacher’s sign

(With school seal)

In the accomplishment of this project successfully, many people have best owned upon we their blessings and the heart pledged support, this time I am utilizing to thanks all the people who have been concerned with project.

Primarily I would thank god for being able to cpmlete this project with success. Then I would like to thank my principal MR………………and Computer teacher MR.GURDEV SINGH whose valuable guidance has been the ones that helped me patch this project and make it full proof success his suggestions and his instructions has served as the major contributor towards the completion of the project.

Then I would like to thank my parents and friends who have helped me with their valuable suggestions and guidance has been helpful in various phases of the completion of the project.

Last but not the Least I would like to thank my classmates who have helped me a lot.

This is a project based on **LIBRARY MANAGEMENT.** The program helps us to enter, display or alter the details of different Books and Members.It includes various function programs to do the above mentioned tasks. Data file handling has been effectively used in the program.

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

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* PROJECT BOOK LIBRARY\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*PASSWORD IS **“LIBRARY”** or **“library”**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* 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 CONTROL ALL THE FUNCTIONS IN THE MENU

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

class MENU

{

public :

void main\_menu(void) ;

void introduction(void) ;

void about(void);

private :

void edit\_menu(void) ;

void edit\_book(void) ;

void edit\_member(void) ;

} ;

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

THIS CLASS CONTAINS FUNCTIONS RELATED TO BOOKS

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

class BOOK

{

public :

void list(void) ;

char \*bookname(int) ;

protected :

void add\_new\_book(int, char tname[33], char tauthor[26], float, int,

int) ;

void update\_copies(int, int, int) ;

void modify(int, char[], char[], float) ;

void deletion(void) ;

int book\_found(int) ;

int bookname\_found(char []) ;

int recordno(int) ;

int available(int) ;

char \*authorname(int) ;

float bookprice(int) ;

int no\_of\_copies(int) ;

int bookcodeof(char[]) ;

void display(int) ;

int reccount(void) ;

void delete\_rec(int) ;

private :

int bookcode, copies ;

char name[33], author[26] ;

float price ;

int avail ;

};

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

THIS CLASS CONTAINS FUNCTIONS RELATED TO MEMBERS

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

class MEMBER

{

public :

void list(void) ;

protected :

void add\_mem(int, int, char [], char [], char[], int, int, int) ;

void modify(int, char[], char[], char[]) ;

void deletion(void) ;

int member\_found(int) ;

void update\_book(int, int, int, int, int) ;

char \*membername(int) ;

char \*memberphone(int) ;

char \*memberaddress(int) ;

int recordno(int) ;

int lastcode(void) ;

int issued(int) ;

int fine(int) ;

void display(int) ;

void delete\_rec(int) ;

private :

int memcode, bookcode ;

char name[26], phone[10], address[33] ;

int dd, mm, yy ; // DATE OF RETURNING THE BOOK //

} ;

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

THIS IS DERIVED FROM CLASS BOOK & MEMBER AND CONTAINS FUNCTIONS FOR WORKING (ISSUE,RETURN,ETC).

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

class WORKING : public BOOK, public MEMBER

{

public :

void issuebook(void) ;

void returnbook(void) ;

void add\_book(void) ;

void add\_member(void) ;

void modify\_book(void) ;

void modify\_member(void) ;

void delete\_book(void) ;

void delete\_member(void) ;

} ;

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

THIS CLASS CONTAINS FUNCTIONS RELATED DATE & FINE

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

class DATE

{

public :

void extend\_date(int,int,int,int) ;

int diff(int, int, int, int, int, int) ;

int day, mon, year ;

} ;

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

FUNCTION TO EXTEND GIVEN DATE BY 15 DAYS

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

void DATE :: extend\_date(int d1, int m1, int y1, int days)

{

static int month[] = {31,29,31,30,31,30,31,31,30,31,30,31} ;

for (int i=1; i<=days; i++)

{

d1++ ;

if ((d1 > month[m1-1]) || (y1%4 != 0 && m1 == 2 && d1 > 28))

{

d1 = 1 ;

m1++ ;

}

if (m1 > 12)

{

m1 = 1 ;

y1++ ;

}

}

day = d1 ;

mon = m1 ;

year = y1 ;

}

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

THIS FUNCTION RETURN THE DIFFERENCE BETWEEN TWO GIVEN DATES

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

int DATE :: diff(int d1, int m1, int y1, int d2, int m2, int y2)

{

int days = 0 ;

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

return days ;

static int month[] = {31,29,31,30,31,30,31,31,30,31,30,31} ;

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

{

days++ ;

d1++ ;

if ((d1 > month[m1-1]) || (y1%4 != 0 && m1 == 2 && d1 > 28))

{

d1 = 1 ;

m1++ ;

}

if (m1 > 12)

{

m1 = 1 ;

y1++ ;

}

}

return days ;

}

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

FUNCTION TO DISPLAY MAIN MENU & CONTROL ALL THE FUNCTION IN THE MAIN MENU.

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

void MENU :: main\_menu(void)

{

char ch ;

while (1)

{

clrscr();

textbackground(0);

int chno=1; char chkcho='1';

\_setcursortype(\_NOCURSOR);

char a; clrscr();

for (int i=15;i<60;i++) //top horizontal

{

gotoxy(i, 3);

if (i%2==0)

{

textcolor(3);cprintf("\*");

}

else

{

textcolor(4);cprintf("\*");

}

}

for (int y=4;y<20;y++) //left vertical

{

gotoxy(15,y);

if (y%2==0)

{

textcolor(3);cprintf("\*");

}

else

{

textcolor(4);cprintf("\*");

}

}

for (int t=4;t<20;t++) //right vertical

{

gotoxy(59,t);

if (t%2==0)

{

textcolor(3);cprintf("\*");

}

else

{

textcolor(4);cprintf("\*");

}

}

for (int g=15;g<60;g++)

{

gotoxy(g,19);

if (g%2==0)

{textcolor(3);cprintf("\*");}

else

{textcolor(4);cprintf("\*");}

}

for (int u=15;u<60;u++) //middle horizontal

{

gotoxy(u,7);

if (u%2==0)

{textcolor(3);cprintf("\*");}

else

{textcolor(4);cprintf("\*");}

}

gotoxy(29,5) ;

cout <<"B O O K L I B R A R Y" ;

gotoxy(30,8) ;

cout <<"1. INTRODUCTION" ;

gotoxy(30,9) ;

cout <<"2. ADD NEW BOOK" ;

gotoxy(30,10) ;

cout <<"3. ADD NEW MEMBER" ;

gotoxy(30,11) ;

cout <<"4. ISSUE BOOK" ;

gotoxy(30,12) ;

cout <<"5. RETURN BOOK" ;

gotoxy(30,13) ;

cout <<"6. LIST OF BOOKS" ;

gotoxy(30,14) ;

cout <<"7. LIST OF MEMBERS" ;

gotoxy(30,15) ;

cout <<"8. EDIT" ;

gotoxy(30,16);

cout<<"9. About";

gotoxy(30,17) ;

cout <<"0. QUIT" ;

gotoxy(30,18) ;

cout <<"Enter your choice : " ;

ch = getche() ;

if (ch == 27)

break ;

else if (ch == '1')

introduction() ;

else if (ch == '2')

