LibManage

Library Management System

C++

Computer Science Project

Kartik Singhal

Roll No. ___________

LibManage

CERTIFICATE

This is to certify that **Kartik Singhal** *of Class* **XII A** *has completed his project under my guidance and supervision. He has worked successfully under my provision and has shown utmost sincerity in completion of this project.*

Mrs. Shipra Ranjan
(Computer Sc. Teacher)

LibManage

ACKNOWLEDGEMENTS

I sincerely acknowledge the guidance and help bestowed on me by my Computer Science Teacher **Mrs. Shipra Ranjan** during my project work.

I give my earnest thanks to our Computer Laboratory Assistant **Mr. Manoj Kumar** for the help rendered to me in the laboratory.

My thanks are due to Microsoft for providing such a great IDE – Visual C++ 2008 Express Edition – as a free download from its website and also to its developers; and the team behind Microsoft Office Word 2007 for developing such a great package without which this documentation would not have been possible.

I also want to thank the Computer Science staff of our school, my friends and my family for the encouragement and support I got from them.

Kartik Singhal

LibManage

CONTENTS

LibManage

INTRODUCTION

EXISTING SYSTEM

In the existing system, libraries are managed manually using registers, pens, etc. It becomes difficult for librarians to maintain a record of a large no. of books and equally large no. of members of the library. The issuing and returning procedure is also a bit slow.

PROPOSED SYSTEM

In the proposed system, issuing and returning a book becomes automatic. The system automatically generates current system date and can maintain record of last dates of returning books and calculates fine for any delay in the returning of books. It shows lists of all the books available in the library and all the members of the library. New books and members can be added; record of old books can be modified and removed from the system. It even provides facility for adding more copies or removing damaged copies of old books in the library.

The system has a fast and responsive interface and is effective in reduction of time taken for book issue and return procedure.

LibManage

DATA DICTIONERY

S. No.	Field	Data Type	Size (in bytes)	Description	Belongs to
1	bookcode	int	4	unique code for each book	BOOK.DAT
2	copies	int	4	total no. of copies	BOOK.DAT
3	name	char []	33	name of book	BOOK.DAT
4	author	char []	26	name of author	BOOK.DAT
5	price	float	4	price of book	BOOK.DAT
6	avail	int	4	no. of copies available	BOOK.DAT
7	memcode	int	4	unique code for each member	MEMBER.DAT
8	bookcode	int	4	code of book issued to member	MEMBER.DAT
9	name	char []	26	name of member	MEMBER.DAT
10	phone	char []	10	phone no. of member	MEMBER.DAT
11	address	char []	33	address of member	MEMBER.DAT
12	returnDate	Date	28	date of return of book	MEMBER.DAT

LibManage

REQUIREMENTS

HARDWARE REQUIREMENTS

- Any x86 machine running one of the two:
  - Intel Pentium III or above
  - AMD Athlon XP or above

SOFTWARE REQUIREMENTS

- For running the core exe
  - Windows 2000 Professional with Service pack 4 or above, Windows XP Service Pack 2 or above (32-bit editions only) or Windows Vista (32-bit editions only)
- For compiling the source code
  - Microsoft Visual C++ 2008 (Standard or Express Edition) on one of the above platforms

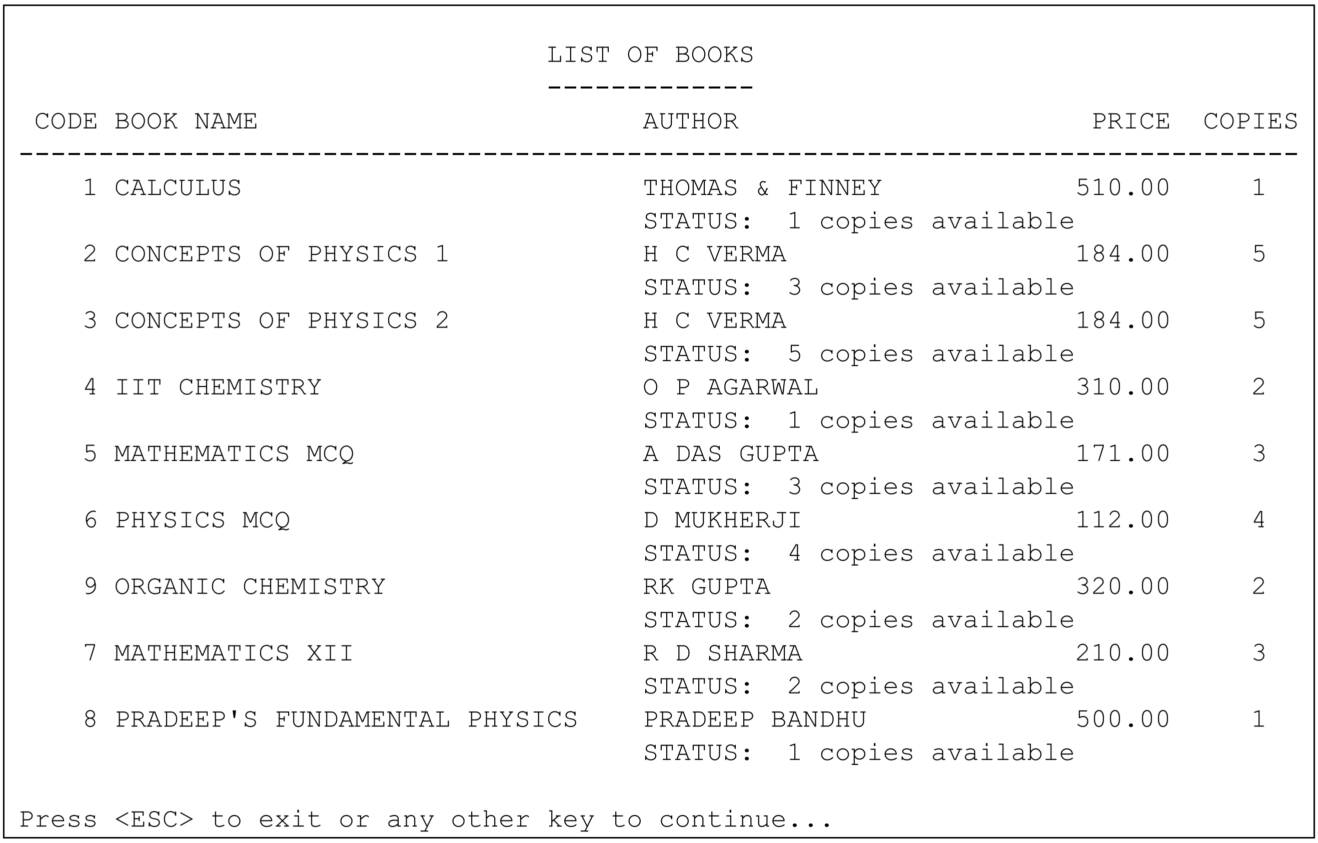
LibManage

REPORTS

BOOK REPORT

Current status of all the books available in the library is reported as and when required on the screen.

The following sample shows the same:



```
                                 LIST OF BOOKS
                                 -------------
 CODE BOOK NAME                        AUTHOR                      PRICE  COPIES
--------------------------------------------------------------------------------
    1 CALCULUS                         THOMAS & FINNEY            510.00     1
                                       STATUS:  1 copies available
    2 CONCEPTS OF PHYSICS 1            H C VERMA                  184.00     5
                                       STATUS:  3 copies available
    3 CONCEPTS OF PHYSICS 2            H C VERMA                  184.00     5
                                       STATUS:  5 copies available
    4 IIT CHEMISTRY                    O P AGARWAL                310.00     2
                                       STATUS:  1 copies available
    5 MATHEMATICS MCQ                  A DAS GUPTA                171.00     3
                                       STATUS:  3 copies available
    6 PHYSICS MCQ                      D MUKHERJI                 112.00     4
                                       STATUS:  4 copies available
    9 ORGANIC CHEMISTRY                RK GUPTA                   320.00     2
                                       STATUS:  2 copies available
    7 MATHEMATICS XII                  R D SHARMA                 210.00     3
                                       STATUS:  2 copies available
    8 PRADEEP'S FUNDAMENTAL PHYSICS    PRADEEP BANDHU             500.00     1
                                       STATUS:  1 copies available

Press <ESC> to exit or any other key to continue...
```

