



LOVELY
PROFESSIONAL
UNIVERSITY

SIX WEEKS SUMMER TRAINING REPORT

On

C++ with Data Structure and Algorithm

Submitted by:

Name: Peddakotla Lohith Kumar Reddy

Registration No: 12101706

Programme Name : BTech(CSE).

Under the Guidance of Codetantra

School of Computer Science & Engineering
Lovely Professional University , Phagwara
(May –July 2023)

DECLARATION

I here by declare that I have completed my twelve weeks summer training at codetantra from May 2023 to July,2023. I have declare that I have worked with full dedication during these six weeks of training and my learning outcomes full fill the requirements of training for the award of degree of B.Tech(CSE) Lovely Professional University, Phagwara.

`Signature of Student

Name:Peddakotla Lohith Kumar Reddy

Reg.No:12101706

Start Date : May 2023

End Date : july 2023

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude I give to my mentor whose contribution in stimulating suggestions and encouragement helped me to coordinate my project especially in writing this report. I express my thanks to my institution Lovely Professional University for giving me an opportunity to learn this interesting topic. I also convey my regards to my faculty assistance all through this training named “c++ with Data structures”. Once again I would like to thank all my supporters from the core of my heart.

Training certificate from organization



TABLES OF CONTENT

- > INTRODUCTION**
- > TECHNOLOGY LEARNT**
- > PROFILE OF THE PROBLEM**
- > EXISTING SYSTEM**
- > PROBLEM ANALYSIS**
- > SOFTWARE REQUIREMENT ANALYSIS**
- > DESIGN**
- > TESTING**
- > IMPLEMENTATION**
- > PROJECT LEGACY**
- > BIBLIOGRAPHY**

INTRODUCTION

C/c++ language

C++ is a general-purpose, object-oriented programming language. It was created by Bjarne Stroustrup at Bell Labs circa 1980. C++ is very similar to C (invented by Dennis Ritchie in the early 1970s).

C++ is so compatible with C that it will probably compile over 99% of C programs without changing a line of source code. Though C++ is a lot of well-structured and safer language than C as it OOPs based.

Some computer languages are written for a specific purpose. Like, Java was initially devised to control toasters and some other electronics. C was developed for programming OS. Pascal was conceptualized to teach proper programming techniques. But C++ is a general-purpose language. It well deserves the widely acknowledged nickname "Swiss Pocket Knife of Languages."

Use of C++ Programming Language

Here are some prime uses of C++ Programming Language:

Operating Systems:

Whether it is Microsoft Windows or Mac OSX or Linux – all of the operating systems have some parts which are programmed in C++. It is the backbone of all the well-known OSs as C++ is a strongly typed and quick programming language, that makes it an ideal choice for developing an operating system.

Games:

Because of the fact that it is one of the fastest programming languages, C++ is widely used in programming of game development engines. C++ can easily manipulate hardware resources and it can also provide procedural programming for CPU intensive functions.

Browsers:

The rendering engines of various web browsers are programmed in C++ because of the speed it offers.

Libraries:

Many high-level libraries use C++ as the core programming language. For example, several Machine Learning libraries use C++ in the backend because of its speed.

Graphics:

C++ is widely used in almost all graphics applications that require fast rendering, image processing, real-time physics and mobile sensors.

Compilers:

Compilers of various programming languages use C++ as the backend programming language.

OOPS Concepts In C++

OOPs, or Object-oriented programming is an approach or a programming pattern where the programs are structured around objects rather than functions and logic. It makes the data partitioned into two memory areas, i.e., data and functions, and helps make the code flexible and modular.

Object-oriented programming mainly focuses on objects that are required to be manipulated. In OOPs, it can represent data as objects that have attributes and functions.

The earlier approaches to programming were not that good, and there were several limitations as well. Like in procedural-oriented programming, you cannot reuse the code again in the program, and there was the problem of global data access, and the approach couldn't solve the real-world problems very well.

In object-oriented programming, it is easy to maintain the code with the help of classes and objects. Using inheritance, there is code reusability, i.e., you don't have to write the same code again and again, which increases the simplicity of the program. Concepts like encapsulation and abstraction provide data hiding as well.

Now have a look at some basic concepts of C++ OOPs.

Basic Object-Oriented Programming (OOPS) Concept in C++

There are some basic concepts that act as the building blocks of OOPs.