{

WORKING W ;

W.add\_book() ;

}

else if (ch == '3')

{

WORKING W ;

W.add\_member() ;

}

else if (ch == '4')

{

WORKING W ;

W.issuebook() ;

}

else if (ch == '5')

{

WORKING W ;

W.returnbook() ;

}

else if (ch == '6')

{

BOOK B ;

B.list() ;

}

else if (ch == '7')

{

MEMBER M ;

M.list() ;

}

else if (ch == '8')

edit\_menu() ;

else if (ch == '9')

about();

else if (ch == '0')

break ;

}

}

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

FUNCTION TO DISPLAY EDIT MENU

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

void MENU :: edit\_menu(void)

{

clrscr();

char ch ;

while (1)

{

textbackground(2);

clrscr() ;

gotoxy(32,9) ;

cout <<"E D I T M E N U";

gotoxy(32,10) ;

cout <<"~~~~~~~~~~~~~~~~" ;

gotoxy(34,13) ;

cout <<"1. BOOKS" ;

gotoxy(34,14) ;

cout <<"2. MEMBERS" ;

gotoxy(34,15) ;

cout <<"0. EXIT" ;

gotoxy(31,17) ;

cout <<"Enter your choice : " ;

ch = getche() ;

if (ch == 27)

break ;

else if (ch =='1')

edit\_book() ;

else if (ch == '2')

edit\_member() ;

else if (ch == '0')

break ;

}

}

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

FUNCTION TO DISPLAY EDIT MENU FOR BOOK & CONTROL ALL THE FUNCTION IN THE EDIT MENU.

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

void MENU :: edit\_book(void)

{

clrscr();

char ch ;

while (1)

{

textbackground(5);

clrscr() ;

gotoxy(31,9) ;

cout <<"E D I T B O O K S";

gotoxy(31,10) ;

cout <<"~~~~~~~~~~~~~~~~~~" ;

gotoxy(34,13) ;

cout <<"1. MODIFY" ;

gotoxy(34,14) ;

cout <<"2. DELETE" ;

gotoxy(34,15) ;

cout <<"0. EXIT" ;

gotoxy(31,17) ;

cout <<"Enter your choice : " ;

ch = getche() ;

if (ch == 27)

break ;

else if (ch == '1')

{

WORKING W ;

W.modify\_book() ;

}

else if (ch == '2')

{

WORKING W ;

W.delete\_book() ;

}

else if (ch == '0')

break ;

}

}

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

FUNCTION TO DISPLAY EDIT MENU FOR MEMBERS & CONTROL ALL THE FUNCTION IN THE EDIT MENU.

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

void MENU :: edit\_member(void)

{

clrscr();

char ch ;

while (1)

{

textbackground(5);

clrscr() ;

gotoxy(29,9) ;

cout <<"E D I T M E M B E R S";

gotoxy(29,10) ;

cout <<"~~~~~~~~~~~~~~~~~~~~~~" ;

gotoxy(34,13) ;

cout <<"1. MODIFY" ;

gotoxy(34,14) ;

cout <<"2. DELETE" ;

gotoxy(34,15) ;

cout <<"0. EXIT" ;

gotoxy(29,17) ;

cout <<"Enter your choice : " ;

ch = getche() ;

if (ch == 27)

break ;

else if (ch == '1')

{

WORKING W ;

W.modify\_member() ;

}

else if (ch == '2')

{

WORKING W ;

W.delete\_member() ;

}

else if (ch == '0')

break ;

}

}

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

FUNCTION TO DISPLAY THE INTRODUCTION FOR THE PROJECT.

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

void MENU ::introduction(void)

{

clrscr();

textbackground(1);

clrscr();

gotoxy(31,5) ;

cout <<"Welcome to Project" ;

gotoxy(33,7) ;

cout<<" BOOK LIBRARY ";

gotoxy(15,10) ;

cout <<"This project has facility of maintaining records" ;

gotoxy(15,11) ;

cout <<"of BOOKS and MEMBERS." ;

gotoxy(15,13) ;

cout <<"This project can hold more than 10,000 books" ;

gotoxy(15,14) ;

cout <<"records." ;

gotoxy(15,16) ;

cout <<"One member can issue one book at a time. If he/she" ;

gotoxy(15,17) ;

cout <<"does not return book upto 3 days he/she have to" ;

gotoxy(15,18) ;

cout <<"pay fine of Rs.2/- per day." ;

gotoxy(27,24) ;

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

getch() ;

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*FUNCTION TO DISPLAY CONTACT OF STUDENT WHO CREATE THIS PROJECT.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\

void MENU::about()

{

clrscr() ;

\_setcursortype(\_NOCURSOR);

textbackground(2);

for (int i=1;i<27;i++)

{

clrscr();

gotoxy(i,1);

cprintf("Library Management System");

delay(100);

}

for (int o=2;o<5;o++)

{

clrscr();

gotoxy(26,o);cout<<"Library Management System";delay(100);

}

delay(1000);

gotoxy(33,7);

cout<<"Made By:-";delay (1000);

gotoxy(28,9);

cout<<"Parmjeet Singh Kainth";delay(1000);

gotoxy(24,10);

cout<<"Class XII-Science(Non-Medical)";delay(1000);

gotoxy(22,11);

cout<<"Address--> House No. 6,Ward No. 8,";delay(1000);

gotoxy(31,12);

cout<<" Main Road Near State Bank of patiala,";delay(1000);

gotoxy(31,13);

cout<<" Khanouri Mandi";delay(1000);

gotoxy(22,14);

cout<<"Contact No.-->”; cout<<”8556913070,905661370,9115226163";delay(1000);

gotoxy(22,25);

cout<<"Press any key to continue.....";delay(1000);

getch() ;

}

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

THIS FUNCTION RETURN 0 IF GIVEN BOOK CODE NOT FOUND

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

int BOOK :: book\_found(int tcode)

{

fstream file ;

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

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

int found=0 ;

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

{

if (bookcode == tcode)

{

found = 1 ;

break ;

}

}

file.close() ;

return found ;

}

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

THIS FUNCTION RETURN 0 IF GIVEN BOOK NAME NOT FOUND

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

int BOOK :: bookname\_found(char t1code[33])

{

fstream file ;

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

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

int found=0 ;

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

{

if (!strcmpi(name,t1code))

{

found = 1 ;

break ;

}

}

file.close() ;

return found ;

}

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

THIS FUNCTION RETURN RECORD NO. FOR THE BOOK CODE

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

int BOOK :: recordno(int tcode)

{

fstream file ;

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

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

int count=0 ;

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

{

count++ ;

if (bookcode == tcode)

break ;

}

file.close() ;

return count ;

}

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

THIS FUNCTION RETURNS THE AVAILABLE COPIES FOR THE GIVEN BOOK CODE.

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

int BOOK :: available(int tcode)

{

fstream file ;

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

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

int tavail=0 ;

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

{

if (bookcode == tcode)

{

tavail = avail ;

break ;

}

}

file.close() ;

return tavail ;

}

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

THIS FUNCTION RETURNS THE NO. OF COPIES FOR THE GIVEN BOOK CODE.