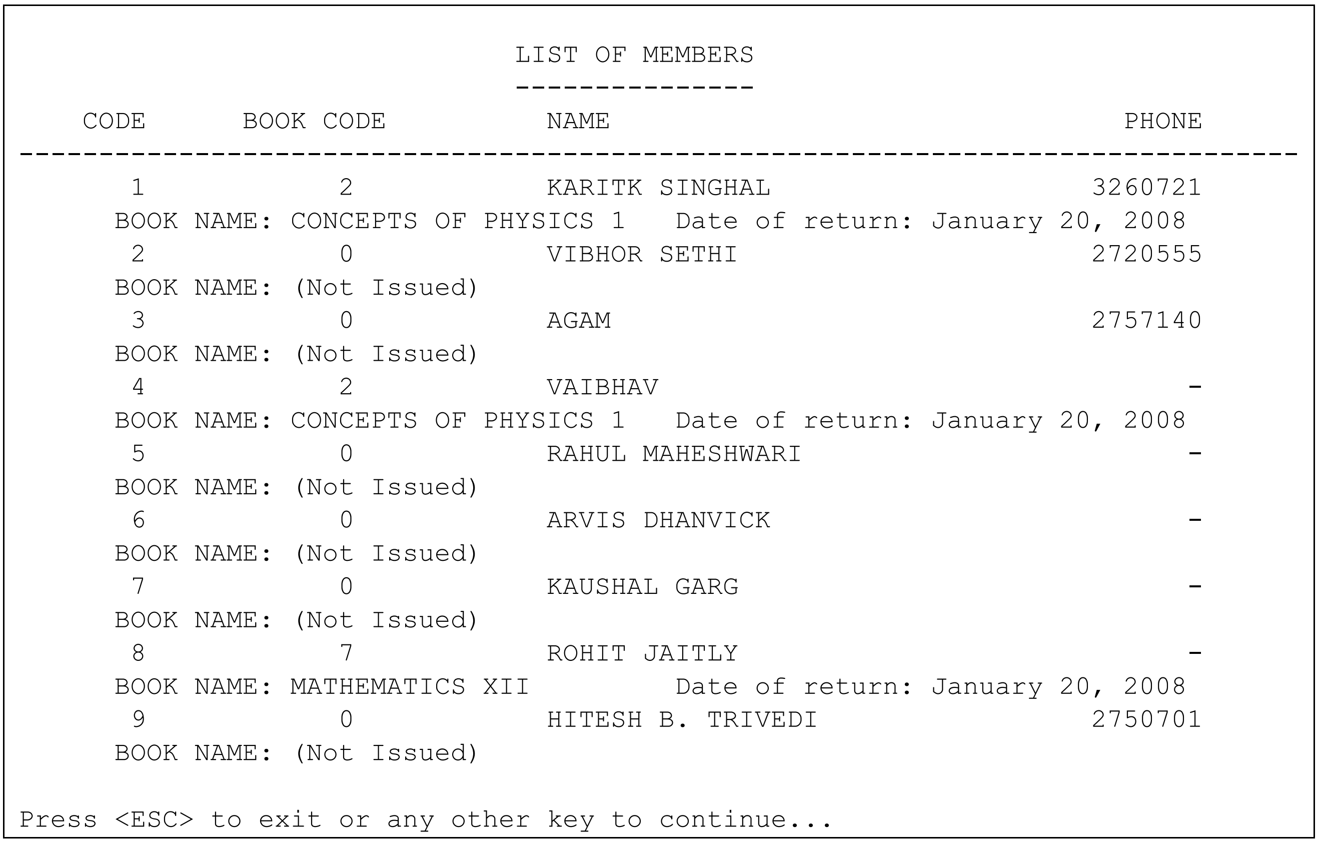
LibManage

REPORTS

MEMBER REPORT

Current status of all the members of the library is reported as and when required on the screen.

The following sample shows the same:



```

                               LIST OF MEMBERS
                               ---------------
    CODE      BOOK CODE          NAME                                   PHONE
--------------------------------------------------------------------------------
       1            2            KARITK SINGHAL                       3260721
      BOOK NAME: CONCEPTS OF PHYSICS 1   Date of return: January 20, 2008
       2            0            VIBHOR SETHI                         2720555
      BOOK NAME: (Not Issued)
       3            0            AGAM                                 2757140
      BOOK NAME: (Not Issued)
       4            2            VAIBHAV                                    -
      BOOK NAME: CONCEPTS OF PHYSICS 1   Date of return: January 20, 2008
       5            0            RAHUL MAHESHWARI                           -
      BOOK NAME: (Not Issued)
       6            0            ARVIS DHANVICK                             -
      BOOK NAME: (Not Issued)
       7            0            KAUSHAL GARG                               -
      BOOK NAME: (Not Issued)
       8            7            ROHIT JAITLY                               -
      BOOK NAME: MATHEMATICS XII         Date of return: January 20, 2008
       9            0            HITESH B. TRIVEDI                    2750701
      BOOK NAME: (Not Issued)

Press <ESC> to exit or any other key to continue...
```

LibManage

INSTALLATION PROCEDURE

TO RUN THE PROGRAM

- Simply run the LibManage.exe file
- The data files – BOOK.DAT and MEMBER.DAT – must be in the same folder as the main executable i.e. LibManage.exe

TO COMPILE THE SOURCE CODE

- You have two options:
    - The easy way:
        - Open the LibManage.sln solution file by double clicking on it or by opening it from the Visual C++ 2008 IDE
        - Build the project using the 'Build Solution' command and run it using 'Start Debugging' command
    - The hard way:
        - Make a new Empty Project in Visual Studio 2008
        - Add the source files – LibManage.cpp, LibManage.h and date.h – to the project
        - Copy files – conio2.h and libconio.a – to your project folder
        - Tell the environment from where to seek for the conio2 library by opening project Property Pages dialog ➡ Linker node ➡ General ➡ Additional Library Directories, set this option to the path where you have copied libconio.a file
        - Tell the name of the library file to the environment by opening project Property Pages dialog ➡ Linker node ➡ Input ➡ Additional Dependencies, type libconio.a in this field
        - Now, you are set to build and run the project using 'Build Solution' command and 'Start Debugging' command respectively
- Note that you actually require Microsoft Visual C++ 2008 Express Edition (which is available as a free download from http://www.microsoft.com/express/vc/) to compile the source of the project
- Sample data files are included in the Test Data directory which can be copied to project working directory and used for testing the project

LibManage

MENU CHOICES

The main menu provides the following choices:

- Introduction
    - Displays the introduction screen which is first displayed on running LibManage
- Add New Book(s)
    - To add new books to the library
    - Can also be used to add/remove copies of older books to/from the library
    - User needs to enter a new code for a new book or an older code to see the details and/or modify the number of copies of previously added books
- Add New Member(s)
    - To add new members to the library
    - User is presented with a self generated code which he/she cannot modify so as to keep a record of number of total members (current members + previous members who left membership) of the library
- Issue Book
    - To issue a book
    - User is presented with the choice of typing the code or the name of desired book and also with the option to see the list of books first and then enter the code or name of desired book
    - After selecting a book user can type the code of the member who wants that book or view the list of members first and then enter the member code
- Return Book
    - To return a previously issued book
    - User needs to type the code of the member who wants to return the issued book either directly or by seeing the list of members first and then entering the code
    - If the member returns the book after the return date, a fine amount is calculated and displayed on the screen
- List of Books
    - Displays a list of all the books available in the library
- List of Members
    - Displays a list of all the members of the library with the code and name of book issued to them and the return date
- Edit
    - Provides options for modifying the details or deleting the record of books and members of the library
    - Book record can be deleted only if its no copy is issued to any member
    - Member record can’t be deleted if he/she has not returned an issued book
- Quit
    - To exit from LibManage

