

Delhi Technological University
Computer Science and Engineering Department
Data Structures (SE-201)
Mid-Term Examination Project



Project Information:

Title: Library Management System

Faculty: Ms Indu Singh, Assistant Professor
Computer Science and Engineering Department
Delhi Technological University
indusingh@dtu.ac.in

Prepared By: Anushka Sethi (2K19/SE/015) &
Aseem Sangalay (2K19/SE/021)

INDEX

ACKNOWLEDGEMENT	3
ABSTRACT	4
INTRODUCTION	5
OBJECTIVE OF THE PROJECT	6
AIM OF THE PROJECT	7
SOFTWARE & HARDWARE REQUIREMENTS	8
Software Requirements:	8
Hardware Requirements:	9
MODULES USED	9
CONCEPTS USED	11
Functions:	11
Arrays:	13
Loops:	14
Switch Statement:	14
Classes & Objects:	15
CODE	16
OUTPUTS	42
CONCLUSION	51
LIMITATIONS & FUTURE SCOPE	52
REFERENCES	53

ACKNOWLEDGEMENT

We, Anushka Sethi and Aseem Sangalay would like to start off by expressing our heartfelt gratitude towards our Data Structures professor **Dr. Indu Singh** for giving us the opportunity to undertake this project under her guidance. Her unending support and encouragement helped us to achieve our true potential.

We would also like to thank **Delhi Technological University** and the Department of Computer Science and Engineering for providing us the facilities which were necessary for us to complete this project and explore this topic as much as our hearts desired.

ABSTRACT

The computerized library management system has proved a boon for the management of large educational institutions. The colleges and universities have immensely benefited from it reducing cost and complexity involved in managing huge libraries, increasing productivity and performance and better accountability on the part of library staff.

The use of a library management system helps the librarian and other staff members to manage the library easily, along with saving time. This allows the librarian to catalogue books and to maintain records of issued, reissued and overdue books easily. It consists of the comprehensive options for entering the information related to books thus helping to maintain the complete library right from the transactions between student, staff and institute to issuing, returning and reissuing of books to maintaining membership information on one centralized server. This has eased and made the process of borrowing books error free for everyone.

In fact every little task which was done manually by library staff has been fully automated so as to smoothen the working of the library along with bringing transparency in its functioning and improving trust between the students and the management.

INTRODUCTION

Library Management System is based on a concept of recording owned books, issued books, returned books, students who have borrowed and many more. Before stepping into the main system a user has to pass through a login system to get access, there are two types of login i.e Student and Librarian Login. To get access to the Librarian's account, a password should be provided for the Librarian only. There are many features in this project. The user can also change the password from the system for their own security.

Talking about the features of the Library Management System, after logging in as a student he/she can only view booklists. All the system is controlled by the librarian as he/she can view, search, modify, add, delete books, and issue books. The other main feature contains reissuing of books and returning of books. The user can also search for students who issued books from the library. While adding books, he/she has to provide the details which includes the book, author and publication's name, book Id, price, and quantity. Similarly, if a user wants to issue books for a few days, he can easily issue just by providing book's name, id, student's name with Id, the issued date and if the user wants to re-issue book then he/she has to provide student's id, book id, and new date. The user can add books easily as it is separated by different categories.

While returning the book, if a user returns late then the system automatically charges fine depending upon issued dates. Another feature is that the user can change the password easily from the system. The librarian login works as the admin panel in this system. The Library Management system is developed using C++ Programming Language and different variables, strings have been used for the development of it. This project provides the best system for managing library records.

OBJECTIVE OF THE PROJECT

The library management system is a project which aims in developing a computerized system to maintain all the daily work of the library. This project has many features which are generally not available in normal library management systems like facility of admin login through which the admin can monitor the whole system. The librarian after logging into his account can generate various options such as a *new book*, *issue book* and *return book details*. Overall this project of ours is being developed to help the staff of the library to maintain the library in the best way possible and also reduce the human efforts.

- The objective of this project is to make the management of a Library System easier and more efficient.
- It aims to make use of **Data Structures** such as Arrays, Functions, Switch Statements and Pointers. We also aim to include the concepts of Object-Oriented Programming such as classes and their objects.
- It will also use *binary files* for reading, writing and manipulating different data, using various binary file operations.

The various options present on the **Main Menu** are :

- Student
- Librarian
- Close Application

The various options available in the **Student Menu** are:

- View Booklist
- Search for a Book
- Go to Main Menu
- Close Application

The various options available in the **Librarian Menu** are:

- View Booklist
- Search for a Book
- Modify/Add Book
- Issue Book
- Go to Main Menu
- Change password
- Close Application

AIM OF THE PROJECT

This project is on how to perform library management effectively, fast and easily. Management using pen and paper is absurd in the modern era. These days most of the schools and institutions have large libraries and students frequently issue books, so libraries often find it difficult to manage the process manually.

On the other hand, a virtual library management system eliminates all these burdens. By using this system, all records can be generated quickly. This computerized system helps in managing the library daily activity in electronic format. It reduces the risk of paperwork such as file loss and file damage. It helps the user to manage the transaction or record more effectively and is timesaving. This is a very versatile project that allows the user whether a student or a librarian to access the library accordingly.

Thus the project aims in creating a good user-friendly library management system reducing human strain to a large extent.

SOFTWARE & HARDWARE REQUIREMENTS

Software Requirements:

- Programming Language: C++
- IDE: Sublime text, code block along with a gcc/g++/c++ 14 Compiler (such as MinGW/ Clang/ Xcode) installed.
- Windows Command Prompt/ MacOS Terminal/ Linux Terminal.
- Operating system: Windows 10/ MacOS/ Linux.

Hardware Requirements:

- Processor: Pentium Processor IV or higher.
- RAM: Minimum 4 GB RAM.
- Hard Disk: 60 GB Hard Disk or more.

MODULES USED

The project is majorly divided into two modules:

1. Librarian/ Administrator Module:

The librarian/Administrator module can be accessed only if the person has the authorised password or security key. Once the person is logged into the system, the person has following options/ sub-modules:

1.1 View Booklist:

The administrator can view all the books that have been stored in the system database.

1.2 Book Search:

The administrator can search for a specific book as per the requirements of a student/ administrator.

1.3 Book Issue:

The administrator can issue book(s) to students and also perform the following actions:

1.3.1 Book Return:

The administrator can update the records when a student returns a book.

1.3.2 Book Reissue:

The administrator can reissue a book to the student if required.

1.3.3 View Issued Books:

The administrator can view all the issued books at once.

1.3.4 Student Search:

The administrator can search a student's name and view the books issued by him/her.

1.4 Modify/ Append Book List:

The administrator can *add new books*, *delete books* and also *edit the book* details.

1.5 Change Password:

The administrator can change the password through this functionality.

2. Student Module:

2.1 View Booklist:

The administrator can view all the books that have been stored in the system database.

2.2 Book Search:

The administrator can search for a specific book as per the requirements of a student/administrator.

CONCEPTS USED

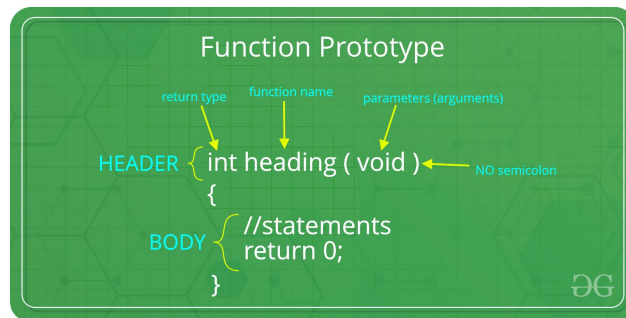
● **Functions:**

A function is a set of statements that take inputs, do some specific computation to produce an output.