Classes & Objects

Abstraction

Encapsulation

Inheritance

Polymorphism

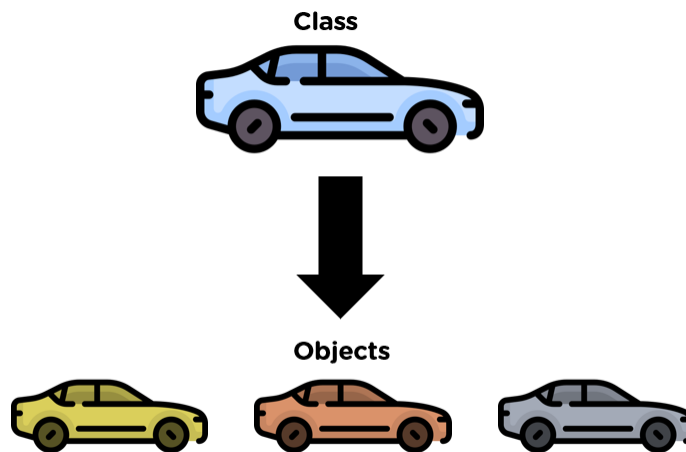
Object

An Object can be defined as an entity that has a state and behavior, or in other words, anything that exists physically in the world is called an object. It can represent a dog, a person, a table, etc.

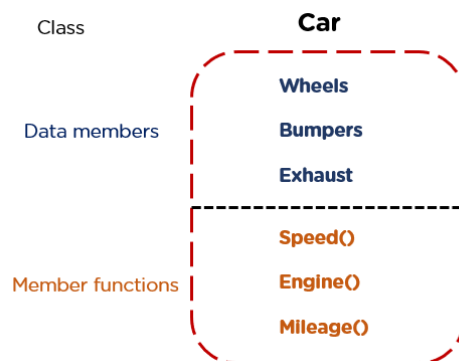
An object means the combination of data and programs, which further represent an entity.

Classes

Class can be defined as a blueprint of the object.
It is basically a collection of objects which act as building blocks.



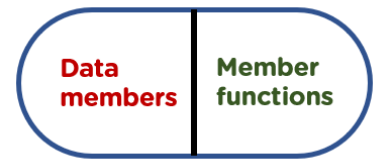
A class contains data members (variables) and member functions.
These member functions are used to manipulate the data members inside the class.



Abstraction

Abstraction helps in the data hiding process.

It helps in displaying the essential features without showing the details or the functionality to the user. It avoids unnecessary information or irrelevant details and shows only that specific part which the user wants to see.



Encapsulation

The wrapping up of data and functions together in a single unit is known as encapsulation.

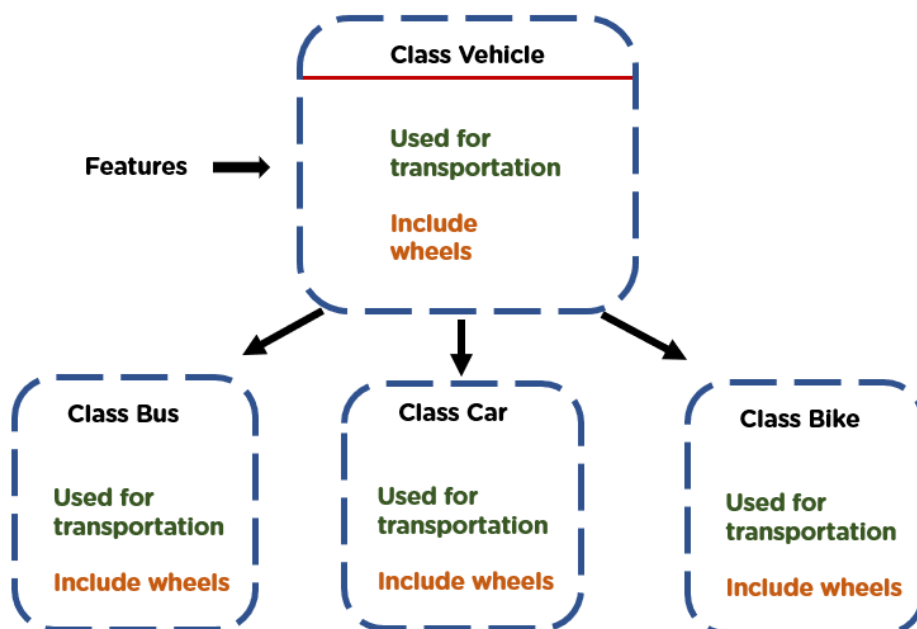
It can be achieved by making the data members' scope private and the member function's scope public to access these data members.

