LIBRARY MANAGEMENT SYSTEM

**Submitted in partial fulfilment of the requirement**

**For the practical in Computer Science of**

**Class XII - Science**

**Of**

**Central Board of Secondary Education, New Delhi**

**By: Project Guide:**

***YASH SRIVASTAVA Ms. Anusha Panicker***

****

**2016 – 2017**

**Senior Secondary Section**

**LAXMI INTERNATIONAL SCHOOL**

**P.B. No. 15, SARIGAM P.O., VALSAD – 396 155**

**GUJRAT,INDIA**

**ACKNOWLEDGEMENT**

We whole heartedly wish to thank Mr.Shaji T Mathew, for his immense help in carrying out this project work.

Our special thanks to our subject teacher Ms .Anusha Panicker for her valuable guidance in our project work.

We would also like to thank all the staff members of the Senior Secondary Wing, Laxmi International School, Sarigam for their effort in building our careers.

Without the support and encouragement of our parents, relatives and friends this project work would have never become a reality.

-YASH SRIVASTAVA

- CLASS-XII A

INDEX :-

|  |  |  |  |
| --- | --- | --- | --- |
| S NO. | TOPIC | PAGE NUMBER | SIGN |
| 1 | Overview of C++ | 4 |
| 2 | About the Project | 5 |
| 3 | System Requirements | 6-7 |
| 4 | Header Files Used | 8-9 |
| 5 | Source Code | 10-28 |
| 6 | Output Window | 29-36 |
| 7 | Bibliography | 37 |

Overview of C++

* C++ was designed at AT&T Bell Labs by Bjarne Stroustrup in the early 80’s – nearly 30 years ago! – The original cfront translated C++ into C for portability
* However, this was difficult to debug and potentially inefficient – Many native host machine compilers now exist
* e.g., C++ Builder, clang, Comeau C/C++, GCC, Intel C++Compiler, Microsoft Visual C++, Sun Studio, etc.

C++ is a mostly upwardly compatible extension of C that provides:

1. Stronger typechecking

2. Support for data abstraction

3. Support for object-oriented programming

4. Support for generic programming

About The Project

Library Manager 1.0 is a basic C++ based library managing system which covers all the basic needs of a library regarding its members and is also very easy to access and is best for school libraries.

It uses the basic features of C++ like Classes and Objects and Data File Handling to provide an efficient managing system.

It allows the user to register new members, modify existing members, maintain the details of the book , display specific book or member and maintains the name of book with the user. It even calculates the fine to be collected from a member if he submits a book late.

System Requirements:-

*Windows\**

|  |  |
| --- | --- |
| **Hardware** | |
| **Recommended** | * Intel® Core™ 2 Duo processor or Intel® Xeon® processor or higher |
| **Minimum Requirements** | * One of the following processors   + Intel® Pentium® 4 processor family and higher   + Intel(R) Xeon Phi(TM) coprocessor   + Non Intel® processors compatible with the above processors |
| **Software** | |
| **Minimum Requirements** | * One of the following operating systems:   + Microsoft\* Windows\* 8, 8.1   + Microsoft\* Windows\* 7 SP1   + Microsoft\* Windows\* Server 2012   + Microsoft\* Windows\* Server 2008 SP2 (IA-32 only)   + Microsoft\* Windows\* Server 2008 R2 SP1   + Microsoft\* Windows\* HPC Server 2008 * One of the following compilers:   + Intel(R) C++ Compiler 13.1 (Intel(R) Parallel Studio XE 2013 SP1) and higher   + Microsoft\* Visual C++ 10.0 (Microsoft\* Visual Studio\* 2010, Windows\* OS only) and higher |
| **Recommended** | * Intel(R) Parallel Studio XE 2013 SP1 * Intel(R) Parallel Studio XE 2015 Professional Edition |

*OS X\**

|  |  |
| --- | --- |
| **Hardware** | |
| **Recommended** | * Intel(R) Core(TM) 2 Duo processor or higher |
| **Software** | |
| **Minimum Requirements** | * One of the following operating systems:   + OS X\* 10.9 and higher * One of the following compilers:   + Xcode\* 5.0 and higher and command line tools   + Intel(R) C++ Compiler 13.1 (Intel(R) Parallel Studio XE 2013 SP1) and higher |
| **Recommended** | * Intel(R) Parallel Studio XE 2013 SP1 * Intel(R) Parallel Studio XE 2015 Professional Edition |

HEADER FILES USED :-

* **#include<fstream.h>**

 For defining several iostreams template classes that manipulate exteral files.   
**fstream** is a standard C++ library that handles reading from and writing to files either in text or in binary formats. It is an object oriented alternative to C's FILE from the C standard library. fstream is the result of a multiple inheritance with ifstream and ofstream, which both inherit from ios.