The idea is to put some commonly or repeatedly done tasks together and make a function so that instead of writing the same code again and again for different inputs, we can call the function.

Why do we need functions?

- Functions help us in reducing code redundancy. If functionality is performed at multiple places in software, then rather than writing the same code, again and again, we create a function and call it everywhere. This also helps in maintenance as we have to change at one place if we make future changes to the functionality.
- Functions make code modular. Consider a big file having many lines of codes. It becomes really simple to read and use the code if the code is divided into functions.
- Functions provide abstraction. For example, we can use library functions without worrying about their internal working.



Function Declaration

A function declaration tells the compiler about the number of parameters function takes, data-types of parameters and return type of function. Putting parameter names in function declaration is optional in the function declaration, but it is necessary to put them in the definition.

Parameter Passing to functions

The parameters passed to function are called *actual parameters*.

The parameters received by function are called *formal parameters*.

There are two most popular ways to pass parameters:

Pass by Value: In this parameter passing method, values of actual parameters are copied to the function's formal parameters and the two types of parameters are stored in different memory locations. So any changes made inside functions are not reflected in the actual parameters of the caller.

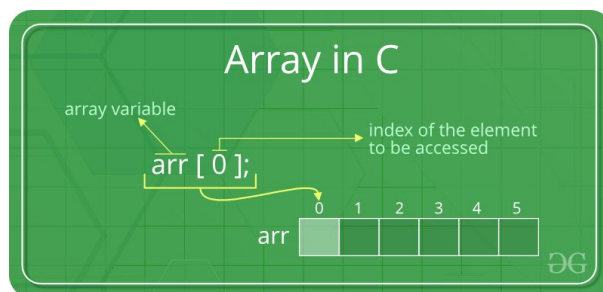
Pass by Reference Both actual and formal parameters refer to the same locations, so any changes made inside the function are actually reflected in actual parameters of the caller.

● Arrays:

An array is a collection of items stored at contiguous memory locations. The idea is to store multiple items of the same type together. This makes it easier to calculate the position of each element by simply adding an offset to a base value, i.e., the memory location of the first element of the array (generally denoted by the name of the array).

Advantages of using arrays:

- Arrays allow random access of elements. This makes accessing elements by position faster.
- Arrays have better *cache locality* that can make a pretty big difference in performance.
- Arrays represent multiple data items of the same type using a single name.



Types of indexing in array:

- 0 (zero-based indexing): The first element of the array is indexed by subscript of 0
- 1 (one-based indexing): The first element of the array is indexed by subscript of 1
- n (n-based indexing): The base index of an array can be freely chosen. Usually programming languages allowing n-based indexing also allow negative index values and other scalar data types like enumerations, or characters may be used as an array index.

Why do we need arrays?

We can use normal variables (v1, v2, v3, ..) when we have a small number of objects, but if we want to store a large number of instances, it becomes difficult to manage them with normal variables. The idea of an array is to represent many instances in one variable.

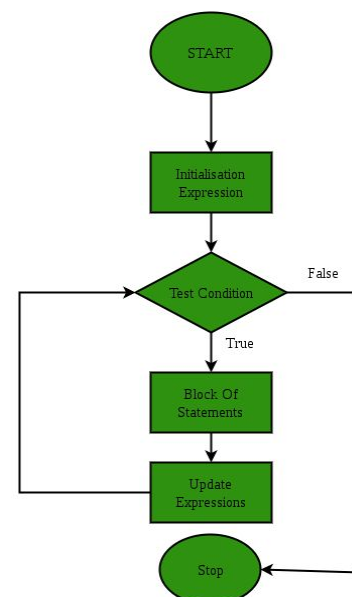
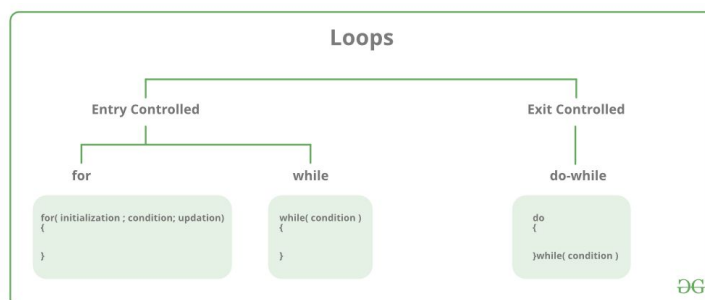
• Loops:

Loops in programming come into use when we need to repeatedly execute a block of statements. In computer programming, a loop is a sequence of instructions that is repeated until a certain condition is reached.

- An operation is done, such as getting an item of data and changing it, and then some condition is checked such as whether a counter has reached a prescribed number.
- **Counter not Reached:** If the counter has not reached the desired number, the next instruction in the sequence returns to the first instruction in the sequence and repeat it.
- **Counter reached:** If the condition has been reached, the next instruction “falls through” to the next sequential instruction or branches outside the loop.

There are mainly two types of loops:

1. **Entry Controlled loops:** In this type of loops the test condition is tested before entering the loop body. **For Loop** and **While Loop** are entry controlled loops.
2. **Exit Controlled Loops:** In this type of loop the test condition is tested or evaluated at the end of the loop body. Therefore, the loop body will execute at least once, irrespective of whether the test condition is true or false. **do – while loop** is exit controlled loop.



● Switch Statement:

Switch case statements are a substitute for long if statements that compare a variable to several integral values

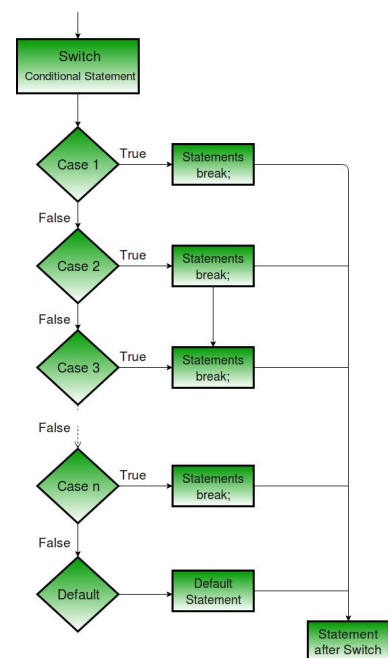
- The switch statement is a multiway branch statement. It provides an easy way to dispatch execution to different parts of code based on the value of the expression.
- Switch is a control statement that allows a value to change control of execution.

Syntax:

```
switch (n)
{
    case 1: // code to be executed if n = 1;
        break;
    case 2: // code to be executed if n = 2;
        break;
    default: // code to be executed if n doesn't match any cases
}
```

Important Points about Switch Case Statements:

1. The expression provided in the switch should result in a constant value otherwise it would not be valid.
2. Duplicate case values are not allowed.
3. The default statement is optional. Even if the switch case statement do not have a default statement, it would run without any problem.
4. The break statement is used inside the switch to terminate a statement sequence. When a break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.
5. The break statement is optional. If omitted, execution will continue on into the next case. The flow of control will fall through to subsequent cases until a break is reached.
6. Nesting of switch statements are allowed, which means you can have switch statements inside another switch. However nested switch statements should be avoided as it makes the program more complex and less readable.