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

int BOOK :: no\_of\_copies(int tcode)

{

fstream file ;

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

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

int tcopies=0 ;

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

{

if (bookcode == tcode)

{

tcopies = copies ;

break ;

}

}

file.close() ;

return tcopies ;

}

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

THIS FUNCTION RETURNS THE BOOK NAME OF THE GIVEN BOOK CODE.

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

char \*BOOK :: bookname(int tcode)

{

fstream file ;

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

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

char tname[33] ;

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

{

if (bookcode == tcode)

{

strcpy(tname,name) ;

break ;

}

}

file.close() ;

return tname ;

}

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

THIS FUNCTION RETURNS THE AUTHOR NAME OF THE GIVEN BOOK CODE.

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

char \*BOOK :: authorname(int tcode)

{

fstream file ;

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

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

char tauthor[26] ;

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

{

if (bookcode == tcode)

{

strcpy(tauthor,author) ;

break ;

}

}

file.close() ;

return tauthor ;

}

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

THIS FUNCTION RETURNS THE BOOK PRICE OF THE GIVEN BOOK CODE.

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

float BOOK :: bookprice(int tcode)

{

fstream file ;

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

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

float tprice=0.0 ;

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

{

if (bookcode == tcode)

{

tprice = price ;

break ;

}

}

file.close() ;

return tprice ;

}

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

THIS FUNCTION RETURNS THE BOOK CODE OF THE GIVEN BOOK NAME.

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

int BOOK :: bookcodeof(char t1code[33])

{

fstream file ;

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

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

int tcode=0 ;

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

{

if (!strcmpi(name,t1code))

{

tcode = bookcode ;

break ;

}

}

file.close() ;

return tcode ;

}

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

THIS FUNCTION RETURNS THE NO. OF THE RECORDS IN THE BOOK FILE.

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

int BOOK :: reccount(void)

{

fstream file ;

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

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

int count=0 ;

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

count++ ;

file.close() ;

return count ;

}

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

THIS FUNCTION DELETES THE RECORD OF THE GIVEN BOOK CODE.

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

void BOOK :: delete\_rec(int tcode)

{

fstream file ;

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

fstream temp ;

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

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

while ( !file.eof() )

{

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

if ( file.eof() )

break ;

if ( bookcode != tcode )

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

}

file.close() ;

temp.close() ;

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

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

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

while ( !temp.eof() )

{

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

if ( temp.eof() )

break ;

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

}

file.close() ;

temp.close() ;

}

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

THIS FUNCTION ADD THE RECORD IN THE BOOK FILE

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

void BOOK :: add\_new\_book(int tcode,char tname[33], char tauthor[26], float tprice, int tcopies, int tavail)

{

fstream file ;

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

bookcode = tcode ;

strcpy(name,tname) ;

strcpy(author,tauthor) ;

price = tprice ;

copies = tcopies ;

avail = tavail ;

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

file.close() ;

}

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

THIS FUNCTION UPDATE THE RECORD IN THE BOOK FILE FOR THE GIVEN BOOK CODE

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

void BOOK :: update\_copies(int tcode, int tcopies, int tavail)

{

int recno ;

recno = recordno(tcode) ;

fstream file ;

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

copies = tcopies ;

avail = tavail ;

int location ;

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

file.seekp(location) ;

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

file.close() ;

}

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

THIS FUNCTION MODIFY THE RECORD IN THE BOOK FILE FOR THE GIVEN BOOK CODE

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

void BOOK :: modify(int tcode, char tname[33], char tauthor[26], float tprice)

{

int recno ;

recno = recordno(tcode) ;

fstream file ;

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

strcpy(name,tname) ;

strcpy(author,tauthor) ;

price = tprice ;

int location ;

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

file.seekp(location) ;

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

file.close() ;

}

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

THIS FUNCTION DISPLAY THE RECORD FROM THE BOOK FILE FOR THE GIVEN BOOK CODE

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

void BOOK :: display(int tcode)

{

clrscr();

fstream file ;

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

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

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

{

if (bookcode == tcode)

{

gotoxy(5,5) ;

cout <<"Book Code : " <<bookcode ;

gotoxy(5,7) ;

cout <<"Book Name : " <<name ;

gotoxy(5,8) ;

cout <<"Author Name : " <<author ;

gotoxy(5,9) ;

cout <<"Price : Rs." <<price ;

gotoxy(5,10) ;

cout <<"Copies : " <<price ;

gotoxy(5,11) ;

cout <<"Available : " <<avail ;

break ;

}

}

file.close() ;

}

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

THIS FUNCTION DISPLAY THE LIST OF BOOKS.

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

void BOOK :: list(void)

{

clrscr();

textbackground(5);

clrscr() ;

int row = 6 , found=0, flag=0 ;

char ch ;

gotoxy(33,2) ;

cout <<"LIST OF BOOKS" ;

gotoxy(32,3) ;

cout <<"~~~~~~~~~~~~~~~" ;

gotoxy(1,4) ;

cout <<"CODE BOOK NAME AUTHOR PRICE”;

cout<<”COPIES" ;

gotoxy(1,5) ;

cout<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~”; cout<<”~~~~~~~~~~~~~~~~~~~~~~~~~~~" ;

fstream file ;

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

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

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

{

flag = 0 ;

delay(20) ;

found = 1 ;

gotoxy(2,row) ;

cout <<bookcode ;

gotoxy(7,row) ;

cout <<name ;

gotoxy(40,row) ;

cout <<author ;

gotoxy(66,row) ;

cout <<price ;

gotoxy(73,row) ;

cout <<copies ;

textbackground(WHITE) ; textcolor(BLACK) ;

gotoxy(40,row+1) ;

cprintf("STATUS: ") ;

textcolor(BLACK+BLINK) ;

cprintf("%d copies available",avail) ;

textbackground(BLACK) ; textcolor(LIGHTGRAY) ;

if ( row == 22 )

{

flag = 1 ;

row = 6 ;

gotoxy(1,25) ;

cout <<"Press any key to continue or Press <ESC> to exit" ;

ch = getch() ;

if (ch == 27)

break ;

clrscr() ;

gotoxy(33,2) ;

cout <<"LIST OF BOOKS" ;

gotoxy(32,3) ;

cout <<"~~~~~~~~~~~~~~~" ;

gotoxy(1,4) ;

cout <<"CODE BOOK NAME AUTHOR “;

cout<<”PRICE COPIES" ;

gotoxy(1,5) ;

cout<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~”;cout<<”~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" ;

}

else

row = row + 2 ;

}

if (!found)

{

gotoxy(5,10) ;

cout <<"\7Records not found" ;

}

if (!flag)

{

gotoxy(1,25) ;

cout <<"Press any key to continue" ;

getche() ;

}

file.close () ;

}

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

THIS FUNCTION RETURN 0 IF THE GIVEN MEMBER CODE NOT FOUND

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

int MEMBER :: member\_found(int mcode)

{

fstream file ;

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

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

int found=0 ;

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

{

if (memcode == mcode)

{

found = 1 ;

break ;

}

}

file.close() ;

return found ;

}

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

THIS FUNCTION RETURN 0 IF THE MEMBER HAVE NOT ISSUED ANY BOOK.