* **#include<conio.h>**

**conio**.**h** header used in c programming contains functions for console input/output. Some of the most commonly used functions of conio.h are clrscr, getch, getche, kbhit etc. Functions of conio.h can be used to clear screen, change color of text and background, move text, check if a key is pressed or not and many more. conio.h file is provided by Borland turbo c compiler and GCC compiler doesn't support it.

* **#include<stdio.h>**

Input and Output operations can also be performed in C++ using the **C** **Standard** **Inputand** **OutputLibrary** (**cstdio**, known as stdio.h in the C language). This library uses what are called *streams* to operate with physical devices such as keyboards, printers, terminals or with any other type of files supported by the system. Streams are an abstraction tointeract with these in an uniform way; All streams have similar properties independently of the individual characteristics of the physical media they are associated with.  
Streams are handled in the cstdio library as pointers to [FILE](http://www.cplusplus.com/FILE) objects. A pointer to a [FILE](http://www.cplusplus.com/FILE) object uniquely identifies a stream, and is used as a parameter in the operations involving that stream.

* **#include<string.h>**

This library enables you to manipulate C strings that end in the char '\0', the null char. Unless noted otherwise, these functions return a pointer to the resulting string in addition to modifying an appropriate argument. The argument ch is a character, n is an integer, and the other arguments are strings, which usually means they are names of a char array, but can be string constants in some cases. For example, strcmp("Hello", "Goodbye");

* **#include<process.h>**

**process.h** is a C [headerfile](https://en.wikipedia.org/wiki/Header_file) which contains function declarations and macros used in working with threads and processes. Most C compilers thattarget [DOS](https://en.wikipedia.org/wiki/DOS), [Windows 3.1x](https://en.wikipedia.org/wiki/Windows_3.1x), [Win32](https://en.wikipedia.org/wiki/Win32), [OS/2](https://en.wikipedia.org/wiki/OS/2), Novell NetWare or [DOS extenders](https://en.wikipedia.org/wiki/DOS_extender) supply this header and the library functions in their C library. Neither the header file nor most of the functions are defined by either the [ANSI/ISO C](https://en.wikipedia.org/wiki/ANSI_C) standard or by [POSIX](https://en.wikipedia.org/wiki/POSIX).

SOURCECODE:

#include<iostream.h>

#include<string.h>

#include<conio.h>

#include<fstream.h>

#include<stdio.h>

#include<process.h>

class book //class definition

{ char type[12];

int id;

char name[50];

char author[40];

public:

void bookentry(char);

void showbook();

int returnid();

};

int book::returnid()

{ return id; }

void book::bookentry(char ch)

{

cout<<"\nEnter book ID:\n";

cin>>id;

cout<<"\nEnter book name:\n";

gets(name);

cout<<"Enter the author's name:\n";

gets(author);

cout<<"Enter the book category (History, Science,";

cout<<" Fiction, Non-Fiction, Subordicals):\n";

gets(type);

if(ch=='n')

cout<<"\nNew Book Created.";

}

void book::showbook()

{

cout<<"\nBook ID: "<<id;

cout<<"\nCategory: ";

puts(type);

cout<<"Book Name: ";

puts(name);

cout<<"Author Name: ";

puts(author);

}

class member

{

int memid;

char name[20];

int bookid;

int x;

public:

void newmem(char);

void dispmem();

int retmemid()

{ return memid; }

int retbookid()

{ return bookid; }

int returnx()

{ return x; }

void changex(int no)

{ x=no; }

void newbook(int b)

{ bookid=b; }

};

void member::newmem(char ch)

{

clrscr();

cout<<"\nEnter member ID.: ";

cin>>memid;

cout<<"\nEnter name of the member: ";

gets(name);

if(ch=='n')