Encapsulation makes the data non-accessible to the outside world.



Inheritance

Inheritance is the process in which two classes have an is-a relationship among each other and objects of one class acquire properties and features of the other class. The class which inherits the features is known as the child class, and the class whose features it inherited is called the parent class. For example, Class Vehicle is the parent class, and Class Bus, Car, and Bike are child classes.



Polymorphism

Polymorphism means many forms. It is the ability to take more than one form.

It is a feature that provides a function or an operator with more than one definition.

It can be implemented using function overloading, operator overload, function overriding, virtual function.

Advantages of OOPs

There are various advantages of object-oriented programming.

OOPs provide reusability to the code and extend the use of existing classes. In OOPs, it is easy to maintain code as there are classes and objects, which helps in making it easy to maintain rather than restructuring.

It also helps in data hiding, keeping the data and information safe from leaking or getting exposed.

Object-oriented programming is easy to implement.

PROBLEM IDENTIFICATION AND CAUSE OF THE PROBLEM

A management system is a key tool in helping to streamline your business processes and build-in efficiency. Implementation of the appropriate management system and certifying to the appropriate standard to your business improves business performance and embeds safe and sustainable practices into your operations.

On a practical level, it demonstrates your capability to meet the needs of your customers and in doing so, empowers your business to improve its relationships with them too. An effective management system looks at managing and optimizing risks, improving performance and transparency, and fueling a culture of continual improvement. Implementation of the standard for that system is often the key driver to customer's expectations being met around the quality or trustworthiness of a business.

This can be achieved by adopting management system standards and getting ISO certified. Taking this approach is not just about box ticking either – for many businesses adopting management standards is a fundamental part of their growth strategy. By ensuring consistent improvement to their products and services as well as efficiency and performance, they secure a more robust and sustainable business, which in turn achieves the goal of meeting or exceeding customer expectations.

Data Structures and Algorithms

Data Structure is a way of collecting and organising data in such a way that we can perform operations on these data in an effective way. Data Structures is about rendering data elements in terms of some relationship, for better organization and storage. For example, we have some data which has, player's name "Virat" and age 26. Here "Virat" is of String data type and 26 is of integer data type.

We can organize this data as a record like Player record, which will have both player's name and age in it. Now we can collect and store player's records in a file or database as a data structure. For example: "Dhoni" 30, "Gambhir" 31, "Sehwag" 33

If you are aware of Object Oriented programming concepts, then a class also does the same thing, it collects different type of data under one single entity. The only difference being, data structures provides for techniques to access and manipulate data efficiently.

In simple language, Data Structures are structures programmed to store ordered data, so that various operations can be performed on it easily. It represents the knowledge of data to be organized in memory. It should be designed and implemented in such a way that it reduces the complexity and increases the efficiency.

Basic types of Data Structures

Anything that can store data can be called as a data structure, hence Integer, Float, Boolean, Char etc, all are data structures. They are known as Primitive Data Structures.

Then we also have some complex Data Structures, which are used to store large and connected data. Some example of Abstract Data Structure are :

Linked List
Tree
Graph
Stack, Queue etc.

All these data structures allow us to perform different operations on data. We select these data structures based on which type of operation is required.

Searching Algorithms

Not even a single day pass, when we do not have to search for something in our day to day life, car keys, books, pen, mobile charger and what not. Same is the life of a computer, there is so much data stored in it, that whenever a user asks for some data, computer has to search it's memory to look for the data and make it available to the user. And the computer has it's own techniques to search through it's memory fast, which you can learn more about in our Operating System tutorial series.