LibManage

SOURCE CODE

LibManage.cpp

```
////////////////////////////////////////////
//                                        //
//   LibManage                            //
//   Library Management Sytem             //
//                                        //
//   Copyright (c) 2007 Kartik Singhal.   //
//   kartiksinghal@gmail.com              //
//   All rights reserved.                 //
//                                        //
////////////////////////////////////////////

//////////////////////////////////////////////////////////////////
// Files required include: (program won't run without these)    //
// * LibManage.cpp * (this source file)                         //
// * LibManage.h *    (header file which includes all the       //
//                     required classes)                        //
// * date.h *         (header file concerning date class        //
//                     and date functions)                      //
// * conio2.h *       (replacement for 16-bit conio.h)          //
// * libconio.a *     (library file for conio2.h)               //
//////////////////////////////////////////////////////////////////
// This program uses the following files for data input/output: //
// * BOOK.DAT *     (to hold book information)                  //
// * MEMBER.DAT *   (to hold member information)                //
// * TEMP.DAT  *    (to do certain temporary stuff)             //
//////////////////////////////////////////////////////////////////


#include "LibManage.h"

#define FINERATE   2              //the rate of fine in Rs. per day

// The names of data files can be changed here
#define BOOKFILE   "BOOK.DAT"
#define MEMBERFILE "MEMBER.DAT"
#define TEMPFILE   "TEMP.DAT"

char *tempText = NULL;            //pointer to hold temporary text data

// Program Execution starts
int main()
{
     Menu menu;
     menu.displayIntroduction();

     cout.setf(ios::showpoint | ios::fixed);
     cout<<setprecision(2);

     menu.displayMainMenu();

     delete tempText;

     return EXIT_SUCCESS;
} //End main()
```

```
///////////////////////////////////////////////////
// Function definitions pertaining to Menu Class //
///////////////////////////////////////////////////

// Displays the introductory screeen
void Menu::displayIntroduction(void)
{
     clrscr();

     gotoxy(33,6);                   cout<<"Welcome to";

     textcolor(BLACK); textbackground(WHITE);
     gotoxy(33,8);                 cprintf("LibManage");

     textcolor(LIGHTGRAY); textbackground(BLACK);
     gotoxy(15,10);cout<<"LibManage can maintain records of Books and Members.";


     gotoxy(15,13);cout<<"One member is allowed to issue one book at a time. If";
     gotoxy(15,14);cout<<"he/she does not return book upto 15 days, he/she has";
     gotoxy(15,15);cout<<"to pay a fine of Rs. "<<FINERATE<<"/- per day.";






     textcolor(LIGHTGRAY); textbackground(BLACK);
     gotoxy(27,22);          cprintf("Press any key to continue...");

     textcolor(LIGHTGRAY);

     _getch();
} //End displayIntroduction()

// Displays Main Menu
void Menu::displayMainMenu(void)
{
     char ch;
     while(1)
     {
          clrscr();
          gotoxy(29,6); cout<<"B O O K   L I B R A R Y";
          gotoxy(29,7); cout<<"-----------------------";
          gotoxy(30,10); cout<<"1. Introduction";
          gotoxy(30,11); cout<<"2. Add New Book(s)";
          gotoxy(30,12); cout<<"3. Add New Member(s)";
          gotoxy(30,13); cout<<"4. Issue Book";
          gotoxy(30,14); cout<<"5. Return Book";
          gotoxy(30,15); cout<<"6. List of Books";
          gotoxy(30,16); cout<<"7. List of Members";
          gotoxy(30,17); cout<<"8. Edit";
          gotoxy(30,18); cout<<"0. Quit";

          gotoxy(29,20);cout<<"Enter your choice: ";
          ch = _getche();

          if(ch == 27)
               break;
          else if(ch == '1')
               displayIntroduction();
          else if(ch == '2')
          {
               Working W;
               W.addBook();
          }
          else if(ch == '3')
```

```
        {
            Working W;
            W.addMember();
        }
        else if(ch == '4')
        {
            Working W;
            W.issueBook();
        }
        else if(ch == '5')
        {
            Working W;
            W.returnBook();
        }
        else if(ch == '6')
        {
            Book B;
            B.displayList();
        }
        else if(ch == '7')
        {
            Member M;
            M.displayList();
        }
        else if(ch == '8')
            displayEditMenu();
        else if(ch == '0')
            break;
    }
} //End displayMainMenu()

// Displays Edit Menu
void Menu::displayEditMenu(void)
{
    char ch;
    while(1)
    {
        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(32,17); cout<<"Enter your choice: ";
        ch = _getche();

        if(ch == 27)
            break;
        else if(ch == '1')
            displayEditBookMenu();
        else if(ch == '2')
            displayEditMemberMenu();
        else if(ch == '0')
            break;
    }
} //End displayEditMenu()

// Displays Book-Edit Menu
void Menu::displayEditBookMenu(void)
{
    char ch;
    while(1)
    {
        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.modifyBook();
          }
          else if(ch == '2')
          {
               Working W;
               W.deleteBook();
          }
          else if(ch == '0')
               break;
     }
} //End displayEditBookMenu()

// Displays Member-Edit Menu
void Menu::displayEditMemberMenu(void)
{
     char ch;
     while(1)
     {
          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.modifyMember();
          }
          else if(ch == '2')
          {
               Working W;
               W.deleteMember();
          }
          else if(ch == '0')
               break;
     }
} //End displayEditMemberMenu()


///////////////////////////////////////////////////
// Function definitions pertaining to Book Class //
///////////////////////////////////////////////////

// Returns 1 if book with given code is found
int Book::bookFound(int tcode)
{
     fstream file(BOOKFILE, 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;
} //End bookFound()

// Returns 1 if book with given name is found
int Book::booknameFound(char t1code[33])
{
    fstream file(BOOKFILE, ios::in);
    file.seekg(0, ios::beg);
    int found = 0;
    while( file.read( (char*)this, sizeof(Book) ) )
    {
        if( !strcmp(name, t1code) )
        {
            found = 1;
            break;
        }
    }
    file.close();
    return found;
} //End booknameFound()

// Returns the serial no. of given record from the data file
int Book::recordnoOf(int tcode)
{
    fstream file(BOOKFILE, 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;
} //End recordnoOf()

// Returns 1 if book is available
int Book::available(int tcode)
{
    fstream file(BOOKFILE, 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;
} //End available()

// Returns the total no. of copies of book
int Book::noOfCopiesOf(int tcode)
```

```
{
	fstream file(BOOKFILE, 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;
} //End noOfCopiesOf()

// Returns the name of book with given code
char* Book::booknameOf(int tcode)
{
	fstream file(BOOKFILE, ios::in);
	file.seekg(0, ios::beg);
	delete tempText;
	tempText = new char[33];
	while( file.read( (char*)this, sizeof(Book) ) )
	{
		if(bookcode == tcode)
		{
			strcpy(tempText, name);
			break;
		}
	}
	file.close();
	return tempText;
} //End booknameOf()

// Returns the name of author the book with given code
char* Book::authornameOf(int tcode)
{
	fstream file(BOOKFILE, ios::in);
	file.seekg(0, ios::beg);
	delete tempText;
	char *tempText = new char[26];
	while( file.read( (char*)this, sizeof(Book) ) )
	{
		if(bookcode == tcode)
		{
			strcpy(tempText, author);
			break;
		}
	}
	file.close();
	return tempText;
} //End authornameOf()

// Returns the price of book with given code
float Book::bookpriceOf(int tcode)
{
	fstream file(BOOKFILE, 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;
} //End bookpriceOf()

// Returns the code of the book with given name
int Book::bookcodeOf(char t1code[33])
{
      fstream file(BOOKFILE, ios::in);
      file.seekg(0, ios::beg);
      int tcode = 0;
      while( file.read( (char*)this, sizeof(Book) ) )
      {
            if( !strcmp(name, t1code) )
            {
                  tcode = bookcode;
                  break;
            }
      }
      file.close();
      return tcode;
} //End bookcodeOf()

// Returns the total no. of records of books
int Book::recCount(void)
{
      fstream file(BOOKFILE, ios::in);
      file.seekg(0, ios::beg);
      int count = 0;
      while( file.read( (char*)this, sizeof(Book) ) )
            count++;
      file.close();
      return count;
} //End recCount()

// Deletes record of book with given code
void Book::deleteRec(int tcode)
{
      fstream file(BOOKFILE, ios::in);
      fstream temp(TEMPFILE, 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();

      ofstream fil2(BOOKFILE);
      temp.open(TEMPFILE, ios::in);
      temp.seekg(0, ios::beg);
      while( !temp.eof() )
      {
            temp.read( (char*)this, sizeof(Book) );
            if( temp.eof() )
                  break;
            fil2.write( (char*)this, sizeof(Book) );
      }
      fil2.close();
      temp.close();
} //End deleteRec()
```

```
// Adds new book to data file
void Book::addNewBook(int tcode, char tname[33], char tauthor[26],
                              float tprice, int tcopies, int tavail)
{
      fstream file(BOOKFILE, ios::app);
      bookcode = tcode;
      strcpy(name, tname);
      strcpy(author, tauthor);
      price = tprice;
      copies = tcopies;
      avail = tavail;
      file.write( (char*)this, sizeof(Book) );
      file.close();
} //End addNewBook()

// Updates the no. of copies of books
void Book::updateCopies(int tcode, int tcopies, int tavail)
{
      int recno;
      recno = recordnoOf(tcode);
      fstream file(BOOKFILE, ios::out | ios::in);
      copies = tcopies;
      avail = tavail;
      int location;
      location = (recno-1) * sizeof(Book);
      file.seekp(location);
      file.write( (char*)this, sizeof(Book) );
      file.close();
} //End updateCopies()

// Modifies the book record
void Book::modify(int tcode, char tname[33], char tauthor[26],
                        float tprice)
{
      int recno;
      recno = recordnoOf(tcode);
      fstream file(BOOKFILE, ios::out | ios::ate | ios::in);
      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();
} //End modify()

// Displays all the information about the book
void Book::display(int tcode)
{
      fstream file(BOOKFILE, 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      : "<<copies;
                  gotoxy(5,11); cout<<"Available   : "<<avail;
                  break;
            }
      }
      file.close();
} //End display()
```

```cpp
// Displays the list of all the books
void Book::displayList(void)
{
      clrscr();
      int row = 6, found = 0, flag = 0;
      char ch;

      gotoxy(34,2);                                 cout<<"LIST OF BOOKS";
      gotoxy(34,3);                                 cout<<"-------------";
      gotoxy(1,4);
      cout<<" CODE BOOK NAME                        AUTHOR                      "
            <<"PRICE  COPIES";
      gotoxy(1,5);
      cout<<"-------------------------------------------------------------------"
            <<"-------------";
      fstream file(BOOKFILE, ios::in);
      file.seekg(0, ios::beg);
      while( file.read( (char*)this, sizeof(Book) ) )
      {
            flag = 0;
            found=1;

            gotoxy(1,row);  cout<<setw(5)<<bookcode;
            gotoxy(7,row);  cout<<name;
            gotoxy(40,row); cout<<author;
            gotoxy(66,row); cout<<setw(7)<<price;
            gotoxy(77,row); cout<<setw(2)<<copies;

            textbackground(WHITE); textcolor(BLACK);
            gotoxy(40,row+1); cprintf("STATUS:");

            textcolor(BLACK);
            cprintf("%3d copies available",avail);

            textbackground(BLACK); textcolor(LIGHTGRAY);
            if(row == 22)
            {
                  flag = 1;
                  row = 6;

                  gotoxy(1,25);
                  cout<<"Press <ESC> to exit or any other key to continue...";
                  ch = _getch();
                  if(ch == 27)
                        break;

                  clrscr();
                  gotoxy(34,2);                         cout<<"LIST OF BOOKS";
                  gotoxy(34,3);                         cout<<"-------------";
                  gotoxy(1,4);
                  cout<<" CODE BOOK NAME                        AUTHOR              "
                        <<"        PRICE  COPIES";
                  gotoxy(1,5);
                  cout<<"-----------------------------------------------------------"
                        <<"---------------------";
            }
            else
                  row += 2;
      }
      if(!found)
      {
            gotoxy(5,10); cerr<<"\7Records not found";
      }
      if(!flag)
      {