{ x=0;

bookid=0;

cout<<"\nNew Member Created.";

}

}

void member::dispmem()

{

cout<<"\nMember ID.: "<<memid;

cout<<"\nMember name: ";

puts(name);

if(x==1)

cout<<"Book no. issued: "<<bookid<<"\n";

else cout<<"No book issued.";

}

fstream f1,f2;

book B;

member M;

void bookinput()

{

char ch;

f1.open("book.dat",ios::out|ios::app);

do

{

clrscr();

B.bookentry('n');

f1.write((char\*)&B,sizeof(book));

cout<<"\n\nDo you want to add more books?(y/n)";

cin>>ch;

}while(ch=='y'||ch=='Y');

f1.close();

}

void memberinput()

{

char ch;

f1.open("member.dat",ios::out|ios::app);

do

{

M.newmem('n');

f1.write((char\*)&M,sizeof(member));

cout<<"\n\nDo you want to add more members?(y/n)";

cin>>ch;

}while(ch=='y'||ch=='Y');

f1.close();

}

void readbook(int n)

{

cout<<"\nBOOK DETAILS\n";

int flag=0;

f1.open("book.dat",ios::in);

while(f1.read((char\*)&B,sizeof(book)))

{ if(B.returnid()==n)

{ B.showbook();

flag=1;

}

}

f1.close();

if(flag==0)

cout<<"\n\nBook does not exist.";

getch( );

}

void readmem(int n)

{

cout<<"\nMEMBER DETAILS\n";

int flag=0;

f1.open("member.dat",ios::in);

while(f1.read((char\*)&M,sizeof(member)))

{ if(M.retmemid()==n)

{ M.dispmem();

flag=1;

}

}

f1.close();

if(flag==0)

cout<<"\n\nMember does not exist.";

getch( );

}

void modifybook()

{

int n;

int flag=0;

cout<<"\n\nMODIFY BOOK REOCORD ";

cout<<"\n\nEnter book ID.: ";

cin>>n;

f1.open("book.dat",ios::in|ios::out);

while(f1.read((char\*)&B,sizeof(book)) && flag==0)

{ if(B.returnid()==n)

{ B.showbook();

cout<<"\nEnter the new details: "<<endl;

B.bookentry('m');

int pos=-1\*sizeof(B);

f1.seekp(pos,ios::cur);

f1.write((char\*)&B,sizeof(book));

cout<<"\nBook Modified.";

flag=1;

}

}

f1.close();

if(flag==0)

cout<<"\n\n Book Not Found ";

getch();

}

void modifymem()

{

int n;

int flag=0;

cout<<"\nMODIFY STUDENT RECORD ";

cout<<"\n\nEnter member ID.: ";

cin>>n;

f1.open("member.dat",ios::in|ios::out);

while(f1.read((char\*)&M,sizeof(member)) && flag==0)

{ if(M.retmemid()==n)

{ M.dispmem();

cout<<"\nEnter the new details: ";

M.newmem('m');

int pos=-1\*sizeof(M);

f1.seekp(pos,ios::cur);

f1.write((char\*)&M,sizeof(member));

cout<<"\n Member Modified.";

flag=1;

}

}

f1.close();

if(flag==0)

cout<<"\n\n Member not found. ";

getch();

}

void delmem()

{

int n;

int flag=0;

cout<<"\n\n DELETE STUDENT";

cout<<"\n\nEnter member ID.: ";

cin>>n;

f1.open("member.dat",ios::in|ios::out);

f2.open("Temp.dat",ios::out);

f1.seekg(0,ios::beg);

while(f1.read((char\*)&M,sizeof(member)))

{ if(M.retmemid()!=n)

f2.write((char\*)&M,sizeof(member));

else

flag=1;

}

f2.close();

f1.close();

remove("member.dat");

rename("Temp.dat","member.dat");

if(flag==1)

cout<<"\n\nMember Deleted.";

else

cout<<"\n\nMember not found.";

getch();

}

void delbook()

{

int n;

cout<<"\n\n DELETE BOOK ";

cout<<"\nEnter book ID. of the book you want to delete: ";

cin>>n;

f1.open("book.dat",ios::in|ios::out);

f2.open("Temp.dat",ios::out);

f1.seekg(0,ios::beg);

while(f1.read((char\*)&B,sizeof(book)))