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

int MEMBER :: issued(int mcode)

{

fstream file ;

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

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

int missue=0 ;

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

{

if (memcode == mcode)

{

missue = bookcode ;

break ;

}

}

file.close() ;

return missue ;

}

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

THIS FUNCTION CALCULATE AND RETURN FINE FOR THE GIVEN MEMBER CODE.

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

int MEMBER :: fine(int mcode)

{

DATE D ;

int d1, m1, y1 ;

struct date d;

getdate(&d);

d1 = d.da\_day ;

m1 = d.da\_mon ;

y1 = d.da\_year ;

fstream file ;

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

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

int days, t\_fine ;

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

{

if (memcode == mcode)

{

days = D.diff(dd,mm,yy,d1,m1,y1) ;

t\_fine = days \* 2 ;

break ;

}

}

file.close() ;

return t\_fine ;

}

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

THIS FUNCTION RETURN THE LAST CODE OF THE MEMBER FILE.

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

int MEMBER :: lastcode(void)

{

fstream file ;

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

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

int mcode=0 ;

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

mcode = memcode ;

file.close() ;

return mcode ;

}

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

THIS FUNCTION RETURNS MEMBER NAME OF THE GIVEN MEMBER CODE.

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

char \*MEMBER :: membername(int mcode)

{

fstream file ;

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

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

char mname[26] ;

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

{

if (memcode == mcode)

{

strcpy(mname,name) ;

break ;

}

}

file.close() ;

return mname ;

}

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

THIS FUNCTION RETURNS MEMBER PHONE OF THE GIVEN MEMBER CODE.

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

char \*MEMBER :: memberphone(int mcode)

{

fstream file ;

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

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

char mphone[10] ;

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

{

if (memcode == mcode)

{

strcpy(mphone,phone) ;

break ;

}

}

file.close() ;

return mphone ;

}

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

THIS FUNCTION RETURNS MEMBER ADDRESS OF THE GIVEN MEMBER CODE.

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

char \*MEMBER :: memberaddress(int mcode)

{

fstream file ;

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

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

char maddress[33] ;

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

{

if (memcode == mcode)

{

strcpy(maddress,address) ;

break ;

}

}

file.close() ;

return maddress ;

}

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

THIS FUNCTION RETURNS RECORD NO. OF THE GIVEN MEMBER CODE.

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

int MEMBER :: recordno(int mcode)

{

fstream file ;

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

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

int count=0 ;

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

{

count++ ;

if (memcode == mcode)

break ;

}

file.close() ;

return count ;

}

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

THIS FUNCTION DELETE RECORD FOR THE GIVEN MEMBER CODE.

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

void MEMBER :: delete\_rec(int mcode)

{

fstream file ;

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

fstream temp ;

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

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

while ( !file.eof() )

{

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

if ( file.eof() )

break ;

if ( memcode != mcode )

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

}

file.close() ;

temp.close() ;

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

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

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

while ( !temp.eof() )

{

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

if ( temp.eof() )

break ;

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

}

file.close() ;

temp.close() ;

}

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

THIS FUNCTION UPDATE RECORD FOR THE GIVEN MEMBER CODE.

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

void MEMBER :: update\_book(int mcode, int tcode, int d1, int m1, int y1)

{

fstream file ;

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

fstream temp ;

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

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

while ( !file.eof() )

{

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

if ( file.eof() )

break ;

if ( memcode == mcode )

{

bookcode = tcode ;

dd = d1 ;

mm = m1 ;

yy = y1 ;

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

}

else

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

}

file.close() ;

temp.close() ;

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

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

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

while ( !temp.eof() )

{

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

if ( temp.eof() )

break ;

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

}

file.close() ;

temp.close() ;

}

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

THIS FUNCTION MODIFY RECORD FOR THE GIVEN MEMBER CODE.

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

void MEMBER :: modify(int mcode, char mname[26], char mphone[10], char maddress[33])

{

int recno ;

recno = recordno(mcode) ;

fstream file ;

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

strcpy(name,mname) ;

strcpy(phone,mphone) ;

strcpy(address,maddress) ;

int location ;

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

file.seekp(location) ;

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

file.close() ;

}

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

THIS FUNCTION ADD RECORD IN THE FILE FOR THE GIVEN MEMBER CODE.

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

void MEMBER :: add\_mem(int mcode, int bcode, char mname[26], char maddress[33], char mphone[10], int d1, int m1, int y1)

{

fstream file ;

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

memcode = mcode ;

bookcode = bcode ;

strcpy(name,mname) ;

strcpy(address,maddress) ;

strcpy(phone,mphone) ;

dd = d1 ;

mm = m1 ;

yy = y1 ;

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

file.close() ;

}

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

THIS FUNCTION DISPLAY THE RECORD FOR THE GIVEN MEMBER CODE.

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

void MEMBER :: display(int mcode)

{

fstream file ;

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

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

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

{

if (memcode == mcode)

{

gotoxy(5,3) ;

cout <<"Member Code # " <<mcode ;

gotoxy(5,4) ;

cout <<"~~~~~~~~~~~~~~~~~" ;

gotoxy(5,6) ;

cout <<"Name : " <<name ;

gotoxy(5,7) ;

cout <<"Phone : " <<phone ;

gotoxy(5,8) ;

cout <<"Address : " <<address ;

break ;

}

}

file.close() ;

}

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

THIS FUNCTION DISPLAY THE LIST OF THE MEMBERS

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

void MEMBER :: list(void)

{

clrscr();

textbackground(3);

clrscr() ;

BOOK B ;

int row = 6 , found=0, flag=0 ;

char ch ;

gotoxy(32,2) ;

cout <<"LIST OF MEMBERS" ;

gotoxy(31,3) ;

cout <<"~~~~~~~~~~~~~~~~~" ;

gotoxy(1,4) ;

cout <<"CODE BOOK CODE NAME PHONE ADDRESS" ;

gotoxy(1,5) ;

cout<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~”;cout<<”~~~~~~~~~~~~~~~~~~~~~~~~~~~" ;

fstream file ;

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

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

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

{

flag = 0 ;

delay(20) ;

found = 1 ;

gotoxy(2,row) ;

cout <<memcode ;

gotoxy(10,row) ;

cout <<bookcode ;

gotoxy(19,row) ;

cout <<name ;

gotoxy(38,row) ;

cout <<phone ;

gotoxy(50,row);

cout<<address;

textbackground(WHITE) ; textcolor(BLACK) ;

gotoxy(7,row+1) ;

if (bookcode == 0)

cprintf("BOOK NAME: (Not Issued)") ;

else

{

cprintf("BOOK NAME: %s",B.bookname(bookcode)) ;

gotoxy(42,row+1) ;

cprintf("Date of return: ") ;

textcolor(BLACK+BLINK) ;

cprintf("%d/%d/%d",dd,mm,yy) ;

}

textbackground(BLACK) ; textcolor(LIGHTGRAY) ;

if ( row == 22 )

{

flag = 1 ;

row = 6 ;

gotoxy(1,25) ;

cout <<"Press any key to continue or Press <ESC> to exit" ;

ch = getch() ;

if (ch == 27)

break ;

clrscr() ;

gotoxy(32,2) ;

cout <<"LIST OF MEMBERS" ;

gotoxy(31,3) ;

cout <<"~~~~~~~~~~~~~~~~~" ;

gotoxy(1,4) ;

cout <<"CODE BOOK CODE NAME “;

cout<<”PHONE" ;

gotoxy(1,5) ;

cout<<"~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~”;cout<<”~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~" ;

}

else

row = row + 2 ;

}

if (!found)

{

gotoxy(5,10) ;

cout <<"\7Records not found" ;

}

if (!flag)

{

gotoxy(1,25) ;

cout <<"Press any key to continue" ;

getche() ;

}

file.close () ;

}

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

THIS FUNCTION GIVES DATA TO ADD RECORD IN THE BOOK FILE.