            gotoxy(1,25); cout<<"Press any key to continue...";
            _getche();
        }
        file.close();
} //End displayList()


//////////////////////////////////////////////////////
// Function definitions pertaining to Member Class //
//////////////////////////////////////////////////////

// Returns 1 if member with given code is found
int Member::memberFound(int mcode)
{
        fstream file(MEMBERFILE, 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;
} //End memberFound()

// Returns the code of the book issued to member
int Member::issuedFor(int mcode)
{
        fstream file(MEMBERFILE, 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;
} //End issuedFor()

// Returns the amount of fine for given member code
int Member::fineFor(int mcode)
{
        Date today;
        fstream file(MEMBERFILE, ios::in);
        file.seekg(0, ios::beg);
        int days;
        int t_fine;
        while( file.read( (char*)this, sizeof(Member) ) )
        {
                if(memcode == mcode)
                {
                        days = today.diff(returnDate);
                        t_fine = days * FINERATE;
                        break;
                }
        }
        file.close();
        return t_fine;
} //End fineFor()
```

```
// Returns the code of the last member from the data file
int Member::lastcode(void)
{
      fstream file(MEMBERFILE, ios::in);
      file.seekg(0, ios::beg);
      int mcode = 0;
      while( file.read( (char*)this, sizeof(Member) ) )
            mcode = memcode;
      file.close();
      return mcode;
} //End lastcode()

// Returns the name of member with given code
char* Member::membername(int mcode)
{
      fstream file(MEMBERFILE, ios::in);
      file.seekg(0, ios::beg);
      delete tempText;
      char *tempText = new char[26];
      while( file.read( (char*)this, sizeof(Member) ) )
      {
            if(memcode == mcode)
            {
                  strcpy(tempText, name);
                  break;
            }
      }
      file.close();
      return tempText;
} //End membername()

// Returns the phone number of member with given code
char* Member::memberphone(int mcode)
{
      fstream file(MEMBERFILE, ios::in);
      file.seekg(0, ios::beg);
      delete tempText;
      char *tempText = new char[10];
      while( file.read( (char*)this, sizeof(Member) ) )
      {
            if(memcode == mcode)
            {
                  strcpy(tempText, phone);
                  break;
            }
      }
      file.close();
      return tempText;
} //End memberphone()

// Returns the address of member with given code
char* Member::memberaddress(int mcode)
{
      fstream file(MEMBERFILE, ios::in);
      file.seekg(0, ios::beg);
      delete tempText;
      char *tempText = new char[33];
      while( file.read( (char*)this, sizeof(Member) ) )
      {
            if(memcode == mcode)
            {
                  strcpy(tempText, address);
                  break;
            }
      }
      file.close();
      return tempText;
```

```
} //End memberaddress()

// Returns the serial no. of given record from the data file
int Member::recordnoOf(int mcode)
{
      fstream file(MEMBERFILE, 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;
} //End recordnoOf()

// Deletes the record of member with given code
void Member::deleteRec(int mcode)
{
      fstream file(MEMBERFILE, ios::in);
      fstream temp(TEMPFILE, 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();

      ofstream fil2(MEMBERFILE);
      temp.open(TEMPFILE, ios::in);
      temp.seekg(0, ios::beg);
      while( !temp.eof() )
      {
            temp.read( (char*)this, sizeof(Member) );
            if( temp.eof() )
                  break;
            fil2.write( (char*)this, sizeof(Member) );
      }
      fil2.close();
      temp.close();
} //End deleteRec()

// Updates the book information of member
void Member::updateBook(int mcode, int tcode, Date ret_date)
{
      fstream file(MEMBERFILE, ios::in);
      fstream temp(TEMPFILE, 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;
                  returnDate = ret_date;
                  temp.write( (char*)this, sizeof(Member) );
            }
            else
```

```
			temp.write( (char*)this, sizeof(Member) );
		}
		file.close();
		temp.close();

		ofstream fil2(MEMBERFILE);
		temp.open(TEMPFILE, ios::in);
		temp.seekg(0, ios::beg);
		while( !temp.eof() )
		{
			temp.read( (char*)this, sizeof(Member) );
			if( temp.eof() )
				break;
			fil2.write( (char*)this, sizeof(Member) );
		}
		fil2.close();
		temp.close();
} //End updateBook()

// Modifies the member record
void Member::modify(int mcode, char mname[26], char mphone[10],
                            char maddress[33])
{
	int recno;
	recno = recordnoOf(mcode);
	fstream file(MEMBERFILE, ios::out | ios::ate | ios::in);
	strcpy(name, mname);
	strcpy(address, maddress);
	strcpy(phone, mphone);
	int location;
	location = (recno-1) * sizeof(Member);
	file.seekp(location);
	file.write( (char*)this, sizeof(Member) );
	file.close();
} //End modify()

// Adds new member to data file
void Member::addMem(int mcode, int bcode, char mname[26], char maddress[33],
                            char mphone[10], Date ret_date)
{
	fstream file(MEMBERFILE, ios::app);
	memcode = mcode;
	bookcode = bcode;
	strcpy(name, mname);
	strcpy(address, maddress);
	strcpy(phone, mphone);
	returnDate = ret_date;
	file.write( (char*)this, sizeof(Member) );
	file.close();
} //End addMem

// Displays all information about the member
void Member::display(int mcode)
{
	fstream file(MEMBERFILE, 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(5,5);  cout<<"Member Name    : "<<name;
			gotoxy(5,6);  cout<<"Phone Number   : "<<phone;
			gotoxy(5,7);  cout<<"Address        : "<<address;
			break;
		}
	}
```

```
     file.close();
} //End display()

// Displays the list of all the members
void Member::displayList(void)
{
     clrscr();
     Book B;
     int row = 6, found = 0, flag = 0;
     char ch;
     gotoxy(32,2); cout<<"LIST OF MEMBERS";
     gotoxy(32,3); cout<<"---------------";
     gotoxy(1,4);
     cout<<"    CODE      BOOK CODE          NAME                              "
          <<"  PHONE      ";
     gotoxy(1,5);
     cout<<"-------------------------------------------------------------------"
          <<"-------------";
     fstream file(MEMBERFILE, ios::in);
     file.seekg(0, ios::beg);
     while( file.read( (char*)this, sizeof(Member) ) )
     {
          flag = 0;
          found=1;
          gotoxy(4,row); cout<<setw(5)<<memcode;
          gotoxy(17,row); cout<<setw(5)<<bookcode;
          gotoxy(34,row); cout<<name;
          gotoxy(66,row); cout<<setw(9)<<phone;
          textbackground(WHITE); textcolor(BLACK);
          gotoxy(7,row+1);
          if( bookcode == 0)
               cprintf("BOOK NAME: (Not Issued)");
          else
          {
               cout<<"BOOK NAME: "<<B.booknameOf(bookcode);
               gotoxy(42, row+1);
               cprintf("Date of return: ");
               textcolor(BLACK);
               returnDate.printDate();
          }
          textbackground(BLACK); textcolor(LIGHTGRAY);
          if(row == 22)
          {
               flag = 1;
               row = 6;
               gotoxy(1,25);
               cout<<"Press <ESC> to exit or any other key to continue...";
               ch = _getch();
               if(ch == 27)
                    break;
               clrscr();
               gotoxy(32,2); cout<<"LIST OF MEMBERS";
               gotoxy(32,3); cout<<"---------------";
               gotoxy(1,4);
               cout<<"    CODE      BOOK CODE          NAME                      "
                    <<"          PHONE      ";
               gotoxy(1,5);
               cout<<"-----------------------------------------------------------"
                    <<"---------------------";
          }
          else
               row += 2;
     }
     if(!found)
     {
          gotoxy(5,10); cerr<<"\7Records not found";
     }
     if(!flag)
     {
          gotoxy(1,25); cout<<"Press any key to continue...";
          _getche();
     }
     file.close();
} //End displayList()


//////////////////////////////////////////////////////
// Function definitions pertaining to Working Class //
//////////////////////////////////////////////////////

// To add a new book
void Working::addBook(void)
{
     if(!recCount())
     {
          addNewBook(0, "null", "null", 0.0, 0, 0);
          Book::deleteRec(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 = NULL;
          clrscr();
          gotoxy(29,3); cout<<"ADDITION OF THE BOOKS";
          gotoxy(29,4); cout<<"---------------------";
          gotoxy(71,1); cout<<"<0> = Exit";
          gotoxy(5,25); cout<<"Enter code no. of the book";
          gotoxy(5,5); cout<<"Code no. ";
          t = new char[10];
          gets(t);
          if ( strlen(t)>8 )
               exitLM();
          tc = atoi(t);
          tcode = tc;
          delete t;
          if (tcode == 0)
               return;
          if (bookFound(tcode))
          {
               found = 1;
               gotoxy(19,8); cout<<booknameOf(tcode);
               gotoxy(19,9); cout<<authornameOf(tcode);
               gotoxy(23,10); cout<<bookpriceOf(tcode);
               gotoxy(19,12); cout<<noOfCopiesOf(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) > 31)
            {
                  valid = 0;
                  gotoxy(5,25); clreol();
                  cerr<<"\7Enter correctly (Range: 1..31)";
                  _getch();
            }
      }
      valid = 0;
      while(!valid && !found)
      {
            valid = 1;
            gotoxy(5,25); clreol();
            cout<<"Enter the name of the author of the book";
            gotoxy(19,9); clreol();
            gets(tauthor);
            _strupr(tauthor);
            if(tauthor[0] == '0')
                  return;
            if(strlen(tauthor) < 1 || strlen(tauthor) > 24)
            {
                  valid = 0;
                  gotoxy(5,25); clreol();
                  cerr<<"\7Enter correctly (Range: 1..24)";
                  _getch();
            }
      }
      valid = 0;
      while(!valid && !found)
      {
            valid = 1;
            gotoxy(5,25); clreol();
            cout<<"Enter the price of the book";
            gotoxy(23,10); clreol();
            t = new char[10];
            gets(t);
            if ( strlen(t)>8 )
                  exitLM();
            t2 = (float)atof(t);
            tprice = t2;
            if(t[0] == '0')
            {
                  delete t;
                  return;
            }
            delete t;
            if(tprice < 1 || tprice > 9999)
            {
                  valid = 0;
                  gotoxy(5,25); clreol();
                  cerr<<"\7Enter correctly";
                  _getch();
            }
      }
      valid = 0;
      while(!valid && !found)
      {
            valid = 1;
            gotoxy(5,25); clreol();
            cout<<"Enter no. of copies of book to be added";
            gotoxy(19,12); clreol();
            t = new char[10];
            gets(t);
            if ( strlen(t)>8 )
                  exitLM();
            tc = atoi(t);
            tcopies = tc;
            if(t[0] == '0')
            {
                delete t;
                return;
            }
            delete t;
            if(tcopies < 1 || tcopies > 50)
            {
                valid = 0;
                gotoxy(5,25); clreol();
                cerr<<"\7Enter correctly";
                _getch();
            }
        }
        if(found)
        {
            gotoxy(5,25); clreol();
            cout<<"Press <SPACE> to change the no. of copies";
            char c=_getch();
            if(c != ' ')
                goto skipsaving;
            valid = 0;
            while(!valid)
            {
                valid = 1;
                gotoxy(5,25); clreol();
                cout<<"Enter no. of copies to be added, use minus (-) sign to "
                    <<"remove copies";
                gotoxy(5,13); cout<<"Add Copies  :";
                gotoxy(19,13); clreol();
                t = new char[10];
                gets(t);
                if ( strlen(t)>8 )
                    exitLM();
                tc = atoi(t);
                tcopies = tc;
                if(t[0] == '0')
                {
                    delete t;
                    return;
                }
                delete t;
                if(tcopies+noOfCopiesOf(tcode) < 1
                    || tcopies+noOfCopiesOf(tcode) > 50)
                {
                    valid = 0;
                    gotoxy(5,25); clreol();
                    cerr<<"\7Enter correctly";
                    _getch();
                }
            }
        }
        if(!found)
            tavail = tcopies;
        else
        {
            tavail = available(tcode) + tcopies;
            tcopies = noOfCopiesOf(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)
                    updateCopies(tcode, tcopies, tavail);
                else
                    addNewBook(tcode, tname, tauthor, tprice, tcopies, tavail);
            }