{ if(B.returnid()!=n)

{ f2.write((char\*)&B,sizeof(book));

}

}

f1.close();

f2.close();

remove("book.dat");

rename("Temp.dat","book.dat");

cout<<"\n\n Record Deleted.";

getch();

}

// function to issue book

void bookissue()

{

int m, n;

int found=0,flag=0;

cout<<"\n\nBOOK ISSUE.";

cout<<"\n\nEnter member ID.: ";

cin>>m;

f1.open("member.dat",ios::in|ios::out);

f2.open("book.dat",ios::in|ios::out);

while(f1.read((char\*)&M,sizeof(member)) && found==0)

{ if(M.retmemid()==m)

{ found=1;

M.dispmem();

if(M.returnx()==0)

{ cout<<"\nEnter the book no.: ";

cin>>n;

while(f2.read((char\*)&B,sizeof(book))&& flag==0)

{ if(B.returnid()==n)

{ B.showbook();

flag=1;

M.changex(1);

M.newbook(B.returnid());

int pos=-1\*sizeof(M);

f1.seekp(pos,ios::cur);

f1.write((char\*)&M,sizeof(member));

cout<<"\nBook issued successfully.";

cout<<"\nBook should be returned within 10 days.";

cout<<"\nOtherwise the fine is Rs.5 per day.";

}

}

if(flag==0)

cout<<"Book does not exist.";

}

else

cout<<"The last book has not been returned.";

}

}

if(found==0)

cout<<"Member record does not exist.";

getch();

f1.close();

f2.close();

}

void bookdeposit()

{

int m,n;

int found=0,flag=0,day,fine;

cout<<"\n\n BOOK DEPOSIT";

cout<<"\n\nEnter member ID.: ";

cin>>m;

f1.open("member.dat",ios::in|ios::out);

f2.open("book.dat",ios::in|ios::out);

while(f1.read((char\*)&M,sizeof(member)) && found==0)

{ if(M.retmemid()==m)

{ found=1;

M.dispmem();

if(M.returnx()==1)

{ while(f2.read((char\*)&B,sizeof(book)) && flag==0)

{ if(B.returnid()==M.retbookid())

{ B.showbook();

flag=1;

cout<<"\n\nNo. of days taken to return book:";

cin>>day;

if(day>10)

{ fine=(day-10)\*5;

cout<<"\nFine to be deposited="

<<" Rs."<<fine;

}

M.changex(0);

int pos=-1\*sizeof(M);

f1.seekp(pos,ios::cur);

f1.write((char\*)&M,sizeof(member));

cout<<"\nBook deposited"

<<" successfully";

}

}

if(flag==0)

cout<<"\nBook does not exist";

}

else

cout<<"No book is issued!!";

}

}

if(found==0)

cout<<"Member record does not exist.";

getch();

f1.close();

f2.close();

}

int adminmenu()

{ clrscr();

int ch;

cout<<"\n\nADMINISTRATOR MENU";

cout<<"\n\n1.CREATE MEMBER RECORD";

cout<<"\n\n2.DISPLAY SPECIFIC MEMBER RECORD ";

cout<<"\n\n3.MODIFY MEMBER RECORD";

cout<<"\n\n4.DELETE MEMBER RECORD";

cout<<"\n\n5.CREATE BOOK ";

cout<<"\n\n6.DISPLAY SPECIFIC BOOK ";

cout<<"\n\n7.MODIFY BOOK ";

cout<<"\n\n8.DELETE BOOK ";

cout<<"\n\n9.BACK TO MAIN MENU";

cout<<"\n\nPlease Enter Your Choice (1-9) ";

cin>>ch;

clrscr();

switch(ch)

{ case 1: memberinput(); break;

case 2: int num;

cout<<"\n\nEnter member ID.: ";

cin>>num;

readmem(num); break;

case 3: modifymem(); break;

case 4: delmem(); break;

case 5: bookinput(); break;

case 6: int n;

cout<<"\n\nEnter book ID.: ";

cin>>n;

readbook(n); break;

case 7: modifybook(); break;

case 8: delbook(); break;

case 9: return 0;

default:cout<<"\n\nWrong Choice!!";