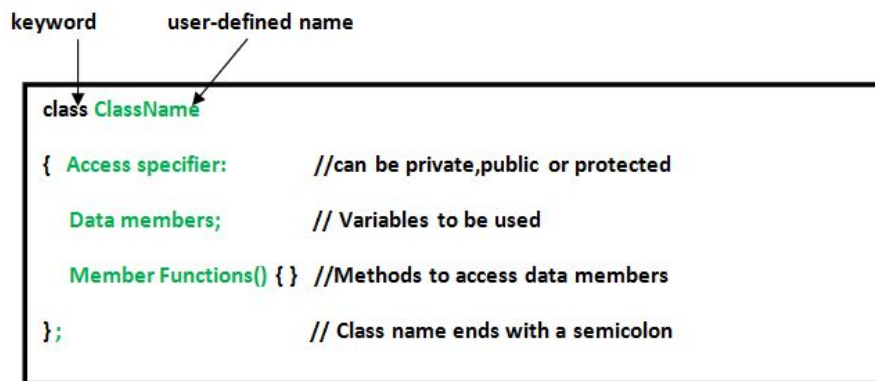


● Classes & Objects:

Class: A class in C++ is the building block that leads to Object-Oriented programming. It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A C++ class is like a blueprint for an object.

- A Class is a user defined data-type which has data members and member functions.
- Data members are the data variables and member functions are the functions used to manipulate these variables and together these data members and member functions define the properties and behavior of the objects in a Class.

An **Object** is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated.



CODE

```
#include<iostream>
#include<stdio.h>
#include<stdlib.h>
#include<fstream>
#include<string.h>
#include<conio.h>
using namespace std;

class Lib
{
    public:
        char bookname[100],auname[50],sc[20],sc1[50];
        int q,B,p;
        Lib()
        {
            strcpy(bookname,"NO Book Name");
            strcpy(auname,"No Author Name");
            strcpy(sc,"No Book ID");
            strcpy(sc1,"No Book ID");
            q=0;
            B=0;
            p=0;
        }

        void get();
        void student();
        void pass();
        void librarian();
        void password();
        void getdata();
        void show(int);
        void booklist(int);
        void modify();
        void see(int);
        int branch(int);
        void issue();
    }
```



```

        void der(char[],int,int);
        void fine(int,int,int,int,int,int);
};

void Lib::getdata()
{
    int i;
    fflush(stdin);
    cout<<"\n\t\tEnter the details :-\n";
    cout<<"\n\t\tEnter Book's Name : ";
    cin.getline(bookname,100);

    for(i=0;bookname[i]!='\0';i++)
    {
        if(bookname[i]>='a'&&bookname[i]<='z')
            bookname[i]-=32;
    }

    cout<<"\n\t\tEnter Author's Name : ";
    cin.getline(auname,50);
    cout<<"\n\t\tEnter Publication name : ";
    cin.getline(sc1,50);
    cout<<"\n\t\tEnter Book's ID : ";
    cin.getline(sc,20);
    cout<<"\n\t\tEnter Book's Price : ";
    cin>>p;
    cout<<"\n\t\tEnter Book's Quantity : ";
    cin>>q;
}

void Lib::show(int i)
{
    cout<<"\n\t\tBook Name : "<<bookname<<endl;
    cout<<"\n\t\tBook's Author Name : "<<auname<<endl;
    cout<<"\n\t\tBook's ID : "<<sc<<endl;
    cout<<"\n\t\tBook's Publication : "<<sc1<<endl;

    if(i==2)
    {
        cout<<"\n\t\tBook's Price : "<<p<<endl;
    }
}

```

```

        cout<<"\n\t\tBook's Quantity : "<<q<<endl;
    }
}
void Lib::booklist(int i)
{
    int b,r=0;
    system("cls");
    b=branch(i);
    system("cls");
    ifstream intf("Booksdata.txt",ios::binary);

    if(!intf)
        cout<<"\n\t\tFile Not Found.";
    else
    {
        cout<<"\n\t\t***** Book List ***** \n\n";
        intf.read((char*)this,sizeof(*this));
        while(!intf.eof())
        {
            if(b==B)
            {
                if(q==0 && i==1)
                {

                }
                else
                {
                    r++;
                    cout<<"\n\t\t***** "<<r<<".
***** \n";

                    show(i);
                }
            }
            intf.read((char*)this,sizeof(*this));
        }
        cout<<"\n\t\tPress any key to continue.....";
        getch();
        system("cls");
    }
}

```

```

        if(i==1)
            student();
        else
            librarian();
    }
void Lib::modify()
{
    char ch,st1[100];
    int i=0,b,cont=0;
    system("cls");

    cout<<"\n\t\t>>Please Choose one option :-\n";
    cout<<"\n\t\t1.Modification In Current Books\n\n\t\t2.Add New
Book\n\n\t\t3.Delete A Book\n\n\t\t4.Go back\n";
    cout<<"\n\n\t\tEnter your choice : ";
    cin>>i;

    if(i==1)
    {
        system("cls");
        b=branch(2);
        ifstream intf1("Booksdata.txt",ios::binary);
        if(!intf1)
        {
            cout<<"\n\t\tFile Not Found\n";
            cout<<"\n\t\tPress any key to continue.....";
            getch();
            system("cls");
            librarian();
        }

        intf1.close();
        system("cls");
        cout<<"\n\t\tPlease Choose One Option :-\n";
        cout<<"\n\t\t1.Search By Book Name\n\n\t\t2.Search By
Book's ID\n";
        cout<<"\n\t\tEnter Your Choice : ";
        cin>>i;
        fflush(stdin);
        if(i==1)

```

```

        {
            system("cls");
            cout<<"\n\t\tEnter Book Name : ";
            cin.getline(st1,100);
            system("cls");
            fstream
intf("Booksdata.txt",ios::in|ios::out|ios::ate|ios::binary);
            intf.seekg(0);
            intf.read((char*)this,sizeof(*this));

            while(!intf.eof())
            {

for(i=0;b==B&&bookname[i]!='\0'&&st1[i]!='\0'&&(st1[i]==bookname[i]||s
t1[i]==bookname[i]+32);i++);
                if(bookname[i]=='\0'&&st1[i]=='\0')
                {
                    cont++;
                    getdata();

intf.seekp(intf.tellp()-sizeof(*this));

intf.write((char*)this,sizeof(*this));
                    break;
                }

intf.read((char*)this,sizeof(*this));
            }
            intf.close();
        }
        else if(i==2)
        {
            cout<<"\n\t\tEnter Book's ID : ";
            cin.getline(st1,100);
            system("cls");
            fstream
intf("Booksdata.txt",ios::in|ios::out|ios::ate|ios::binary);
            intf.seekg(0);
            intf.read((char*)this,sizeof(*this));

```

```

        while(!intf.eof())
        {
for(i=0;b==B&&sc[i]!='\0'&&st1[i]!='\0'&&st1[i]==sc[i];i++);
            if(sc[i]=='\0'&&st1[i]=='\0')
            {
                    cont++;
                    getdata();

intf.seekp(intf.tellp()-sizeof(*this));

intf.write((char*)this,sizeof(*this));

                    break;
            }
            intf.read((char*)this,sizeof(*this));
        }

        intf.close();
    }
    else
    {
        cout<<"\n\t\tIncorrect Input.....:\n";
        cout<<"\n\t\tPress any key to continue.....";
        getch();
        system("cls");
        modify();
    }

    if(cont==0)
    {
        cout<<"\n\t\tBook Not Found.\n";
        cout<<"\n\t\tPress any key to continue.....";
        getch();
        system("cls");
        modify();
    }
    else
        cout<<"\n\t\tUpdate Successful.\n";