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

void WORKING :: add\_book(void)

{

clrscr();

textbackground(2);

clrscr();

if (!reccount()) // MEMBER FUNCTION OF BOOK

{

add\_new\_book(0,"null","null",0.0,0,0) ;

BOOK::delete\_rec(0) ;

}

char ch ;

int tcode, tcopies, tavail ;

char tname[33], tauthor[26] ;

float tprice=0.0 ;

do

{

int found=0, valid=0 ;

int tc ;

float t2=0.0 ;

char t[10], t1[10] ;

clrscr() ;

gotoxy(29,3) ;

cout <<"ADDITION OF THE BOOKS" ;

gotoxy(29,4) ;

cout <<"~~~~~~~~~~~~~~~~~~~~~" ;

gotoxy(72,1) ;

cout <<"<0>=Exit" ;

gotoxy(5,25) ;

cout <<"Enter code no. of the book" ;

gotoxy(5,5) ;

cout <<"Code no. " ;

gets(t) ;

tc = atoi(t) ;

tcode = tc ;

if (tcode == 0)

return ;

if (book\_found(tcode))

{

found = 1 ;

gotoxy(19,8) ;

cout <<bookname(tcode) ;

gotoxy(19,9) ;

cout <<authorname(tcode) ;

gotoxy(22,10) ;

cout <<bookprice(tcode) ;

}

gotoxy(5,8) ;

cout <<"Book Name : " ;

gotoxy(5,9) ;

cout <<"Author Name : " ;

gotoxy(5,10) ;

cout <<"Price : Rs." ;

gotoxy(5,12) ;

cout <<"Copies : " ;

valid = 0 ;

while (!valid && !found)

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter the name of the book" ;

gotoxy(19,8) ; clreol() ;

gets(tname) ;

strupr(tname) ;

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

return ;

if (strlen(tname) < 1 || strlen(tname) > 32)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout <<"\7Enter correctly (Range: 1..32)" ;

getch() ;

}

}

valid = 0 ;

while (!valid && !found)

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter the author's name of the book" ;

gotoxy(19,9) ; clreol() ;

gets(tauthor) ;

strupr(tauthor) ;

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

return ;

if (strlen(tauthor) < 1 || strlen(tauthor) > 25)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout <<"\7Enter correctly (Range: 1..25)" ;

getch() ;

}

}

valid = 0 ;

while (!valid && !found)

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter the price of the book" ;

gotoxy(22,10) ; clreol() ;

gets(t1) ;

t2 = atof(t1) ;

tprice = t2 ;

if (t1[0] == '0')

return ;

if (tprice < 1 || tprice > 9999)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout <<"\7Enter correctly" ;

getch() ;

}

}

valid = 0 ;

while (!valid)

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter no. of copies of book to be added" ;

gotoxy(19,12) ; clreol() ;

gets(t) ;

tc = atoi(t) ;

tcopies = tc ;

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

return ;

if (tcopies < 1 || tcopies > 50)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout <<"\7Enter correctly" ;

getch() ;

}

}

tavail = available(tcode) + tcopies ;

tcopies = no\_of\_copies(tcode) + tcopies ;

gotoxy(5,25) ; clreol() ;

do

{

gotoxy(5,15) ; clreol() ;

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

ch = getche() ;

ch = toupper(ch) ;

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

if (ch == 'Y')

{

if (found)

update\_copies(tcode,tcopies,tavail) ;

else

add\_new\_book(tcode,tname,tauthor,tprice,tcopies,tavail) ;

}

do

{

gotoxy(5,17) ; clreol() ;

cout <<"Do you want to add more (y/n) : " ;

ch = getche() ;

ch = toupper(ch) ;

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

} while (ch == 'Y') ;

}

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

THIS FUNCTION GIVES DATA TO ADD RECORD IN THE MEMBER FILE

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

void WORKING :: add\_member(void)

{

clrscr();

textbackground(3);

char ch ;

int mcode, bcode ;

char mname[26], mphone[10], maddress[33] ;

int d1, m1, y1 ;

mcode = lastcode() ;

mcode++ ;

do

{

int valid=0 ;

clrscr() ;

gotoxy(28,3) ;

cout <<"ADDITION OF THE MEMBERS" ;

gotoxy(28,4) ;

cout <<"~~~~~~~~~~~~~~~~~~~~~~~" ;

gotoxy(72,1) ;

cout <<"<0>=Exit" ;

gotoxy(5,7) ;

cout <<"Member Code # " <<mcode ;

gotoxy(5,8) ;

cout <<"~~~~~~~~~~~~~~~~~" ;

gotoxy(5,10) ;

cout <<"Name : " ;

gotoxy(5,12) ;

cout <<"Phone : " ;

gotoxy(5,14) ;

cout <<"Address : " ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter the name of the New Member" ;

gotoxy(15,10) ; clreol() ;

gets(mname) ;

strupr(mname) ;

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

return ;

if (strlen(mname) < 1 || strlen(mname) > 25)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout <<"\7Enter correctly (Range: 1..25)" ;

getch() ;

}

} while (!valid) ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter Phone no. of the Member or Press <ENTER>”;

cout<<”for none" ;

gotoxy(15,12) ; clreol() ;

gets(mphone) ;

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

return ;

if ((strlen(mphone) < 8 && strlen(mphone) > 0) || (strlen(mphone) > 10))

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout <<"\7Enter correctly" ;

getch() ;

}

} while (!valid) ;

if (strlen(mphone) == 0)

strcpy(mphone,"-") ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter the address of the New Member" ;

gotoxy(15,14) ; clreol() ;

gets(maddress) ;

strupr(maddress) ;

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

return ;

if (strlen(maddress) < 1 || strlen(maddress) > 32)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout <<"\7Enter correctly (Range: 1..32)" ;

getch() ;

}

} while (!valid) ;

gotoxy(5,25) ; clreol() ;

do

{

gotoxy(5,17) ; clreol() ;

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

ch = getche() ;

ch = toupper(ch) ;

if (ch == '0')

return ;

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

if (ch == 'Y')

{

bcode = 0 ;

d1 = 0 ;

m1 = 0 ;

y1 = 0 ;

add\_mem(mcode,bcode,mname,maddress,mphone,d1,m1,y1) ;

mcode++ ;