}

adminmenu();

}

void main()

{

gotoxy(25,2);

cout<<"LIBRARY";

gotoxy(27,4);

cout<<"MANAGEMENT ";

gotoxy(29,6);

cout<<"PROJECT";

getch();

gotoxy(10,15);

cout<<"Made by:-";

gotoxy(15,19);

cout<<"VIKASH MISHRA";

gotoxy(19,23);

cout<<"YASH SRIVASTAVA";

gotoxy(23,27);

cout<<"POONAM PANCHAL";

getch();

cout<<"enter any key";

int choice;

do

{ clrscr();

cout<<"\n\n\n\tMAIN MENU";

cout<<"\n\n\t01. BOOK ISSUE";

cout<<"\n\n\t02. BOOK DEPOSIT";

cout<<"\n\n\t03. ADMINISTRATOR MENU";

cout<<"\n\n\t04. EXIT";

cout<<"\n\n\tPlease Select Your Option (1-4) ";

cin>>choice;

clrscr();

switch(choice)

{ case 1 : bookissue(); break;

case 2 : bookdeposit(); break;

case 3 : adminmenu(); break;

case 4 : exit(0);

default : cout<<"\n\nWrong Choice!!";

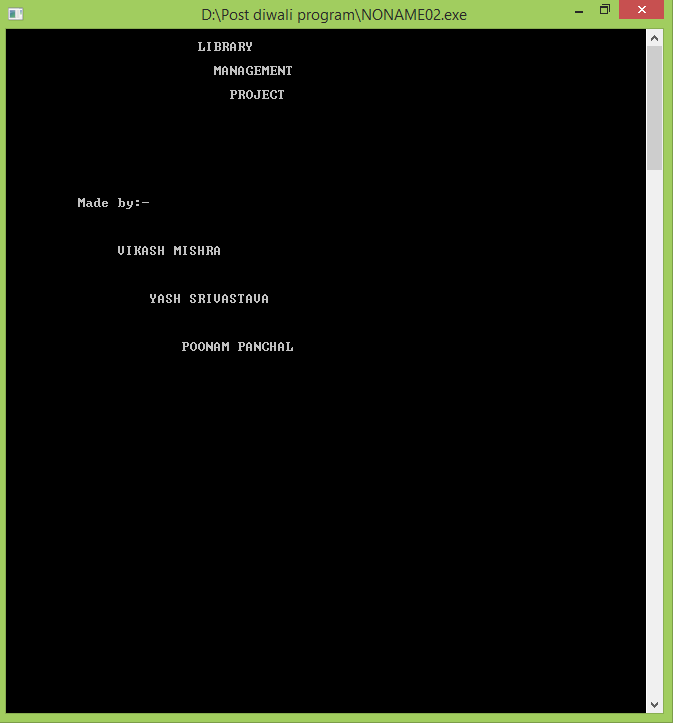
}

}while(choice!=4);