skipsaving:
            do
            {
                gotoxy(5,25); clreol();
                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');
} //End addBook()

// To add a new member
void Working::addMember(void)
{
        char ch;
        int mcode, bcode;
        char mname[26], mphone[10], maddress[33];
        mcode = lastcode();
        mcode++;
        do
        {
            int valid = 0;
            clrscr();
            gotoxy(28,3); cout<<"ADDITION OF THE MEMBERS";
            gotoxy(28,4); cout<<"-----------------------";
            gotoxy(71,1); 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) > 24)
                {
                    valid = 0;
                    gotoxy(5,25); clreol();
                    cerr<<"\7Enter correctly (Range: 1..24)";
                    _getch();
                }
            }while(!valid);
            do
            {
                valid = 1;
                gotoxy(5,25); clreol();
                cout<<"Enter Phone no. of the Member or Press <ENTER> for none";
                gotoxy(15,12); clreol();
                gets(mphone);
                if(mphone[0] == '0')
                    return;
                if((strlen(mphone) < 7 && strlen(mphone) > 0) || strlen(mphone) > 8)
                {
                    valid = 0;
                    gotoxy(5,25); clreol();
                    cerr<<"\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) > 31)
                {
                    valid = 0;
                    gotoxy(5,25); clreol();
                    cerr<<"\7Enter correctly (Range: 1..31)";
                    _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;
                addMem(mcode, bcode, mname, maddress, mphone, Date(0));
                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');
    } //End addMember()

    // To issue a book
    void Working::issueBook(void)
    {
        Book B;
        Member M;
        Date today;
        char t1code[33];
        char ch;
        int t2code = 0, tcode = 0, mcode = 0;
        int valid;
        do
```

```
{
        valid = 1;
        while(1)
        {
                clrscr();
                gotoxy(5,2); cout<<"Date : "; today.printDate();
                gotoxy(71,1); 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.displayList();
                else
                        break;
        }
        t2code = atoi(t1code);
        tcode = t2code;

        if( (tcode == 0 && !booknameFound( _strupr(t1code) ) )
                || (tcode != 0 && !bookFound(tcode)) )
        {
                valid = 0;
                gotoxy(5,10);
                cerr<<"\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);
        cerr<<"\7Sorry! Book ("<<booknameOf(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(5,2); cout<<"Date : "; today.printDate();
                gotoxy(71,1); cout<<"<0> = Exit";
                gotoxy(5,5); cout<<"Book Name: "<<booknameOf(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.displayList();
                else
                        break;
        }
        t2code = atoi(t1code);
        mcode = t2code;
        if(mcode == 0)
        {
            valid = 0;
            gotoxy(5,25);
            cerr<<"\7Enter Correctly";
            _getch();
        }
        if( !memberFound(mcode) && valid )
        {
            valid = 0;
            gotoxy(5,13);
            cerr<<"\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);
    if( issuedFor(mcode) )
    {
        gotoxy(5,13);
        cerr<<"\7Sorry, Member already has a book issued";
        gotoxy(5,14);
        cout<<"Press any key to continue";
        _getch();
        return;
    }
    int tcopies, tavail;
    tcopies = noOfCopiesOf(tcode);
    tavail = available(tcode) - 1;
    updateCopies(tcode, tcopies, tavail);
    today.extendDate(15);
    updateBook(mcode, tcode, today);
    gotoxy(5,13);
    cout<<"\7Book issued to "<<membername(mcode);
    gotoxy(5,15);
    cout<<"Date of Return: "; today.printDate();
    _getch();
} //End issueBook()

// To return a book
void Working::returnBook(void)
{
    Member M;
    char t1code[5];
    char ch;
    int t2code = 0, mcode = 0;
    int valid;
    Date today;
    do
    {
        valid = 1;
        while(1)
        {
            clrscr();
            gotoxy(5,2); cout<<"Date : "; today.printDate();
            gotoxy(71,1); cout<<"<0> = Exit";
            gotoxy(5,5); cout<<"Enter Code no. of the Member who wants to retu"
                <<"rn book";
            gotoxy(5,6); cout<<"                 or           ";
            gotoxy(5,7); cout<<"Press <ENTER> for help ";
            gets(t1code);
            if(t1code[0] == '0')
                return;
            if(strlen(t1code) == 0)
                M.displayList();
            else
                break;
        }
        t2code = atoi(t1code);
        mcode = t2code;
        if(mcode == 0)
        {
            valid = 0;
            gotoxy(5,25);
            cerr<<"\7Enter Correctly";
            _getch();
        }
        if( !memberFound(mcode) && valid )
        {
            valid = 0;
            gotoxy(5,13);
            cerr<<"\7Record not found";
            gotoxy(5,14);
            cout<<"Press <ESC> to exit or any other key to continue...";
            ch = _getch();
            if(ch == 27)
                return;
        }
        if( !issuedFor(mcode) && valid )
        {
            valid = 0;
            gotoxy(5,13);
            cerr<<"\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 = issuedFor(mcode);
    gotoxy(5,13);
    cout<<"Book Code :  "<<bcode;
    gotoxy(5,14);
    cout<<"Book Name :  "<<booknameOf(bcode);
    tcopies = noOfCopiesOf(bcode);
    tavail = available(bcode) + 1;
    int f;
    f = fineFor(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";
    }
    updateCopies(bcode, tcopies, tavail);
    updateBook(mcode, 0, Date(0));
    gotoxy(5,19);
    cout<<"\7Book has been returned";
    _getch();
} //End returnBook()

// To modify the record of a book
void Working::modifyBook(void)
{
    Book B;
    char t1code[5], tname[33], tauthor[26], *t1 = NULL;
    char ch;
```

```
int t2code = 0, tcode = 0;
float t2 = 0.0, tprice = 0.0;
int valid;
do
{
    valid = 1;
    while(1)
    {
        clrscr();
        gotoxy(71,1); 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.displayList();
        else
            break;
    }
    t2code = atoi(t1code);
    tcode = t2code;

    if( (tcode == 0 && !booknameFound( _strupr(t1code) ) )
        || (tcode != 0 && !bookFound(tcode)) )
    {
        valid = 0;
        gotoxy(5,10);
        cerr<<"\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(71,1); 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) > 31)
```

```
                {
                        valid = 0;
                        gotoxy(5,25); clreol();
                        cerr<<"\7Enter correctly (Range: 1..31)";
                        _getch();
                }
        }while(!valid);
        if(strlen(tname) == 0)
                strcpy(tname, booknameOf(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) > 24)
                {
                        valid = 0;
                        gotoxy(5,25); clreol();
                        cerr<<"\7Enter correctly (Range: 1..24)";
                        _getch();
                }
        }while(!valid);
        if(strlen(tauthor) == 0)
                strcpy(tauthor, authornameOf(tcode));
        do
        {
                delete t1;
                t1 = new char[6];
                valid = 1;
                gotoxy(5,25); clreol();
                cout<<"Enter price or <ENTER> for no change";
                gotoxy(23,18); clreol();
                gets(t1);
                t2 = (float)atof(t1);
                tprice = t2;
                if(t1[0] == '0')
                {
                        delete t1;
                        return;
                }
                if(strlen(t1) == 0)
                        break;
                if(tprice < 1 || tprice > 9999)
                {
                        valid = 0;
                        gotoxy(5,25); clreol();
                        cerr<<"\7Enter correctly";
                        _getch();
                }
        }while(!valid);
        if(strlen(t1) == 0)
                tprice = bookpriceOf(tcode);
        delete t1;
        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();
} //End modifyBook()

// To modify the record of a member
void Working::modifyMember(void)
{
        Member M;
        char m1code[10], mname[26], mphone[10], maddress[33];
        char ch;
        int m2code = 0, mcode = 0;
        int valid;
        do
        {
                valid = 1;
                while(1)
                {
                        clrscr();
                        gotoxy(71,1); cout<<"<0> = Exit";
                        gotoxy(5,7); cout<<"Enter Code no. of the Member to be modified";
                        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.displayList();
                        else
                                break;
                }
                if(mcode == 0)
                {
                        valid = 0;
                        gotoxy(5,25); cerr<<"\7Enter Correctly";
                        _getch();
                }
                if(valid && !memberFound(mcode))
                {
                        valid = 0;
                        gotoxy(5,13);
                        cerr<<"\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(71,1); 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 member or <ENTER> for no change";
     gotoxy(21,13); clreol();
     gets(mname);
     _strupr(mname);
     if(mname[0] == '0')
          return;
     if(strlen(mname) > 24)
     {
          valid = 0;
          gotoxy(5,25); clreol();
          cerr<<"\7Enter correctly (Range: 1..24)";
          _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(21,14); clreol();
     gets(mphone);
     if(mphone[0] == '0')
          return;
     if((strlen(mphone) < 7 && strlen(mphone) > 0) || strlen(mphone) > 8)
     {
          valid = 0;
          gotoxy(5,25); clreol();
          cerr<<"\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 change";
     gotoxy(21,15); clreol();
     gets(maddress);
     _strupr(maddress);
     if(maddress[0] == '0')
          return;
     if(strlen(maddress) > 31)
     {
          valid = 0;
          gotoxy(5,25); clreol();
          cerr<<"\7Enter correctly (Range: 1..31)";
          _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();
} //End modifyMember()

// To delete the record of a book
void Working::deleteBook(void)
{
      Book B;
      char t1code[5];
      char ch;
      int t2code = 0, tcode = 0;
      int valid;
      do
      {
            valid = 1;
            while(1)
            {
                  clrscr();
                  gotoxy(71,1); 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.displayList();
                  else
                        break;
            }
            t2code = atoi(t1code);
            tcode = t2code;

            if( (tcode == 0 && !booknameFound( _strupr(t1code) ) )
                  || (tcode != 0 && !bookFound(tcode)) )
            {
                  valid = 0;
                  gotoxy(5,10);
                  cerr<<"\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(71,1); 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 = noOfCopiesOf(tcode);
        if(tavail != tcopies)
        {
            gotoxy(5,15);
            cerr<<"\7Record cannot be deleted. This book is issued.";
            _getch();
            return;
        }
        Book::deleteRec(tcode);
        gotoxy(5,23); cout<<"\7Record Deleted";
        _getch();
} //End deleteBook()

// To delete the record of a member
void Working::deleteMember(void)
{
        Member M;
        char m1code[5];
        char ch;
        int m2code = 0, mcode = 0;
        int valid;
        do
        {
            valid = 1;
            while(1)
            {
                clrscr();
                gotoxy(71,1); 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.displayList();
                else
                    break;
            }
            if(mcode == 0)
            {
                valid = 0;
                gotoxy(5,25); cerr<<"\7Enter Correctly";
                _getch();
            }
            if(valid && !memberFound(mcode))
            {
                valid = 0;
                gotoxy(5,13);
                cerr<<"\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(71,1); 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(issuedFor(mcode))
     {
          gotoxy(5,15);
          cerr<<"\7Record cannot be deleted. Member has a book.";
          _getch();
          return;
     }
     Member::deleteRec(mcode);
     gotoxy(5,23); cout<<"\7Record Deleted";
     _getch();
} //End deleteMember()

//////////////////////////////////////////
//                                      //
//   THE END                            //
//                                      //
//////////////////////////////////////////
```