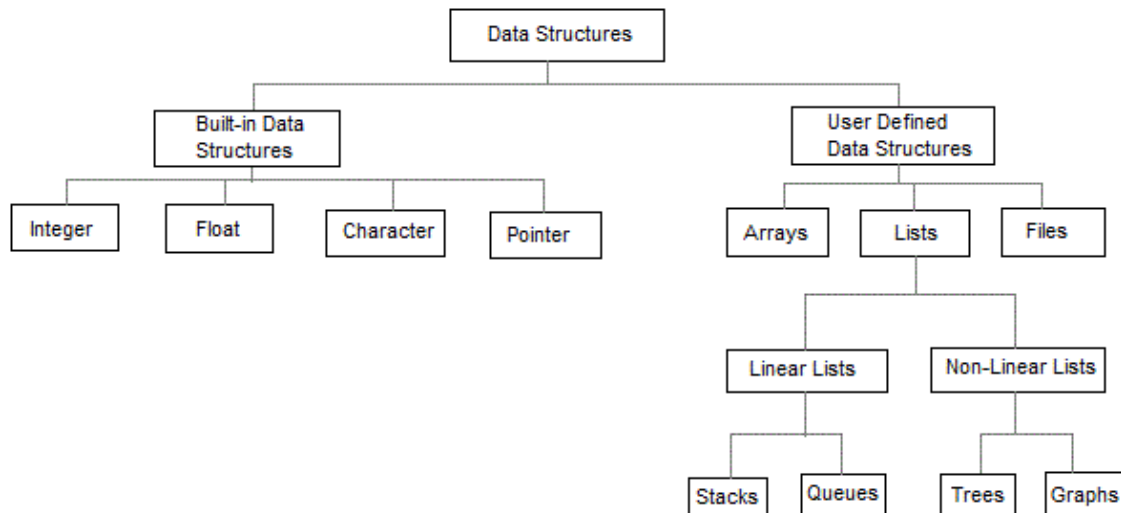
What if you have to write a program to search a given number in an array? How will you do it?

Well, to search an element in a given array, there are two popular algorithms available:

1. Linear Search
2. Binary Search

Features of Linear Search Algorithm

1. It is used for unsorted and unordered small list of elements.
2. It has a time complexity of $O(n)$, which means the time is linearly dependent on the number of elements, which is not bad, but not that good too.
3. It has a very simple implementation.



INTRODUCTION TO DATA STRUCTURES

What is an Algorithm ?

An algorithm is a finite set of instructions or logic, written in order, to accomplish a certain predefined task. Algorithm is not the complete code or program, it is just the core logic(solution) of a problem, which can be expressed either as an informal high level description as pseudocode or using a flowchart.

Every Algorithm must satisfy the following properties:

Input- There should be 0 or more inputs supplied externally to the algorithm.

Output- There should be atleast 1 output obtained.

Definiteness- Every step of the algorithm should be clear and well defined.

Finiteness- The algorithm should have finite number of steps.

Correctness- Every step of the algorithm must generate a correct output.

An algorithm is said to be efficient and fast, if it takes less time to execute and consumes less memory space. The performance of an algorithm is measured on the basis of following properties :

1. Time Complexity
2. Space Complexity

Features of Binary Search

- 1.It is great to search through large sorted arrays.
- 2.It has a time complexity of $O(\log n)$ which is a very good time complexity.
- 3.It has a simple implementation.

Sorting Algorithms

Sorting is nothing but arranging the data in ascending or descending order. The term sorting came into picture, as humans realised the importance of searching quickly.

There are so many things in our real life that we need to search for, like a particular record in database, roll numbers in merit list, a particular telephone number in telephone directory, a particular page in a book etc. All this would have been a mess if the data was kept unordered and unsorted, but fortunately the concept of sorting came into existence, making it easier for everyone to arrange data in an order, hence making it easier to search.

Sorting arranges data in a sequence which makes searching easier.