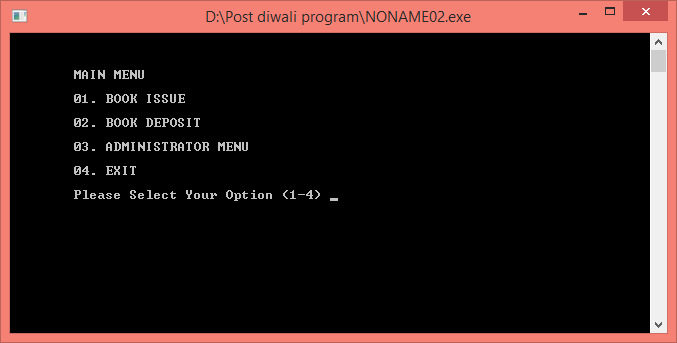
}

OUTPUT WINDOW :-

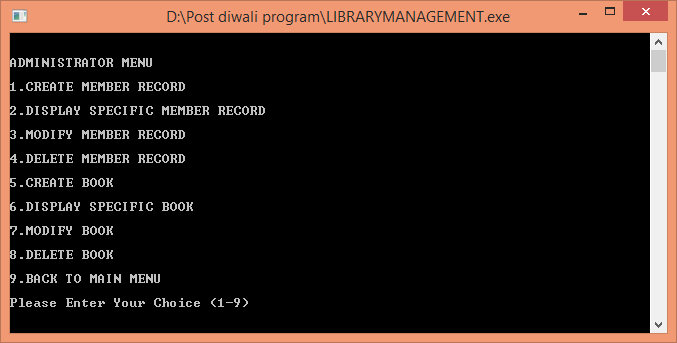
MAINSCREEN



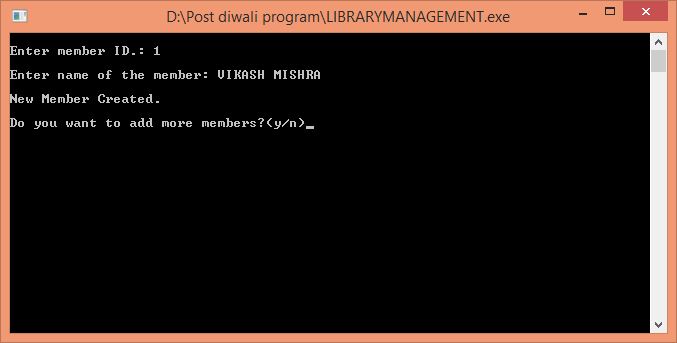
MAIN MENU



ADMINISTRATION MENU



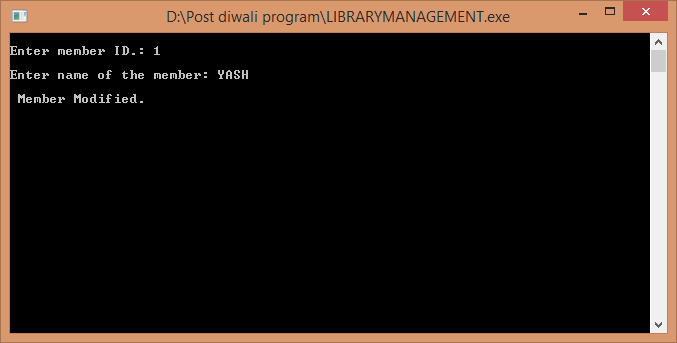
CREATE MEMBER RECORD



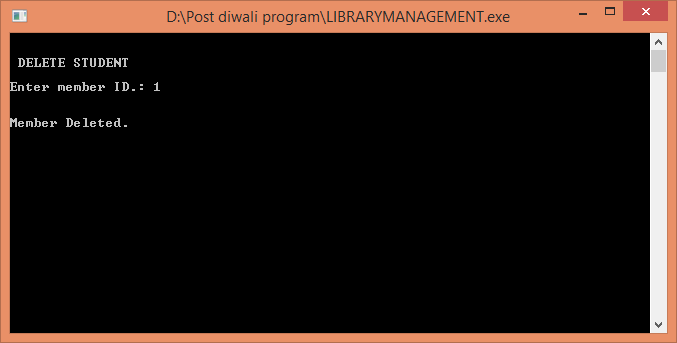
DISPLAY SPECIFIC MEMBER RECORD



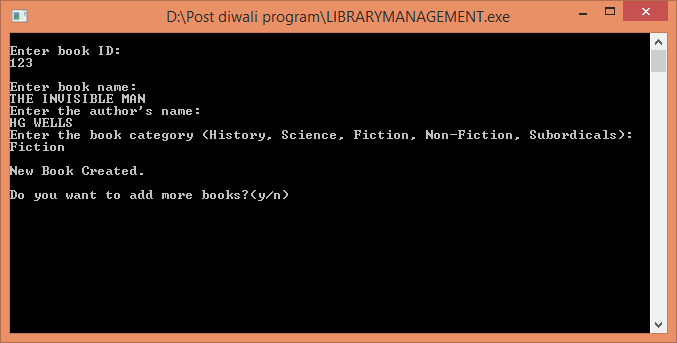
MODIFY MEMBER RECORD



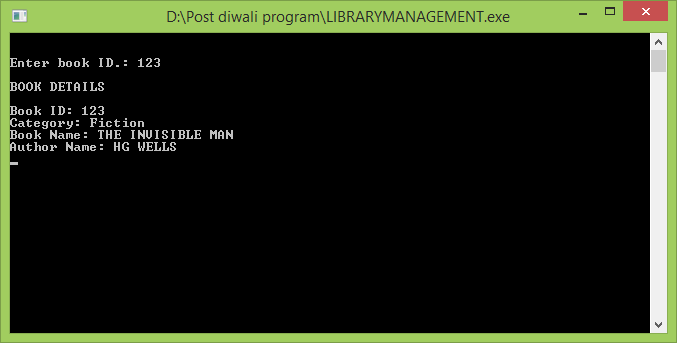
DELETE MEMBER RECORD



CREATE BOOK