}

do

{

gotoxy(5,19) ; clreol() ;

cout <<"Do you want to add more (y/n) : " ;

ch = getche() ;

ch = toupper(ch) ;

if (ch == '0')

return ;

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

} while (ch == 'Y') ;

}

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

THIS FUNCTION ISSUES THE BOOK

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

void WORKING :: issuebook(void)

{

clrscr();

textbackground(5);

BOOK B ;

MEMBER M ;

DATE D ;

char t1code[33], ch ;

int t2code=0, tcode=0, mcode=0 ;

int valid ;

int d1, m1, y1 ;

struct date d;

getdate(&d);

d1 = d.da\_day ;

m1 = d.da\_mon ;

y1 = d.da\_year ;

do

{

valid = 1 ;

while (1)

{

clrscr() ;

gotoxy(5,2) ;

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

gotoxy(72,1) ;

cout <<"<0>=Exit" ;

gotoxy(5,5) ;

cout <<"Enter Code or Name of the Book to be issued" ;

gotoxy(5,6) ;

cout <<" or " ;

gotoxy(5,7) ;

cout <<"Press <ENTER> for help " ;

gets(t1code) ;

if (t1code[0] == '0')

return ;

if (strlen(t1code) == 0)

B.list() ;

else

break ;

}

t2code = atoi(t1code) ;

tcode = t2code ;

if ((tcode == 0 && !bookname\_found(t1code)) || (tcode != 0 && !book\_found(tcode)))

{

valid = 0 ;

gotoxy(5,10) ;

cout <<"\7Record not found" ;

gotoxy(5,11) ;

cout <<"Press <ESC> to exit or any other key to continue" ;

ch = getch() ;

if (ch == 27)

return ;

}

} while (!valid) ;

if (tcode == 0)

tcode = bookcodeof(t1code) ;

if (!available(tcode))

{

gotoxy(5,10) ;

cout <<"\7Sorry! Book (" <<bookname(tcode) <<") is not available" ;

gotoxy(5,11) ;

cout <<"Kindly issue any other Book" ;

gotoxy(5,12) ;

cout <<"See List of Books" ;

getch() ;

return ;

}

do

{

valid = 1 ;

while (1)

{

clrscr() ;

gotoxy(72,1) ;

cout <<"<0>=Exit" ;

gotoxy(5,2) ;

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

gotoxy(5,5) ;

cout <<"Book Name: " <<bookname(tcode) ;

gotoxy(5,7) ;

cout <<"Enter Code no. of the Member" ;

gotoxy(5,8) ;

cout <<" or " ;

gotoxy(5,9) ;

cout <<"Press <ENTER> for help " ;

gets(t1code) ;

if (t1code[0] == '0')

return ;

if (strlen(t1code) == 0)

M.list() ;

else

break ;

}

t2code = atoi(t1code) ;

mcode = t2code ;

if (mcode == 0)

{

valid = 0 ;

gotoxy(5,25) ;

cout <<"\7Enter Correctly" ;

getch() ;

}

if (!member\_found(mcode) && valid)

{

valid = 0 ;

gotoxy(5,13) ;

cout <<"\7Record not found" ;

gotoxy(5,14) ;

cout <<"Press <ESC> to exit or any other key to continue" ;

ch = getch() ;

if (ch == 27)

return ;

}

} while (!valid) ;

int tcopies, tavail ;

tcopies = no\_of\_copies(tcode) ; // member function of BOOK

tavail = available(tcode) - 1 ; // member function of BOOK

update\_copies(tcode,tcopies,tavail) ; // member function of BOOK

D.extend\_date(d1,m1,y1,3) ;

d1 = D.day ;

m1 = D.mon ;

y1 = D.year ;

update\_book(mcode,tcode,d1,m1,y1) ; // member function of MEMBER

gotoxy(5,13) ;

cout <<"\7Book is issued to " <<membername(mcode) ;

gotoxy(5,15) ;

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

getch() ;

}

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

THIS FUNCTION RETURNS THE BOOK FOR THE MEMBER

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

void WORKING :: returnbook(void)

{

clrscr();

textbackground(6);

clrscr();

MEMBER M ;

char t1code[5], ch ;

int t2code=0, mcode=0, valid ;

int d1, m1, y1 ;

struct date d;

getdate(&d);

d1 = d.da\_day ;

m1 = d.da\_mon ;

y1 = d.da\_year ;

do

{

valid = 1 ;

while (1)

{

clrscr() ;

gotoxy(72,1) ;

cout <<"<0>=Exit" ;

gotoxy(5,2) ;

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

gotoxy(5,7) ;

cout <<"Enter Code no. of the Member who wants to return”;

cout<<”book" ;

gotoxy(5,8) ;

cout <<" or " ;

gotoxy(5,9) ;

cout <<"Press <ENTER> for help " ;

gets(t1code) ;

if (t1code[0] == '0')

return ;

if (strlen(t1code) == 0)

M.list() ;

else

break ;

}

t2code = atoi(t1code) ;

mcode = t2code ;

if (mcode == 0)

{

valid = 0 ;

gotoxy(5,25) ;

cout <<"\7Enter Correctly" ;

getch() ;

}

if (!member\_found(mcode) && valid)

{

valid = 0 ;

gotoxy(5,13) ;

cout <<"\7Record not found" ;

gotoxy(5,14) ;

cout <<"Press <ESC> to exit or any other key to continue" ;

ch = getch() ;

if (ch == 27)

return ;

}

if (!issued(mcode) && valid)

{

valid = 0 ;

gotoxy(5,13) ;

cout <<"\7Member have no book to return" ;

gotoxy(5,14) ;

cout <<"Press <ESC> to exit or any other key to continue" ;

ch = getch() ;

if (ch == 27)

return ;

}

} while (!valid) ;

int bcode, tcopies, tavail ;

bcode = issued(mcode) ;

gotoxy(5,13) ;

cout <<"Book Code : " <<bcode ;

gotoxy(5,14) ;

cout <<"Book Name : " <<bookname(bcode) ;

tcopies = no\_of\_copies(bcode) ;

tavail = available(bcode) + 1 ;

int f ;

f = fine(mcode) ;

if (f != 0)

{

gotoxy(5,16) ;

cout <<"You have to pay a fine of Rs." <<f ;

gotoxy(5,17) ;

cout <<"Please do not delay the Return of Book again" ;

}

update\_copies(bcode,tcopies,tavail) ;

update\_book(mcode,0,0,0,0) ;

gotoxy(5,19) ;

cout <<"\7Book has been returned" ;

getch() ;

}

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

THIS FUNCTION GIVES DATA TO MODIFY THE BOOK RECORD

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

void WORKING :: modify\_book(void)

{

clrscr();

textbackground(1);

BOOK B ;

char t1code[5], tname[33], tauthor[26], \*t1, ch ;

int t2code=0, tcode=0 ;

float t2=0.0, tprice=0.0 ;

int valid ;

do

{

valid = 1 ;

while (1)

{

clrscr() ;

gotoxy(72,1) ;

cout <<"<0>=Exit" ;

gotoxy(5,5) ;

cout <<"Enter Code or Name of the Book to be modified" ;

gotoxy(5,6) ;

cout <<" or " ;

gotoxy(5,7) ;

cout <<"Press <ENTER> for help " ;

gets(t1code) ;

if (t1code[0] == '0')

return ;

if (strlen(t1code) == 0)