```

```

}
else if(i==2)
{
    system("cls");
    B=branch(2);
    system("cls");
    getdata();
    ofstream outf("Booksdata.txt",ios::app|ios::binary);
    outf.write((char*)this,sizeof(*this));
    outf.close();
    cout<<"\n\t\tBook added Successfully.\n";
}
else if(i==3)
{
    system("cls");
    b=branch(2);
    ifstream intf1("Booksdata.txt",ios::binary);
    if(!intf1)
    {
        cout<<"\n\t\tFile Not Found\n";
        cout<<"\n\t\tPress any key to continue.....";
        getch();
        intf1.close();
        system("cls");
        librarian();
    }
    intf1.close();
    system("cls");
    cout<<"\n\t\tPlease Choose One Option for
deletion:-\n";
    cout<<"\n\t\t1.By Book Name\n\n\t\t2.By Book's ID\n";
    cout<<"\n\t\tEnter Your Choice : ";
    cin>>i;
    fflush(stdin);
    if(i==1)
    {
        system("cls");
        cout<<"\n\t\tEnter Book Name : ";

```

```

        cin.getline(st1,100);
        ofstream
outf("temp.txt",ios::app|ios::binary);
        ifstream intf("Booksdata.txt",ios::binary);
        intf.read((char*)this,sizeof(*this));

        while(!intf.eof())
        {

for(i=0;b==B&&bookname[i]!='\0'&&st1[i]!='\0'&&(st1[i]==bookname[i]||s
t1[i]==bookname[i]+32);i++);

if(bookname[i]=='\0'&&st1[i]=='\0')
        {
                cont++;

intf.read((char*)this,sizeof(*this));

                }
                else
                {

outf.write((char*)this,sizeof(*this));

intf.read((char*)this,sizeof(*this));
                }

        }

        intf.close();
        outf.close();
        system("erase Booksdata.txt");
        system("rename temp.txt Booksdata.txt");

}

else if(i==2)
{
        cout<<"\n\t\tEnter Book's ID : ";
        cin.getline(st1,100);

```

```

        ofstream
outf("temp.txt",ios::app|ios::binary);
        ifstream intf("Booksdata.txt",ios::binary);
        intf.read((char*)this,sizeof(*this));
        while(!intf.eof())
        {

for(i=0;b==B&&sc[i]!='\0'&&st1[i]!='\0'&&st1[i]==sc[i];i++);
            if(sc[i]=='\0'&&st1[i]!='\0')
            {
                cont++;

intf.read((char*)this,sizeof(*this));
            }
            else
            {
                outf.write((char*)this,sizeof(*this));
                intf.read((char*)this,sizeof(*this));
            }
        }
        outf.close();
        intf.close();
        system("erase Booksdata.txt");
        system("rename temp.txt Booksdata.txt");

    }

    else
    {
        cout<<"\n\t\tIncorrect Input.....(\n";
        cout<<"\n\t\tPress any key to continue.....";
        getch();
        system("cls");
        modify();
    }

    if(cont==0)
    {
        cout<<"\n\t\tBook Not Found.\n";
    }

```



```

        cout<<"\n\t\tPress any key to continue.....";
        getch();
        system("cls");
        modify();
    }
    else
        cout<<"\n\t\tDeletion Successful.\n";

}
else if(i==4)
{
    system("cls");
    librarian();
}
else
{
    cout<<"\n\t\tWrong Input.\n";
    cout<<"\n\t\tPress any key to continue.....";
    getch();
    system("cls");
    modify();
}

cout<<"\n\t\tPress any key to continue.....";
getch();
system("cls");
librarian();

}

int Lib::branch(int x)
{
    int i;
    cout<<"\n\t\t>>Please Choose one Branch :-\n";

    cout<<"\n\t\t1.COMPUTER\n\n\t\t2.EE\n\n\t\t3.EC\n\n\t\t4.CIVIL\n\n\t\t5.MECHANICAL\n\n\t\t6.1ST YEAR\n\n\t\t7.Go to menu\n";
    cout<<"\n\t\tEnter youur choice : ";
    cin>>i;

```

```

switch(i)
{
    case 1: return 1;
            break;
    case 2: return 2;
            break;
    case 3: return 3;
            break;
    case 4: return 4;
            break;
    case 5: return 5;
            break;
    case 6: return 6;
            break;
    case 7: system("cls");
            if(x==1)
                student();
            else
                librarian();
    default : cout<<"\n\t\tPlease enter correct option :(";
            getch();
            system("cls");
            branch(x);
}
}

```

```

void Lib::see(int x)
{
    int i,b,cont=0;
    char ch[100];
    system("cls");
    b=branch(x);
    ifstream intf("Booksdata.txt",ios::binary);
    if(!intf)
    {
        cout<<"\n\t\tFile Not Found.\n";
        cout<<"\n\t\t->Press any key to continue.....";
        getch();
    }
}

```

```

        system("cls");
        if(x==1)
            student();
        else
            librarian();
    }

```

```

system("cls");
cout<<"\n\t\tPlease Choose one option :-\n";
cout<<"\n\t\t1.Search By Name\n\n\t\t2.Search By Book's ID\n";
cout<<"\n\t\tEnter Your Choice : ";
cin>>i;
fflush(stdin);
intf.read((char*)this,sizeof(*this));

```

```

if(i==1)
{
    cout<<"\n\t\tEnter Book's Name : ";
    cin.getline(ch,100);
    system("cls");

    while(!intf.eof())
    {

```

```

        for(i=0;b==B&&q!=0&&bookname[i]!='\0'&&ch[i]!='\0'&&(ch[i]==bookname[i]
        ||ch[i]==bookname[i]+32);i++);

```

```

            if(bookname[i]=='\0'&&ch[i]=='\0')
            {
                cout<<"\n\t\tBook Found :-\n";
                show(x);
                cont++;
                break;
            }

```

```

            intf.read((char*)this,sizeof(*this));

```

```

        }
    else if(i==2)
    {
        cout<<"\n\t\tEnter Book's ID : ";

```

```

        cin.getline(ch,100);
        system("cls");

        while(!intf.eof())
        {
for(i=0;b==B&&q!=0&&sc[i]!='\0'&&ch[i]!='\0'&&ch[i]==sc[i];i++);
            if(sc[i]=='\0'&&ch[i]=='\0')
            {
                    cout<<"\n\t\tBook Found :-\n";
                    show(x);
                    cont++;
                    break;
            }
            intf.read((char*)this,sizeof(*this));
        }

    }
    else
    {
            cont++;
            cout<<"\n\t\tPlease enter correct option :(";
            getch();
            system("cls");
            see(x);
    }
    intf.close();
    if(cont==0)
        cout<<"\n\t\tThis Book is not available :( \n";

        cout<<"\n\t\tPress any key to continue.....";
getch();
system("cls");
if(x==1)
    student();
else
    librarian();