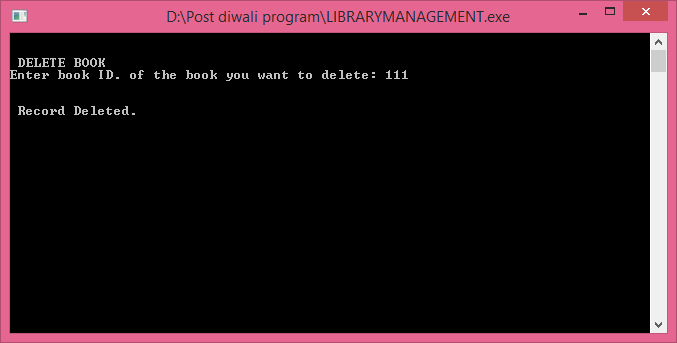
DISPLAY SPECIFIC BOOK



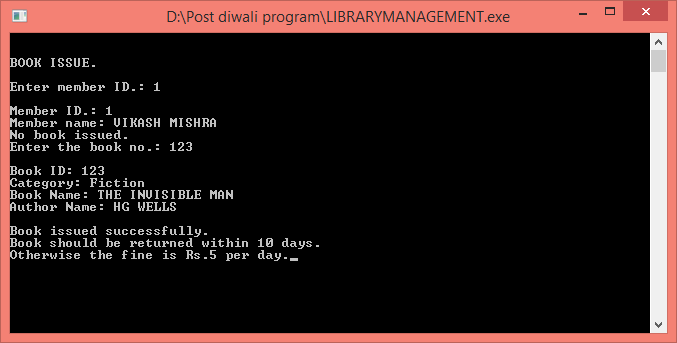
MODIFY BOOK RECORD



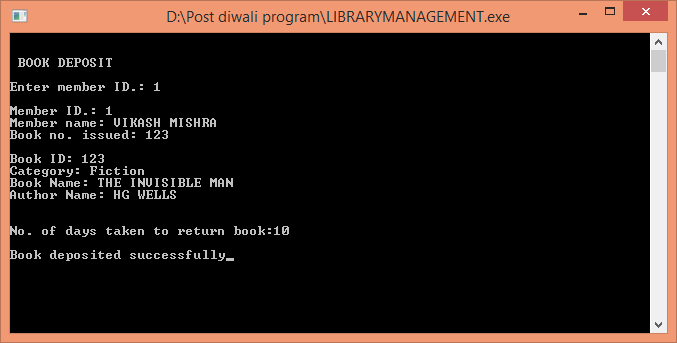
DELETE BOOK



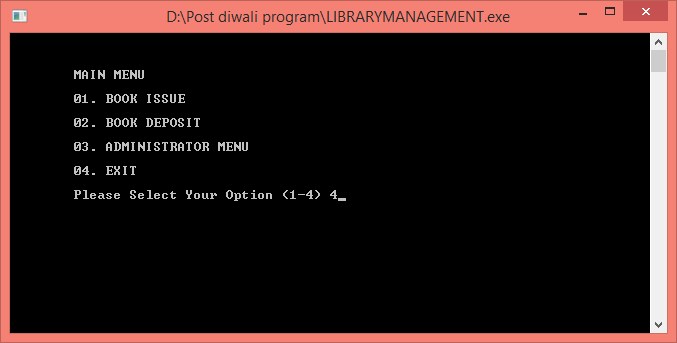
BOOK ISSUED



BOOK DEPOSIT



EXIT



BIBLIOGRAPHY :-

The Information Present in this project has been collected from following sources :-

* 1. Website :- (a) www.google.com

(b) [www.wikipedia.org](http://www.wikipedia.org)

2) Textbook :- (a) Computer Science with C++

By- Sumita Arora