Sorting Efficiency

1. Time taken to sort the given data.
2. Memory Space required to do so.

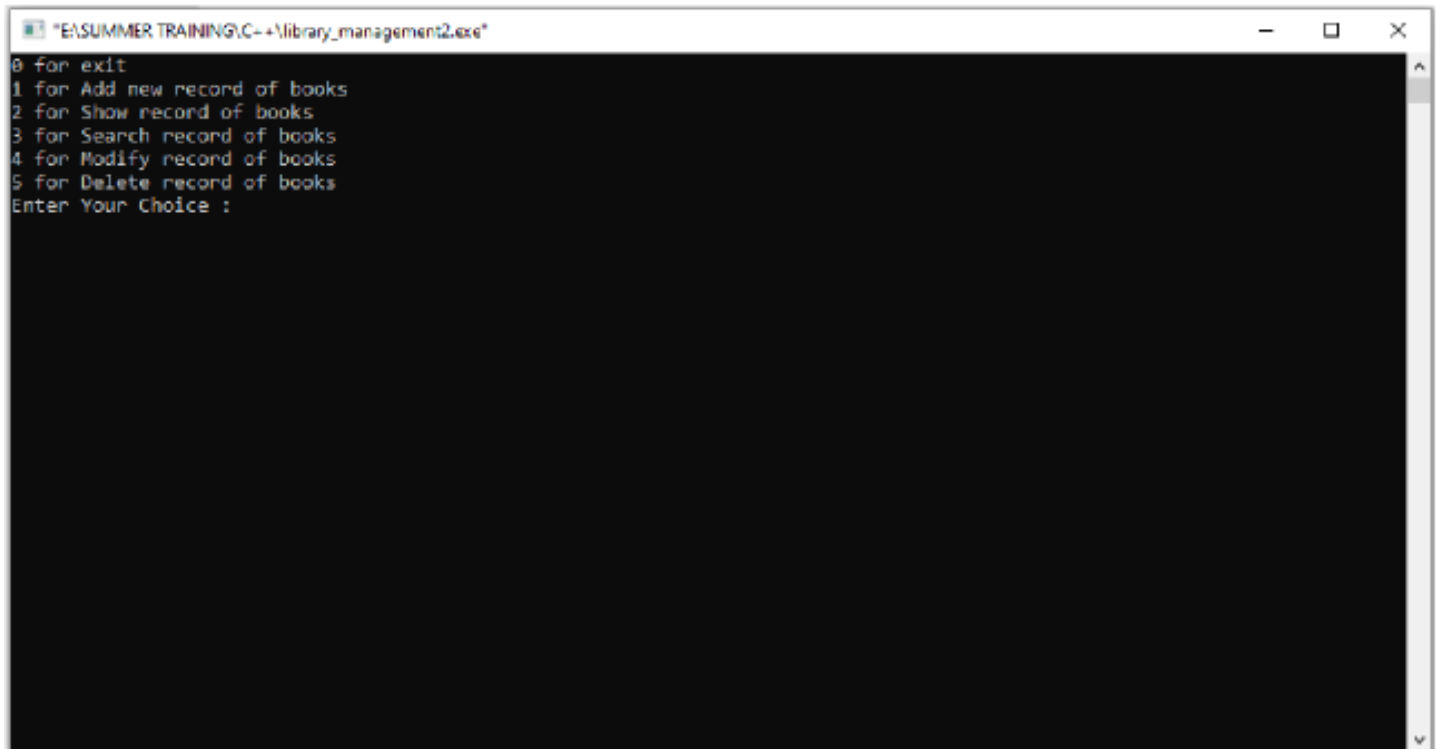
Different Sorting Algorithms

There are many different techniques available for sorting, differentiated by their efficiency and space requirements.

- 1. Bubble Sort**
- 2. Insertion Sort**
- 3. Selection Sort**
- 4. Quick Sort**
- 5. Merge Sort**
- 6.Heap Sort**

Technology Learnt

While coming to this phase, I have created an application programming i.e., Library management system and its actual purpose in it designed to help us to manage books in library



```
"E:\SUMMER TRAINING\C++\library_management2.exe"
0 for exit
1 for Add new record of books
2 for Show record of books
3 for Search record of books
4 for Modify record of books
5 for Delete record of books
Enter Your Choice :
```

It can also perform all crud operations like update,add,delete and search on database

PROFILE OF THE PROBLEM

TO CREATE A LIBRARY MANAGEMENT SYSTEM.TO MANAGE BOOKS IN THE LIBRARY

- > ADD BOOK
- > SEARCH BOOK
- > UPDATE BOOK
- > DELETE BOOK

EXISTING SYSTEM

It is hard to manage books in library while writing on copy its also hard to search for books. Hope my idea helps in managing books

Problem Analysis

A fully featured library management system where user can add book, update book, delete book, search book by name and its id.

Feasibility Analysis –

This study reveals all the technical aspects and its corresponding results. The project entitles “Library management System” is technically feasible because that the project was developed in using C++ Programming Language with Object Oriented Programming. It provides High-Level of reliability, availability, and compatibility.

As far as the project is concerned, there are financial benefits arising due to improved reliability and accuracy of information. The computerized system will help in automate the selection leading the profits and detail of the organization.

Software requirement analysis

- o Modern Operating System required minimum of windows 7 or 10 with 4GB RAM in minimum required i3 intel processors in 64-bit CPU as intel/AMD architecture or Linux-Ubuntu 16.04 to 17.10.
- o Required 5GB free disk space for C++ latest version IDLE module document to write code (in my suggestion).
- o We can use VS Code or CodeBlocks software's as IDE for C++ development.
- o Also, it needs at least one C++ installation to be available on the machine as a software creates an isolated virtual environment and can change it or create new interpreters according to the previous point.