```

```
}
```

```
void Lib::issue()
```

```
{
```

```
    char st[50],st1[20];
```

```
    int b,i,j,d,m,y,dd,mm,yy,cont=0;
```

```
    system("cls");
```

```
    cout<<"\n\t\t->Please Choose one option :-\n";
```

```
    cout<<"\n\t\t1.Issue Book\n\n\t\t2.View Issued
```

```
Book\n\n\t\t3.Search student who issued books\n\n\t\t4.Reissue
```

```
Book\n\n\t\t5.Return Book\n\n\t\t6.Go back to menu\n\n\t\tEnter Your  
Choice : ";
```

```
    cin>>i;
```

```
    fflush(stdin);
```

```
    if(i==1)
```

```
{
```

```
    system("cls");
```

```
    b=branch(2);
```

```
    system("cls");
```

```
    fflush(stdin);
```

```
    cout<<"\n\t\t->Please Enter Details :-\n";
```

```
    cout<<"\n\t\tEnter Book Name : ";
```

```
    cin.getline(bookname,100);
```

```
    cout<<"\n\t\tEnter Book's ID : ";
```

```
    cin.getline(sc,20);
```

```
    //strcpy(st,sc);
```

```
    der(sc,b,1);
```

```
    cout<<"\n\t\tEnter Student Name : ";
```

```
    cin.getline(auname,100);
```

```
    cout<<"\n\t\tEnter Student's ID : ";
```

```
    cin.getline(sc1,20);
```

```
    cout<<"\n\t\tEnter date : ";
```

```
    cin>>q>>B>>p;
```

```
    ofstream outf("student.txt",ios::binary|ios::app);
```

```
    outf.write((char*)this,sizeof(*this));
```

```
    outf.close();
```

```
    cout<<"\n\n\t\tIssue Successfully.\n";
```

```
}
```

```

else if(i==2)
{
    ifstream intf("student.txt",ios::binary);
    system("cls");
    cout<<"\n\t\t->The Details are :-\n";
    intf.read((char*)this,sizeof(*this));
    i=0;

    while(!intf.eof())
    {
        i++;
        cout<<"\n\t\t***** "<<i<<" ***** \n";
        cout<<"\n\t\tStudent Name :
"<<auname<<"\n\t\t"<<"Student's ID : "<<sc1<<"\n\t\t"<<"Book Name :
"<<bookname<<"\n\t\t"<<"Book's ID : "<<sc<<"\n\t\t"<<"Date :
"<<q<<"/"<<B<<"/"<<p<<"\n";
        intf.read((char*)this,sizeof(*this));
    }
    intf.close();
}
else if(i==3)
{
    system("cls");
    fflush(stdin);
    cout<<"\n\t\t->Please Enter Details :-\n";
    cout<<"\n\n\t\tEnter Student Name : ";
    cin.getline(st,50);
    cout<<"\n\n\t\tEnter Student's ID : ";
    cin.getline(st1,20);
    system("cls");
    ifstream intf("student.txt",ios::binary);
    intf.read((char*)this,sizeof(*this));
    cont=0;
    while(!intf.eof())
    {
        for(i=0;sc1[i]!='\0'&&st1[i]!='\0'&&st1[i]==sc1[i];i++);
        if(sc1[i]=='\0'&&st1[i]=='\0')
        {

```

```

        cont++;
        if(cont==1)
        {
            cout<<"\n\t\t->The Details are :-\n";
            cout<<"\n\t\tStudent Name : "<<auname;
            cout<<"\n\t\tStudent's ID : "<<sc1;
        }
        cout<<"\n\n\t\t***** "<<cont<<". Book details
*****\n";

        cout<<"\n\t\tBook Name : "<<bookname;
        cout<<"\n\t\tBook's ID : "<<sc;
        cout<<"\n\t\tDate : "<<q<<"/"<<B<<"/"<<p<<"\n";
    }

    intf.read((char*)this,sizeof(*this));

}

intf.close();
if(cont==0)
    cout<<"\n\t\tNo record found.";
}

else if(i==4)
{
    system("cls");
    fflush(stdin);
    cout<<"\n\t\t->Please Enter Details :-\n";
    cout<<"\n\n\t\tEnter Student's ID : ";
    cin.getline(st,50);
    cout<<"\n\t\tEnter Book's ID : ";
    cin.getline(st1,20);
    fstream
intf("student.txt",ios::in|ios::out|ios::ate|ios::binary);
    intf.seekg(0);
    intf.read((char*)this,sizeof(*this));
    while(!intf.eof())
    {
        for(i=0;sc[i]!='\0'&&st1[i]!='\0'&&st1[i]==sc[i];i++);
        for(j=0;sc1[j]!='\0'&&st[j]!='\0'&&st[j]==sc1[j];j++);
    }
}

```

```

if(sc[i]!='\0'&&sc1[j]!='\0'&&st[j]!='\0'&&st1[i]!='\0')
{
    d=q;
    m=B;
    y=p;
    cout<<"\n\t\tEnter New Date : ";
    cin>>q>>B>>p;
    fine(d,m,y,q,B,p); //fn1
    intf.seekp(intf.tellp()-sizeof(*this)); //fn3
    intf.write((char*)this,sizeof(*this)); //fn5
    cout<<"\n\n\t\tReissue successfully."; //fn3
    break;
}
    intf.read((char*)this,sizeof(*this));
}
    intf.close();
}

else if(i==5)
{
    system("cls");
    b=branch(2);
    system("cls");
    fflush(stdin);
    cout<<"\n\t\t->Please Enter Details :-\n";
    cout<<"\n\t\tEnter Book's ID : ";
    cin.getline(st1,20);
    der(st1,b,2);
    cout<<"\n\n\t\tEnter Student's ID : ";
    cin.getline(st,20);
    cout<<"\n\t\tEnter Present date : ";
    cin>>d>>m>>y;
    ofstream outf("temp.txt",ios::app|ios::binary);
    ifstream intf("student.txt",ios::binary);
    intf.read((char*)this,sizeof(*this));
    while(!intf.eof())
    {
        for(i=0;sc[i]!='\0'&&st1[i]!='\0'&&st1[i]==sc[i];i++);
    }
}

```



```

        for(j=0;sc1[j]!='\0'&&st[j]!='\0'&&st[j]==sc1[j];j++);

if(sc[i]=='\0'&&sc1[j]=='\0'&&st[j]=='\0'&&st1[i]=='\0'&&cont==0)
    {
        cont++;
        intf.read((char*)this,sizeof(*this));
        fine(q,B,p,d,m,y);
        cout<<"\n\t\tReturned successfully.";
    }
else
    {
        outf.write((char*)this,sizeof(*this));
        intf.read((char*)this,sizeof(*this));
    }
}

    intf.close();
    outf.close();
    getch();
    remove("student.txt");
    rename("temp.txt","student.txt");
}

else if(i==6)
{
    system("cls");
    librarian();
}
else
    cout<<"\n\t\tWrong Input.\n";

    cout<<"\n\n\t\tPress any key to continue.....";
    getch();
    system("cls");
    librarian();
}

void Lib::fine(int d,int m,int y,int dd,int mm,int yy)
{

```

```

    long int n1,n2;
    int years,l,i;
    const int monthDays[12] = {31, 28, 31, 30, 31, 30,31, 31, 30, 31,
30, 31};
    n1 = y*365 + d;
    for (i=0; i<m - 1; i++)
        n1 += monthDays[i]; //fn1353
    years = y;
    if (m <= 2)
        years--;
    l= years / 4 - years / 100 + years / 400;
    n1 += l;
    n2 = yy*365 + dd;
    for (i=0; i<mm - 1; i++)
        n2 += monthDays[i];
    years = yy;
    if (m <= 2)
        years--;
    l= years / 4 - years / 100 + years / 400;
    n2 += l;
    n1=n2-n1;
    n2=n1-15;
    if(n2>0)
        cout<<"\n\t\tSince the book was issued for more than 2 weeks. The
Total Fine incurred is : "<<n2;