B.list() ;

else

break ;

}

t2code = atoi(t1code) ;

tcode = t2code ;

if ((tcode == 0 && !bookname\_found(t1code)) || (tcode != 0 && !book\_found(tcode)))

{

valid = 0 ;

gotoxy(5,10) ;

cout <<"\7Record not found" ;

gotoxy(5,11) ;

cout <<"Press <ESC> to exit or any other key to continue" ;

ch = getch() ;

if (ch == 27)

return ;

}

} while (!valid) ;

if (tcode == 0)

tcode = bookcodeof(t1code) ;

clrscr() ;

gotoxy(72,1) ;

cout <<"<0>=Exit" ;

BOOK::display(tcode) ;

do

{

gotoxy(5,13) ; clreol() ;

cout <<"Do you want to modify this record (y/n) : " ;

ch = getche() ;

ch = toupper(ch) ;

if (ch == '0')

return ;

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

if (ch == 'N')

return ;

gotoxy(5,16) ;

cout <<"Book Name : " ;

gotoxy(5,17) ;

cout <<"Author Name : " ;

gotoxy(5,18) ;

cout <<"Price : Rs." ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter the name of the book or <ENTER> for no change" ;

gotoxy(19,16) ; clreol() ;

gets(tname) ;

strupr(tname) ;

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

return ;

if (strlen(tname) > 32)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout <<"\7Enter correctly (Range: 1..32)" ;

getch() ;

}

} while (!valid) ;

if (strlen(tname) == 0)

strcpy(tname,bookname(tcode)) ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter the author's name or <ENTER> for no change" ;

gotoxy(19,17) ; clreol() ;

gets(tauthor) ;

strupr(tauthor) ;

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

return ;

if (strlen(tauthor) > 25)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout <<"\7Enter correctly (Range: 1..25)" ;

getch() ;

}

} while (!valid) ;

if (strlen(tauthor) == 0)

strcpy(tauthor,authorname(tcode)) ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter price or <ENTER> for no change" ;

gotoxy(22,18) ; clreol() ;

gets(t1) ;

t2 = atof(t1) ;

tprice = t2 ;

if (t1[0] == '0')

return ;

if (strlen(t1) == 0)

break ;

if (tprice < 1 || tprice > 9999)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout <<"\7Enter correctly" ;

getch() ;

}

} while (!valid) ;

if (strlen(t1) == 0)

tprice = bookprice(tcode) ;

gotoxy(5,25) ; clreol() ;

do

{

gotoxy(5,20) ; clreol() ;

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

ch = getche() ;

ch = toupper(ch) ;

if (ch == '0')

return ;

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

if (ch == 'N')

return ;

BOOK::modify(tcode,tname,tauthor,tprice) ;

gotoxy(5,23) ;

cout <<"\7Record Modified" ;

getch() ;

}

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

THIS FUNCTION GIVES DATA TO MODIFY THE MEMBER RECORD

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

void WORKING :: modify\_member(void)

{

clrscr();

textbackground(1);

MEMBER M ;

char m1code[10], mname[26], mphone[10], maddress[33], ch ;

int m2code=0, mcode=0 ;

int valid ;

do

{

valid = 1 ;

while (1)

{

clrscr() ;

gotoxy(72,1) ;

cout <<"<0>=Exit" ;

gotoxy(5,7) ;

cout <<"Enter Code no. of the Member to be Modify" ;

gotoxy(5,8) ;

cout <<" or " ;

gotoxy(5,9) ;

cout <<"Press <ENTER> for help " ;

gets(m1code) ;

m2code = atoi(m1code) ;

mcode = m2code ;

if (m1code[0] == '0')

return ;

if (strlen(m1code) == 0)

M.list() ;

else

break ;

}

if (mcode == 0)

{

valid = 0 ;

gotoxy(5,25) ;

cout <<"\7Enter Correctly" ;

getch() ;

}

if (valid && !member\_found(mcode))

{

valid = 0 ;

gotoxy(5,13) ;

cout <<"\7Record not found" ;

gotoxy(5,14) ;

cout <<"Press <ESC> to exit or any other key to continue" ;

ch = getch() ;

if (ch == 27)

return ;

}

} while (!valid) ;

clrscr() ;

gotoxy(72,1) ;

cout <<"<0>=Exit" ;

MEMBER::display(mcode) ;

do

{

gotoxy(5,10) ; clreol() ;

cout <<"Do you want to modify this record (y/n) : " ;

ch = getche() ;

ch = toupper(ch) ;

if (ch == '0')

return ;

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

if (ch == 'N')

return ;

gotoxy(5,13) ;

cout <<"Name : " ;

gotoxy(5,14) ;

cout <<"Phone : " ;

gotoxy(5,15) ;

cout <<"Address : " ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter the name of the member or <ENTER> for no change" ;

gotoxy(19,13) ; clreol() ;

gets(mname) ;

strupr(mname) ;

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

return ;

if (strlen(mname) > 25)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout <<"\7Enter correctly (Range: 1..25)" ;

getch() ;

}

} while (!valid) ;

if (strlen(mname) == 0)

strcpy(mname,membername(mcode)) ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter the Phone no. of Member or <ENTER> for no change" ;

gotoxy(19,14) ; clreol() ;

gets(mphone) ;

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

return ;

if ((strlen(mphone) < 8 && strlen(mphone) > 0) || (strlen(mphone) > 10))

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout <<"\7Enter correctly" ;

getch() ;

}

} while (!valid) ;

if (strlen(mphone) == 0)

strcpy(mphone,memberphone(mcode)) ;

do

{

valid = 1 ;

gotoxy(5,25) ; clreol() ;

cout <<"Enter the address of the member or <ENTER> for no”;

cout<<”change" ;

gotoxy(19,15) ; clreol() ;

gets(maddress) ;

strupr(maddress) ;

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

return ;

if (strlen(maddress) > 32)

{

valid = 0 ;

gotoxy(5,25) ; clreol() ;

cout <<"\7Enter correctly (Range: 1..32)" ;

getch() ;

}

} while (!valid) ;

if (strlen(maddress) == 0)

strcpy(maddress,memberaddress(mcode)) ;

gotoxy(5,25) ; clreol() ;

do

{

gotoxy(5,18) ; clreol() ;

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

ch = getche() ;

ch = toupper(ch) ;

if (ch == '0')

return ;

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

if (ch == 'N')

return ;

MEMBER::modify(mcode,mname,mphone,maddress) ;

gotoxy(5,23) ;

cout <<"\7Record Modified" ;

getch() ;

}

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

THIS FUNCTION GIVES BOOK CODE TO DELETE THE BOOK RECORD

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

void WORKING :: delete\_book(void)

{

clrscr();

textbackground(2);

BOOK B ;

char t1code[5], tname[33], tauthor[26], ch ;

int t2code=0, tcode=0 ;

int valid ;

do

{

valid = 1 ;

while (1)