LibManage

SOURCE CODE

LibManage.h

```
////////////////////////////////////////
//                                    //
// LibManage                          //
// Library Management Sytem           //
//                                    //
// Copyright (c) 2007 Kartik Singhal. //
// kartiksinghal@gmail.com            //
// All rights reserved.               //
//                                    //
////////////////////////////////////////


#define _CRT_SECURE_NO_WARNINGS //to eliminate deprecation warnings


// Included header files
#include <cstdlib>
#include <iostream>
#include <fstream>
#include <string>
#include <cstdio>
#include <cctype>
#include <iomanip>
#include "conio2.h"
#include "date.h"

using namespace std;

// To exit the program incase memory error occurs
void exitLM(void)
{
	clrscr();
	gotoxy(28,13);
	cerr<<"Memory error! Exiting...";
	_getch();
	exit(0);
} //End exitLM()

//////////////////////////////////////
//          The Book Class          //
//////////////////////////////////////
class Book                  //76 bytes
{
public:
	void displayList(void);
	char* booknameOf(int);

protected:
	void addNewBook(int, char[], char[], float, int, int);
	void updateCopies(int, int, int);
	void modify(int, char[], char[], float);
	int bookFound(int);
	int booknameFound(char[]);
	int recordnoOf(int);
	int available(int);
	char* authornameOf(int);
```

```
      float bookpriceOf(int);
      int noOfCopiesOf(int);
      int bookcodeOf(char[]);
      void display(int);
      int recCount(void);
      void deleteRec(int);

private:
      int bookcode, copies;
      char name[33], author[26];
      float price;
      int avail;
}; //Book class ends


////////////////////////////////////////
//          The Member Class          //
////////////////////////////////////////
class Member                 //108 bytes
{
public:
      void displayList(void);

protected:
      void addMem(int, int, char[], char[], char[], Date);
      void modify(int, char[], char[], char[]);
      int memberFound(int);
      void updateBook(int, int, Date);
      char* membername(int);
      char* memberphone(int);
      char* memberaddress(int);
      int recordnoOf(int);
      int lastcode(void);
      int issuedFor(int);
      int fineFor(int);
      void display(int);
      void deleteRec(int);

private:
      int memcode, bookcode;
      char name[26], phone[10], address[33];
      Date returnDate;
}; //Member class ends


//////////////////////////////////////
//          The Menu Class          //
//////////////////////////////////////
class Menu
{
public:
      void displayMainMenu(void);
      void displayIntroduction(void);

private:
      void displayEditMenu(void);
      void displayEditBookMenu(void);
      void displayEditMemberMenu(void);
}; //Menu class ends


//////////////////////////////////////////
//           The Working Class          //
//////////////////////////////////////////
class Working : public Book, public Member
{
public:
     void issueBook(void);
     void returnBook(void);
     void addBook(void);
     void addMember(void);
     void modifyBook(void);
     void modifyMember(void);
     void deleteBook(void);
     void deleteMember(void);
}; //Working class ends

//////////////////////////////////////////
//                                      //
//   THE END                            //
//                                      //
//////////////////////////////////////////
```