```

```

}
void Lib::der(char st[],int b,int x)
{
    int i,cont=0;
    fstream
    intf("Booksdata.txt",ios::in|ios::out|ios::ate|ios::binary);
    intf.seekg(0);
    intf.read((char*)this,sizeof(*this));
    while(!intf.eof())
    {
        for(i=0;b==B&&sc[i]!='\0'&&st[i]!='\0'&&st[i]==sc[i];i++);
        if(sc[i]=='\0'&&st[i]=='\0')
        {

```

```

        cont++;
        if(x==1)
        {
            q--;
        }
        else
        {
            q++;
        }
        intf.seekp(intf.tellp()-sizeof(*this));
        intf.write((char*)this,sizeof(*this));
        break;
    }
    intf.read((char*)this,sizeof(*this));
}
if(cont==0)
{
    cout<<"\n\t\tBook not found.\n";
    cout<<"\n\n\t\tPress any key to continue.....";
    getch();
    system("cls");
    issue();
}
intf.close();
}

void Lib::get()
{
    int i;
    cout<<"\n\t***** LIBRARY MANAGEMENT SYSTEM
*****\n";
    cout<<"\n\t* Prepared by: Anushka Sethi 2K19/SE/015
*\n";
    cout<<"\n\t*                               Aseem Sangalay 2K19/SE/021
*\n";

    cout<<"\n\t*****\n";
    cout<<"\n\t\tPlease Choose Any Option To login \n";

```

```

        cout<<"\n\t\t1.Student\n\n\t\t2.Librarian\n\n\t\t3.Close
Application\n";
        cout<<"\n\t\tEnter your choice : ";
        cin>>i;
        if(i==1)
        {
            system("cls");
            student();
        }
        else if(i==2)
            pass();

        else if(i==3)
            exit(0);
        else
        {
            cout<<"\n\t\tPlease enter correct option :(";
            getch();
            system("CLS");
            get();
        }
    }
}

```

```

void Lib::student()
{
    int i;
    cout<<"\n\t***** WELCOME STUDENT *****\n";
    cout<<"\n\t\t>>Please Choose One Option:\n";
    cout<<"\n\t\t1.View BookList\n\n\t\t2.Search for a
Book\n\n\t\t3.Go to main menu\n\n\t\t4.Close Application\n";
    cout<<"\n\t\tEnter your choice : ";
    cin>>i;
    if(i==1)
        booklist(1);
    else if(i==2)
        see(1);
    else if(i==3)
    {
        system("cls");
    }
}

```

```

        get();
    }
    else if(i==4)
        exit(0);
    else
    {
        cout<<"\n\t\tPlease enter correct option :(";
        getch();
        system("cls");
        student();
    }
}

```

```

void Lib::pass()
{
    int i=0;
    char ch,st[21],ch1[21]="pass";
    cout<<"\n\t\tEnter Password : ";
    while(1)
    {
        ch=getch();
        if(ch==13)
        {
            st[i]='\0';
            break;
        }

        else if(ch==8&& i>0)
        {
            i--;
            cout<<"\b \b";
        }

        else
        {
            cout<<"*";
            st[i]=ch;
            i++;
        }
    }
}

```

```

    }
    ifstream inf("password.txt");
    inf>>ch1;
    inf.close();
    for(i=0;st[i]==ch1[i]&&st[i]!='\0'&&ch1[i]!='\0';i++);
        if(st[i]=='\0'&&ch1[i]=='\0')
        {
            system("cls");
            librarian();
        }
        else
        {
            cout<<"\n\n\t\tWrong Password.\n\n\t\ttry again.....\n";
            getch();
            system("cls");
            get();
        }
    }

void Lib::librarian()
{
    int i;
    cout<<"\n\t***** WELCOME LIBRARIAN *****\n";
    cout<<"\n\t\t>>Please Choose One Option:\n";
    cout<<"\n\t\t1.View BookList\n\n\t\t2.Search for a
Book\n\n\t\t3.Modify/Add Book\n\n\t\t4.Issue Book\n\n\t\t5.Go to main
menu\n\n\t\t6.Change Password\n\n\t\t7.Close Application\n";
    cout<<"\n\t\tEnter your choice : ";
    cin>>i;
    switch(i)
    {
        case 1:booklist(2);
            break;
        case 2:see(2);
            break;
        case 3:modify();
            break;
        case 4:issue();
            break;
    }
}

```

```

        case 5:system("cls");
                get();
                break;
        case 6:password();
                break;
        case 7:exit(0);
        default:cout<<"\n\t\tPlease enter correct option :(";
        getch();
        system("cls");
        librarian();
    }
}

```

```

void Lib::password()
{
    int i=0,j=0;
    char ch,st[21],ch1[21]={"pass"};
    system("cls");
    cout<<"\n\n\t\tEnter Old Password : ";
    while(1)
    {
        ch=getch();
        if(ch==13)
        {
            st[i]='\0';
            break;
        }
        else if(ch==8&&i>0)
        {
            i--;
            cout<<"\b \b";
        }
        else
        {
            cout<<"*";
            st[i]=ch;
            i++;
        }
    }
}

```

```

ifstream intf("password.txt");
intf>>ch1;
intf.close();
for(i=0;st[i]==ch1[i]&&st[i]!='\0'&&ch1[i]!='\0';i++);
if(st[i]=='\0'&&ch1[i]=='\0')
{
    system("cls");
    cout<<"\n\t**The Password Should be less than 20 characters &
don't use spaces**\n\n";
    cout<<"\n\t\tEnter New Password : ";
    fflush(stdin);
    i=0;
    while(1)
    {
        j++;
        ch=getch();
        if(ch==13)
        {
            for(i=0;st[i]!=' ' &&st[i]!='\0';i++);
            if(j>20 || st[i]==' ')
            {
                cout<<"\n\n\t\tYou didn't follow the
instruction \n\n\t\tPress any key for try again.....";
                getch();
                system("cls");
                password();
                librarian();
            }
            st[i]='\0';
            break;
        }
        else if(ch==8&&i>0)
        {
            i--;
            cout<<"\b \b";
        }
        else
        {

```



```

        cout<<"*";
        st[i]=ch;
        i++;
    }
}
ofstream outf("password.txt");
outf<<st;
outf.close();
cout<<"\n\n\t\tYour Password has been changed Successfully.";
cout<<"\n\t\tPress any key to continue.....";
getch();
system("cls");
librarian();
}
else
{
    cout<<"\n\n\t\tPassword is incorrect.....\n";
    cout<<"\n\t\tEnter 1 for retry or 2 for menu";
    cin>>i;
    if(i==1)
    {
        system("cls");
        password();
    }
    else
    {
        system("cls");
        librarian();
    }
}
}

int main()
{
    Lib obj;
    obj.get();
    getch();
    return 0;
}

```

OUTPUTS

- Opening page - the Main Menu

```
***** LIBRARY MANAGEMENT SYSTEM *****  
  
* Prepared by: Anushka Sethi 2K19/SE/015 *  
* Aseem Sangalay 2K19/SE/021 *  
* Delhi Technological University *  
*****  
  
Please Choose Any Option To login  
  
1.Student  
2.Librarian  
3.Close Application  
  
Enter your choice :
```

- Menu for the *Student* after pressing menu button 1 in the main menu

```
***** WELCOME STUDENT *****  
  
>>Please Choose One Option:  
  
1.View BookList  
2.Search for a Book  
3.Go to main menu  
4.Close Application  
  
Enter your choice :
```