{

clrscr() ;

gotoxy(72,1) ;

cout <<"<0>=Exit" ;

gotoxy(5,5) ;

cout <<"Enter Code or Name of the Book to be Deleted" ;

gotoxy(5,6) ;

cout <<" or " ;

gotoxy(5,7) ;

cout <<"Press <ENTER> for help " ;

gets(t1code) ;

if (t1code[0] == '0')

return ;

if (strlen(t1code) == 0)

B.list() ;

else

break ;

}

t2code = atoi(t1code) ;

tcode = t2code ;

if ((tcode == 0 && !bookname\_found(t1code)) || (tcode != 0 && !book\_found(tcode)))

{

valid = 0 ;

gotoxy(5,10) ;

cout <<"\7Record not found" ;

gotoxy(5,11) ;

cout <<"Press <ESC> to exit or any other key to continue" ;

ch = getch() ;

if (ch == 27)

return ;

}

} while (!valid) ;

if (tcode == 0)

tcode = bookcodeof(t1code) ;

clrscr() ;

gotoxy(72,1) ;

cout <<"<0>=Exit" ;

BOOK::display(tcode) ;

do

{

gotoxy(5,13) ; clreol() ;

cout <<"Do you want to delete this record (y/n) : " ;

ch = getche() ;

ch = toupper(ch) ;

if (ch == '0')

return ;

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

if (ch == 'N')

return ;

int tavail, tcopies ;

tavail = available(tcode) ;

tcopies = no\_of\_copies(tcode) ;

if (tavail != tcopies)

{

gotoxy(5,15) ;

cout <<"\7Record cannot be deleted. This book is issued." ;

getch() ;

return ;

}

BOOK::delete\_rec(tcode) ;

gotoxy(5,23) ;

cout <<"\7Record Deleted" ;

getch() ;

}

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

THIS FUNCTION GIVES MEMBER CODE TO DELETE THE MEMBER RECORD

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

void WORKING :: delete\_member(void)

{

clrscr();

textbackground(2);

MEMBER M ;

char m1code[5], mname[26], mphone[10], maddress[33], ch ;

int m2code=0, mcode=0 ;

int valid ;

do

{

valid = 1 ;

while (1)

{

clrscr() ;

gotoxy(72,1) ;

cout <<"<0>=Exit" ;

gotoxy(5,7) ;

cout <<"Enter Code no. of the Member to be Deleted" ;

gotoxy(5,8) ;

cout <<" or " ;

gotoxy(5,9) ;

cout <<"Press <ENTER> for help " ;

gets(m1code) ;

m2code = atoi(m1code) ;

mcode = m2code ;

if (m1code[0] == '0')

return ;

if (strlen(m1code) == 0)

M.list() ;

else

break ;

}

if (mcode == 0)

{

valid = 0 ;

gotoxy(5,25) ;

cout <<"\7Enter Correctly" ;

getch() ;

}

if (valid && !member\_found(mcode))

{

valid = 0 ;

gotoxy(5,13) ;

cout <<"\7Record not found" ;

gotoxy(5,14) ;

cout <<"Press <ESC> to exit or any other key to continue" ;

ch = getch() ;

if (ch == 27)

return ;

}

} while (!valid) ;

clrscr() ;

gotoxy(72,1) ;

cout <<"<0>=Exit" ;

MEMBER::display(mcode) ;

do

{

gotoxy(5,10) ; clreol() ;

cout <<"Do you want to Delete this record (y/n) : " ;

ch = getche() ;

ch = toupper(ch) ;

if (ch == '0')

return ;

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

if (ch == 'N')

return ;

if (issued(mcode))

{

gotoxy(5,15) ;

cout <<"\7Record cannot be delete. Member has a book" ;

getch() ;

return ;

}

MEMBER::delete\_rec(mcode) ;

gotoxy(5,23) ;

cout <<"\7Record Modified" ;

getch() ;

}

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

THIS FUNCTION IS TO ENTER PASSWORD WHICH IS “LIBRARY” or “library”

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

int passwords()

{

char p1,p2,p3,p4,p5,p6,p7;

gotoxy(30,10);

cout<<"ENTER THE PASSWORD.....\n";

gotoxy(34,12);

p1=getch();

cout<<"\*";

p2=getch();

cout<<"\*";

p3=getch();

cout<<"\*";

p4=getch();

cout<<"\*";

p5=getch();

cout<<"\*";

p6=getch();

cout<<"\*";

p7=getch();

cout<<"\*";

getch();

gotoxy(30,20);

if((p1=='l'||p1=='L')&&(p2=='i'||p2=='I')&&(p3=='b'||p3=='B')&&(p4=='r'||p4=='R')&&(p5=='a'||p5=='A')&&(p6=='r'||p6=='R')&&(p7=='y'||p7=='Y'))

return 1;

else

return 0;

}

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

MAIN FUNCTION CALLING INTRODUCTION AND MAIN MENU

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

void main(void)

{

textbackground(1);

MENU menu ;

cout<<"\n\t%% %% %%%%%%% %% %% %%%%%%”; cout<<”%%%%%% %%%% %%%% %%%%%%% ";

cout<<"\n\t%% %% %% %% %% %% %% %% %%”;

cout<<” %%% %% %% ";

cout<<"\n\t%% %% %% %%%%% %% %% %% %% %%”;

cout<<”%% %%% %% %%%%% ";

cout<<"\n\t%% %% %% %% %% %% %% %% %% %% “;

cout<<”%% %% ";

cout<<"\n\t%%%%%%%%%% %%%%%%% %%%%%%% %%%%%%”;

cout<”%%%%%%% %%%%%% %% %% %%%%%%% ";

cout<<"\n\n\t\t\t $$$$$$$$ $$$$$ ";

cout<<"\n\t\t\t $$ $ $ ";

cout<<"\n\t\t\t $$ $$$$$ ";

cout<<"\n\n\tCOMPUTER PROJECT (\*\*\*\*\*\*\*\* Library Management”; cout<<”\*\*\*\*\*\*\*\*)";

cout<<"\n\n\t\t\t $$$$$ Made BY: $$$$$";

cout<<"\n\n\n\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~”;

cout<<”~~~~~\n";

cout<<"\n Parmjeet Singh Kainth"<<"\t Class--> XII-Science ";

cout<<"\n\t\t~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~\n";

cout<<"\n\n\n\n\t\t\tPress any key to continue......";

getch();

clrscr();

cout<<"\n\n\t\t\t Ganga International SCHOOL Khanouri\n" ;

cout<<"\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n";

cout<<"\n\n\t\t LIBRARTY PROJECT\n";

cout<<"\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* \n";

int passwords();

if(!passwords())

{

for(int i=0;i<2;i++)

{

textbackground(2);

clrscr();

cout<<"\nWrong password try once more.....\n";

if(passwords())

{

goto last;

}

else

{

clrscr();

cout<<"\n\n\t\t\t all attempts failed.....";

cout<<"\n\n\n\t\t\t see you.................. ";

exit(0);

}

}

cout<<"\t\t\t sorry all attempts failed............. \n \t\t\tinactive";

}

else

{

clrscr();

last:;

clrscr();

cout<<"loading……………….\n";

menu.main\_menu() ;

}

}

****



















