LibManage

SOURCE CODE

date.h

```
////////////////////////////////////////
//                                    //
// LibManage                          //
// Library Management Sytem           //
//                                    //
// Copyright (c) 2007 Kartik Singhal. //
// kartiksinghal@gmail.com            //
// All rights reserved.               //
//                                    //
////////////////////////////////////////


#include <ctime>
#include <iostream>
#include <string>

using namespace std;



//////////////////////////////////////
//          The Date Class          //
//////////////////////////////////////
class Date                  //28 bytes
{
private:
     int day, month, year;
     char nameOfMonth[15];
     void giveNameOfMonth();

public:
     Date(void);
     Date(int);
     void printDate(void);
     void extendDate(int);
     int diff(Date);

     int returnDay(void)
     {
          return day;
     }

     int returnMonth(void)
     {
          return month;
     }

     int returnYear(void)
     {
          return year;
     }
}; //Date class ends
```

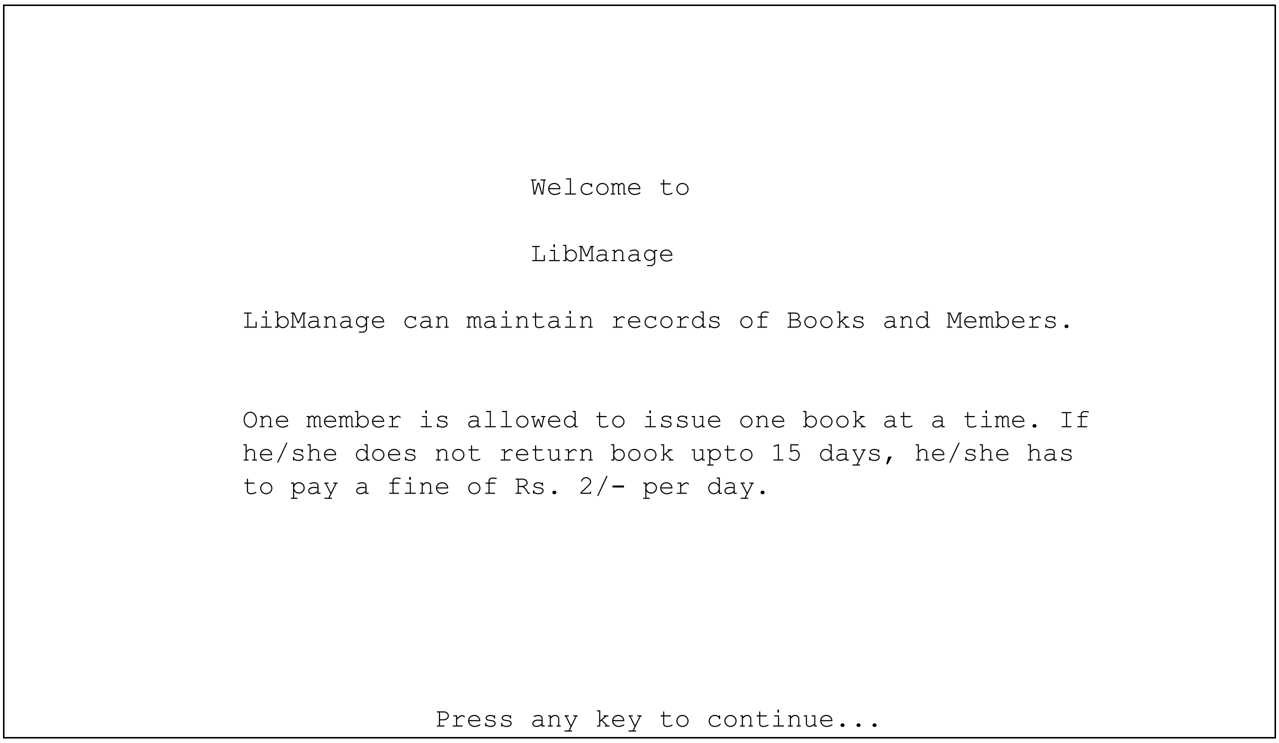
```
// Constructor to initialize date to current system time
Date::Date(void)
{

      ///////////////////////////////////////////////////////////////////////////
      struct tm *ptr;                     //                                  //
      time_t sec;                         //                                  //
      time(&sec);                         //Thanks to http://java2s.com for  //
      ptr = localtime(&sec);              //                                  //
                                          //particularly this piece of code  //
      month = (int) ptr->tm_mon + 1;      //                                  //
      day   = (int) ptr->tm_mday;         //to generate current system time. //
      year  = (int) ptr->tm_year + 1900;  //                                  //
      ///////////////////////////////////////////////////////////////////////////

      giveNameOfMonth();
} //End Date(void)

// Constructor to initialize date to zeroes
Date::Date(int d)
{
      day = month = year = 0;
} //End Date(int)

// Function to assign name of the month for particular month
void Date::giveNameOfMonth()
{
      switch (month)
      {
      case 1:
            strcpy(nameOfMonth,"January");
            break;
      case 2:
            strcpy(nameOfMonth,"February");
            break;
      case 3:
            strcpy(nameOfMonth,"March");
            break;
      case 4:
            strcpy(nameOfMonth,"April");
            break;
      case 5:
            strcpy(nameOfMonth,"May");
            break;
      case 6:
            strcpy(nameOfMonth,"June");
            break;
      case 7:
            strcpy(nameOfMonth,"July");
            break;
      case 8:
            strcpy(nameOfMonth,"August");
            break;
      case 9:
            strcpy(nameOfMonth,"September");
            break;
      case 10:
            strcpy(nameOfMonth,"October");
            break;
      case 11:
            strcpy(nameOfMonth,"November");
            break;
      case 12:
            strcpy(nameOfMonth,"December");
            break;
      }
} //End giveNameOfMonth()
```

```
// To print date in a specialized manner
void Date::printDate(void)
{
      cout << nameOfMonth << " " << day << ", " <<year;
} //End printDate()

// To extend current date by given no. of days
void Date::extendDate(int days)
{
      static int monthNum[] = {31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
      for(int i=1; i<=days; i++)
      {
            day++;
            if( (day > monthNum[month-1]) || (year%4 != 0 && month==2 && day>28) )
            {
                  day = 1;
                  month++;
            }
            if(month > 12)
            {
                  month = 1;
                  year++;
            }
      }

      giveNameOfMonth();

} //End extendDate()

// Returns the no. of days between the provided date and current date
int Date::diff(Date ret)
{
      int d = ret.returnDay(), m = ret.returnMonth(), y = ret.returnYear();
      int days=0;
      if( (year < y) || (year==y && month<m)
            || (year==y && month==m && day<d) )
            return days;
      static int month_days[] = {31, 29, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
      while( d != (int)day || m != (int)month || y != (int)year )
      {
            days++;
            d++;
            if( (d > month_days[m-1])
                  || (y%4 != 0 && m==2 && d>28) )
            {
                  d = 1;
                  m++;
            }
            if(m > 12)
            {
                  m = 1;
                  y++;
            }
      }
      return days;
} //End diff()

//////////////////////////////////////////
//                                      //
//   THE END                            //
//                                      //
//////////////////////////////////////////
```

LibManage

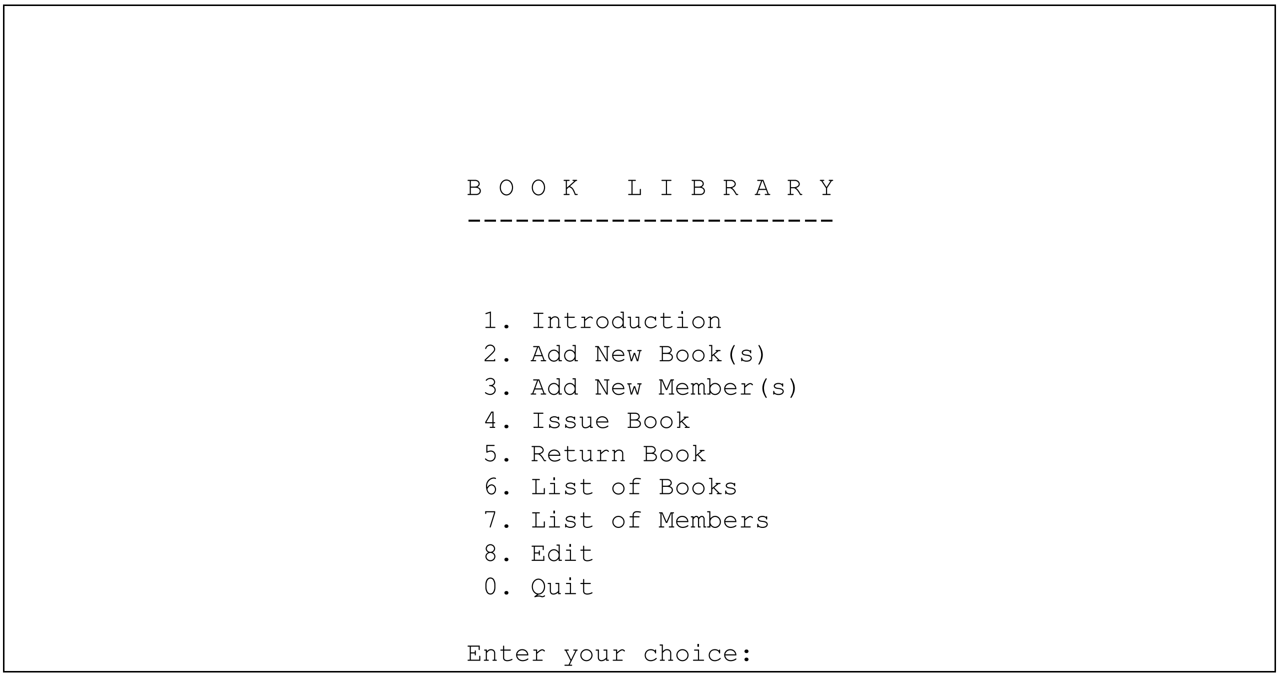
OUTPUT

INTRODUCTION



```
                     Welcome to

                     LibManage

   LibManage can maintain records of Books and Members.


   One member is allowed to issue one book at a time. If
   he/she does not return book upto 15 days, he/she has
   to pay a fine of Rs. 2/- per day.






                Press any key to continue...
```