- *Librarian* Menu after pressing menu button 2 in the main menu accessed by a login password

```
***** LIBRARY MANAGEMENT SYSTEM *****
* Prepared by: Anushka Sethi 2K19/SE/015 *
*           Aseem Sangalay 2K19/SE/021 *
*           Delhi Technological University *
*****
Please Choose Any Option To login
1.Student
2.Librarian
3.Close Application
Enter your choice : 2
Enter Password : ****
```

```
***** WELCOME LIBRARIAN *****
>>Please Choose One Option:
1.View BookList
2.Search for a Book
3.Modify/Add Book
4.Issue Book
5.Go to main menu
6.Change Password
7.Close Application
Enter your choice :
```

- When the *View Booklist* menu button is selected by the *Student*. A similar output is shown when *Librarian* selects the *View Booklist* menu button from their menu.

```
>>Please Choose one Branch :-
1.COMPUTER
2.EE
3.EC
4.CIVIL
5.MECHANICAL
6.1ST YEAR
7.Go to menu
Enter your choice : 1_
```

```
***** Book List *****
***** 1. *****
Book Name : DATABASE
Book's Author Name : Demo
Book's ID : 01
Book's Publication : Demo
***** 2. *****
Book Name : JAVA PROGRAMMING
Book's Author Name : Demo
Book's ID : 02
Book's Publication : Demo
Press any key to continue....._
```

- When the *Search for a book* menu button is selected by the *Student*. A similar output is shown when *Librarian* selects the *Search for a book* menu button from their menu.

```
>>Please Choose one Branch :-  
1.COMPUTER  
2.EE  
3.EC  
4.CIVIL  
5.MECHANICAL  
6.1ST YEAR  
7.Go to menu  
Enter your choice : 1_
```



```
Please Choose one option :-  
1.Search By Name  
2.Search By Book's ID  
Enter Your Choice : 1  
Enter Book's Name : database  
  
Book Found :-  
Book Name : DATABASE  
Book's Author Name : Demo  
Book's ID : 01  
Book's Publication : Demo  
Press any key to continue....._
```

- When the *Modify/Add Book* menu button is selected by the *Librarian*. It presents another menu from where the *Add New Book* menu button is selected to add a book into the library.

```
>>Please Choose one option :-  
1.Modification In Current Books  
2.Add New Book  
3.Delete A Book  
4.Go back  
Enter your choice : 2
```

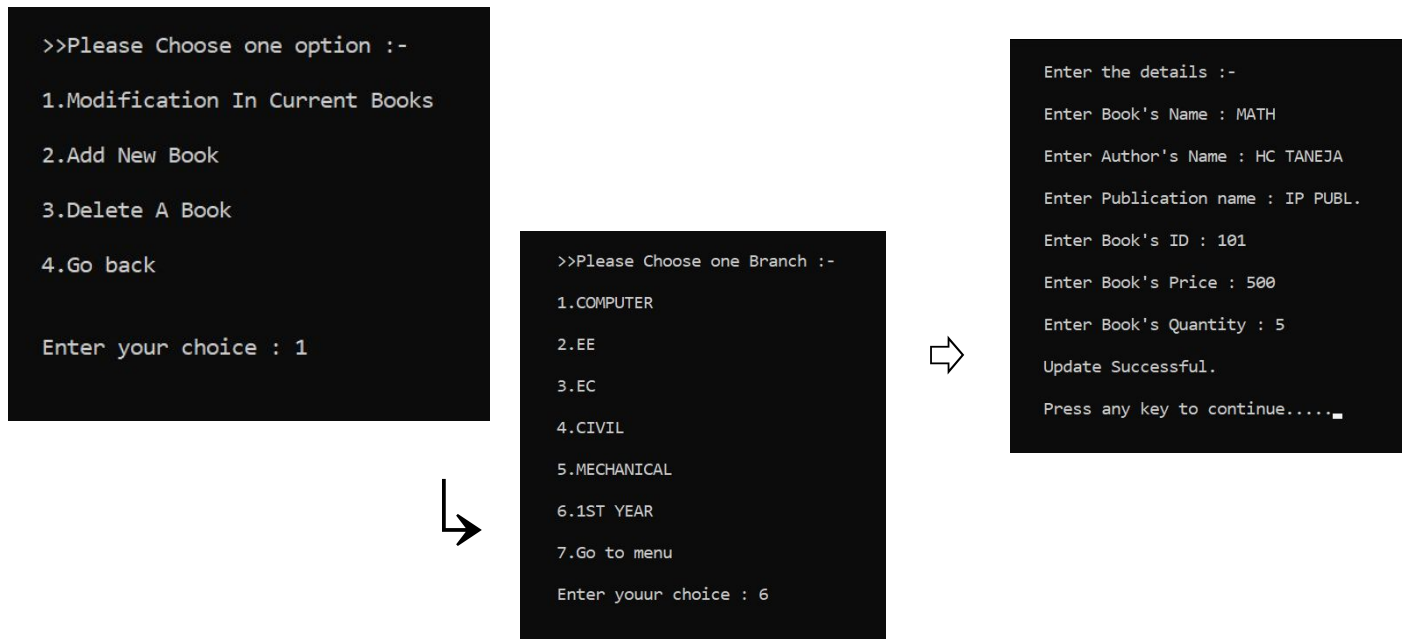


```
>>Please Choose one Branch :-  
1.COMPUTER  
2.EE  
3.EC  
4.CIVIL  
5.MECHANICAL  
6.1ST YEAR  
7.Go to menu  
Enter your choice : 6
```

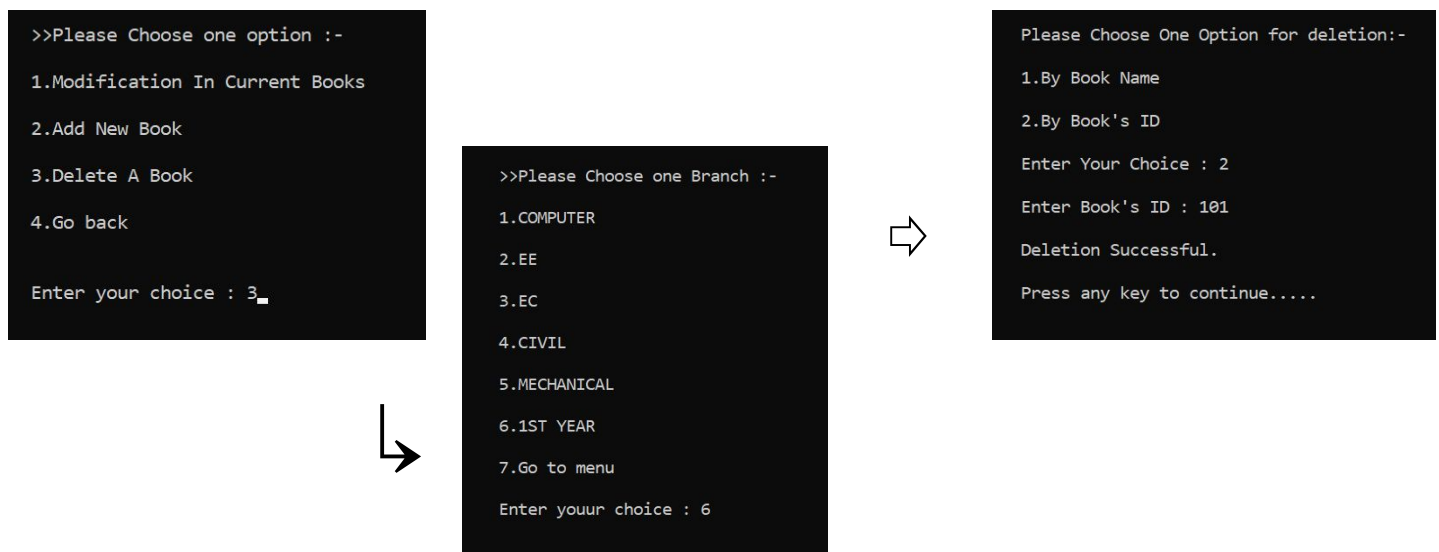


```
Enter the details :-  
Enter Book's Name : MATHEMATICS  
Enter Author's Name : HC TANEJA  
Enter Publication name : IP PUBL.  
Enter Book's ID : 101  
Enter Book's Price : 450  
Enter Book's Quantity : 3  
Book added Successfully.  
Press any key to continue.....
```

- **Adding a Book:** When the *Modify/Add Book* menu button is selected by the *Librarian*. It presents another menu from where the *Modification In Current Books* menu button is selected to modify a book detail in the library.