MAIN MENU



```
              B O O K   L I B R A R Y
              -----------------------


               1. Introduction
               2. Add New Book(s)
               3. Add New Member(s)
               4. Issue Book
               5. Return Book
               6. List of Books
               7. List of Members
               8. Edit
               0. Quit

              Enter your choice:
```

LibManage

OUTPUT

BOOK ISSUE

```
                                                                  <0> = Exit
    Date : January 6, 2008


    Book Name: PHYSICS MCQ

    Enter Code no. of the Member
            or
    Press <ENTER> for help 5



    Book issued to RAHUL MAHESHWARI

    Date of Return: January 21, 2008
```

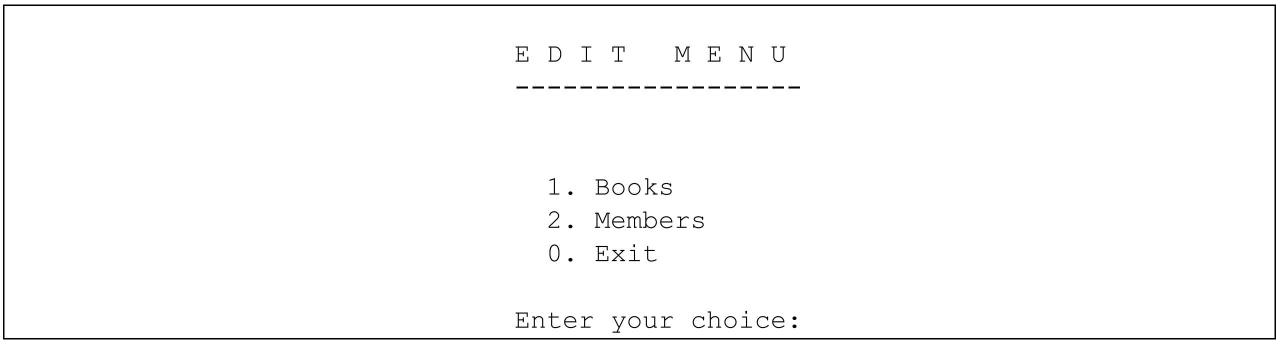
BOOK RETURN

```
                                                                  <0> = Exit
    Date : January 23, 2008


    Enter Code no. of the Member who wants to return book
                     or
    Press <ENTER> for help 6





    Book Code : 4
    Book Name : IIT CHEMISTRY

    You have to pay a fine of Rs. 4
    Please do not delay the Return of Book again

    Book has been returned
```

LibManage

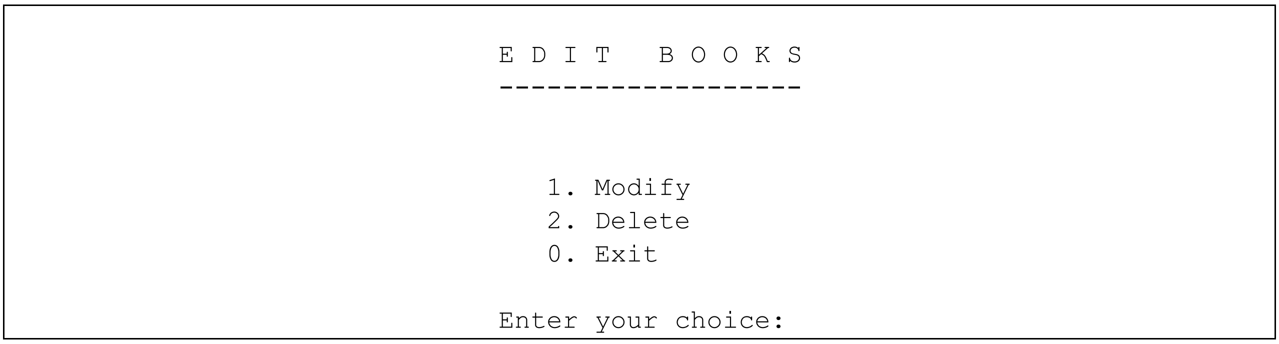
OUTPUT

EDIT MENU



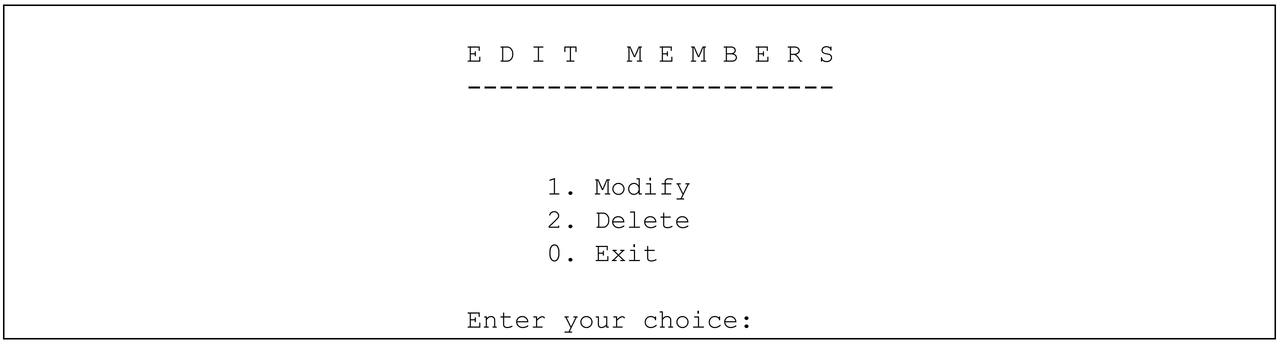
```
                                E D I T   M E N U
                                ------------------


                                  1. Books
                                  2. Members
                                  0. Exit

                                Enter your choice:
```

EDIT BOOK MENU



```
                               E D I T   B O O K S
                               -------------------


                                  1. Modify
                                  2. Delete
                                  0. Exit

                               Enter your choice:
```

EDIT MEMBER MENU



```
                             E D I T   M E M B E R S
                             -----------------------


                                  1. Modify
                                  2. Delete
                                  0. Exit

                             Enter your choice:
```

LibManage

OUTPUT

MODIFY BOOK RECORD

```
                                                                    <0> = Exit

    Book Code   : 8

    Book Name   : PRADEEP'S NEW COURSE CHEISTRY
    Author Name : PRADEEP BANDHU
    Price       : Rs. 500.00
    Copies      : 1
    Available   : 1

    Do you want to modify this record (y/n) : y


    Book Name   : Pradeep's Fundamental Physics
    Author Name :
    Price       : Rs.

    Do you want to save changes (y/n): y


    Record Modified
```

DELETE BOOK RECORD

```
                                                                    <0> = Exit

    Book Code   : 11

    Book Name   : PRADEEP'S NEW CHEMISTRY
    Author Name : PRADEEP BROTHERS
    Price       : Rs. 495.00
    Copies      : 1
    Available   : 1

    Do you want to delete this record (y/n) : y







    Record Deleted
```

LibManage

OUTPUT

MODIFY MEMBER RECORD

```
                                                                   <0> = Exit


   Member Code    : 1

   Member Name    : KARITK SINGHAL
   Phone Number   : 3260721
   Address        : HARDEV SAJAI GHAZIABAD


   Do you want to modify this record (y/n) : y


   Name           : kartik singhal
   Phone          :
   Address        : hardev sahai ghaziabad


   Do you want to save changes (y/n): y




   Record Modified
```

DELETE MEMBER RECORD

```
                                                                   <0> = Exit


   Member Code    : 9

   Member Name    : HITESH B. TRIVEDI
   Phone Number   : 2750701
   Address        : ANAND VIHAR


   Do you want to delete this record (y/n) : y





   Record Deleted
```

LibManage

BIBLIOGRAPHY

- *Practicals & Projects in* C++, by Sumita Arora
- C/C++ *Reference*, http://cppreference.com/
- *CONIO Documentation 2.0*, http://conio.sf.net/
- *The GNU C Library*
- *Microsoft Visual Studio Documentation*
- *Date class: year, month and day*, http://java2s.com
- C/C++ *Programming Style Guidelines*, by Fred Richards
- *Practical* C++ *Programming*, Steve Oualline