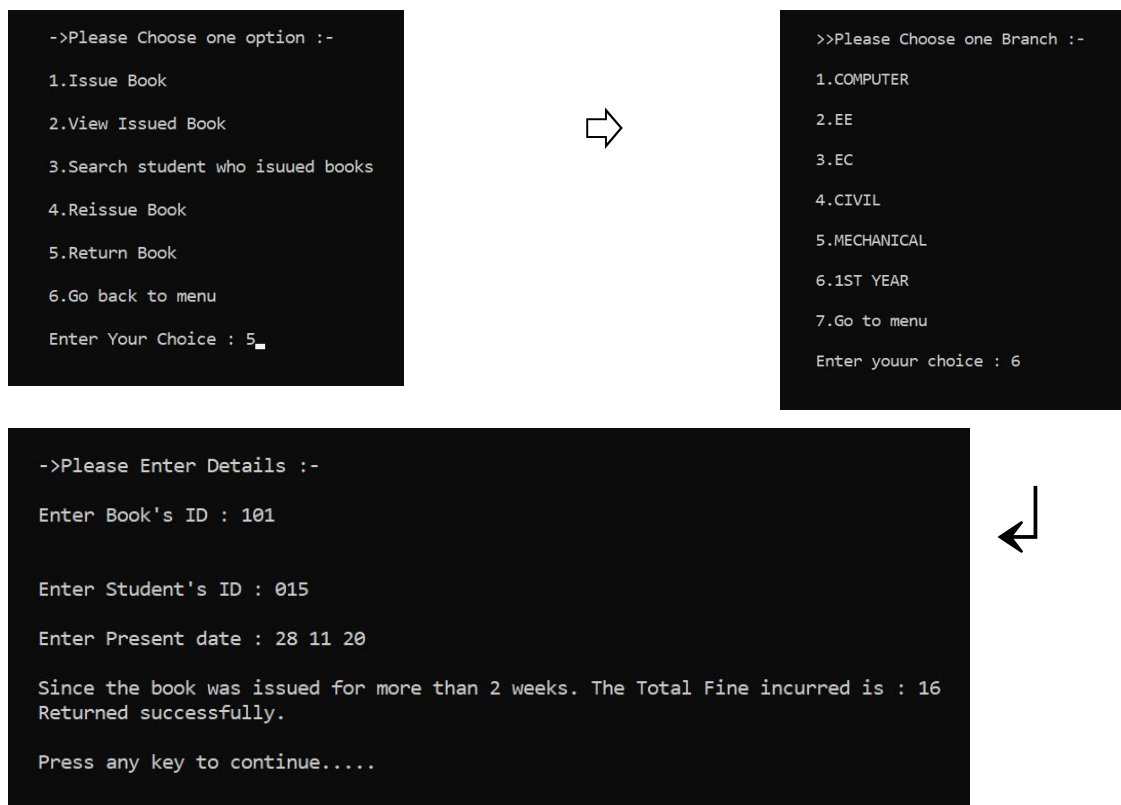
- **Deleting a Book:** When the *Modify/Add Book* menu button is selected by the *Librarian*. It presents another menu from where the *Delete A Book* menu button is selected to delete a book from the library.



- **Issuing of Book:** When the *Issue Book* menu button is selected by the *Librarian*. It presents another menu from where the *Issue Book* menu button is selected to issue a book from the library.



- **Return of Book:** When the *Issue Book* menu button is selected by the *Librarian*. It presents another menu from where the *Return Book* menu button is selected to return a book to the library.



- **View Issued Book**

```
->Please Choose one option :-  
  
1.Issue Book  
  
2.View Issued Book  
  
3.Search student who issued books  
  
4.Reissue Book  
  
5.Return Book  
  
6.Go back to menu  
  
Enter Your Choice : 2_
```

```
->The Details are :-  
  
***** 1. *****  
  
Student Name : Anushka  
Student's ID : 015  
Book Name : MATH  
Book's ID : 101  
Date : 28/10/20  
  
Press any key to continue....._
```

- **Search student who Issued Book**

```
->Please Choose one option :-  
  
1.Issue Book  
  
2.View Issued Book  
  
3.Search student who issued books  
  
4.Reissue Book  
  
5.Return Book  
  
6.Go back to menu  
  
Enter Your Choice : 3_
```

```
->The Details are :-  
  
Student Name : Anushka  
Student's ID : 015  
  
***** 1. Book details *****  
  
Book Name : MATH  
Book's ID : 101  
Date : 28/10/20  
  
Press any key to continue....._
```

- **Reissue Book**

```
->Please Choose one option :-  
  
1.Issue Book  
  
2.View Issued Book  
  
3.Search student who issued books  
  
4.Reissue Book  
  
5.Return Book  
  
6.Go back to menu  
  
Enter Your Choice : 4
```

```
->Please Enter Details :-  
  
Enter Student's ID : 015  
  
Enter Book's ID : 101  
  
Enter New Date : 10 11 20  
  
Reissue successfully.  
  
Press any key to continue....._
```

- Change password for *Librarian* login

****The Password Should be less than 20 characters & don't use spaces****

Enter New Password : ****

Your Password has been changed Successfully.
Press any key to continue.....■

CONCLUSION

The following conclusions can be deduced from the development of the project.

- The Library Management System of the entire system improves the efficiency.
- It provides an interface which proves to be better when compared to the existing system.
- It gives appropriate access to the authorized users depending on their permissions.
- It effectively overcomes the delay in communications.
- Updating information becomes easier.
- System security, data security and reliability are the striking features.
- The system has adequate scope for modification in future if it is necessary.

LIMITATIONS & FUTURE SCOPE

- There is no human cooperation or association if clients have a few requests.
- Graphics could be included to make the application more user friendly and interactive.
- More aspects could be included into the project; for example, the student could have the access to return the issued book himself/herself instead of just the librarian doing so.

REFERENCES

- C++ Classes for a Digital Library
- <https://patentimages.storage.googleapis.com/88/73/64/b1a67bab843a0/US5787413.pdf>
- <https://www.geeksforgeeks.org/functions-in-c/>
- <https://www.geeksforgeeks.org/array-data-structure/>
- <https://www.geeksforgeeks.org/loops-in-c-and-cpp/>
- <https://www.geeksforgeeks.org/switch-statement-cc/>
- <https://www.geeksforgeeks.org/c-classes-and-objects/>
- Data Structures Using C: AS Tanenbaum