Design

Tables and their relationships

This Library Management System project is in C++. Talking about the feature of this system, this c++ application is designed to manage library operations and it is also capable of handling all types of crud operations. Also, the design of this system is simple. so that the user won't get any difficulties while working on it.

Pseudo code-

```
#include<iostream>
#include<fstream>
#include<iomanip>
#include<stdlib.h>
#include<cstring>
#include<string>
#include<ctype.h>
using namespace std;
void capital(char s[])
{
    for(int i=0;i<=strlen(s);i++)
    {
        if(s[i]>=97 && s[i]<=122)
        {
            s[i]=s[i]-32;
        }
    }
}
```

CLASS BOOKS

```
class books
{
public:
    int id;
    char name[30];
    void getdata();
    void showdata();

    int retid()
    {
        return id;
    }

}obj;

void books :: getdata()
```

GET BOOK

```
void books :: getdata()
{
    cout<<" Enter Book id :";
    cin>>id;
    cout<<"Enter Book Name :";
    cin.ignore();

    cin.getline(name,30);
    capital(name);
}
```

SHOW BOOKS

```
void books ::showdata()
{
    cout<<"Book id : "<<id<<endl;

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

    cout<<"-----";
    cout<<"\n";

}
```

ADD BOOK

```
void write()
{
    ofstream fp2;
    fp2.open("books.dat", ios::binary|ios::out|ios::app);
    obj.getdata();
    fp2.write((char*)&obj, sizeof(obj));
    fp2.close();
    cout<<"\n\nData Successfully Saved to File...\n\n";
    system("pause");
}
```

READ BOOK

```
void read()
{
    ifstream fp1;
    fp1.open("books.dat", ios::in|ios::binary);
    while(fp1.read((char*)&obj, sizeof(obj)))
    {
        obj.showdata();
    }
    fp1.close();
    cout<<"\n\nData Reading from file completed Successfully ....\n";
    system("pause");
}
```

SEARCH BOOK BY ID

```
void searchdata(int n)
{
    int check=0;;

    ifstream fp1;
    fp1.open("books.dat", ios::binary);
    while(fp1.read((char*)&obj, sizeof(obj)))
    {
        if(obj.retid()==n){
            obj.showdata();
            check++;
        }
    }
    fp1.close();
    if(check==0)
    {
        cout<<"The book id . "<<n<<" not found....\n\n";
    }
    cout<<"\n\nData Searching Successfully From File....\n";
}
```

SEARCH BY NAME

```
void searchdata(char sl[])
{
    int check=0;

    ifstream fpl;
    fpl.open("books.dat",ios::binary);
    while(fpl.read((char*)&obj,sizeof(obj)))
    {
        if(strcmp(sl,obj.name)==0){
            obj.showdata();
            check++;
        }
    }
    fpl.close();
    if(check==0)
    {
        cout<<"The book name . "<<sl<<" not found....\n\n";
    }
    cout<<"\n\nData Searching Successfully From File....\n";
}
```

UPDATE BY ID

```
void modify(int n)
{
    int chk=0,currpos;
    fstream f;
    f.open("books.dat",ios::out|ios::binary|ios::in);
    while(f.read((char*)&obj,sizeof(obj)))
    {
        currpos=f.tellg();
        if(obj.retid()==n)
        {
            cout<<"The Following Book id. "<<n<<" will be modified with new data:\n\n";
            obj.showdata();
            cout<<"\n\nNow Enter the New Details of book....\n";
            obj.getdata();
            f.seekg(currpos-sizeof(obj));
            f.write((char*)&obj,sizeof(obj));
            chk++;
        }
    }
    f.close();
    if(chk==0){
        cout<<"The book id . "<<n<<" not found....\n\n";
        return ;
    }
    cout<<"\n\nData Modification Successfully Done in File....\n";
}
```

UPDATE BY NAME

```
void modify(char s2[])
{
    int chk=0,currpos;
    fstream f;
    f.open("books.dat",ios::out|ios::binary|ios::in);
    while(f.read((char*)&obj,sizeof(obj)))
    {
        currpos=f.tellg();
        if(strcmp(s2,obj.name)==0)
        {
            cout<<"The Following Book name. "<<s2<<" will be modified with new data:\n\n";
            obj.showdata();
            cout<<"\n\nNow Enter the New Details of book...\n";
            obj.getdata();
            f.seekg(currpos-sizeof(obj));
            f.write((char*)&obj,sizeof(obj));
            chk++;}
    }
    f.close();
    if(chk==0)
    {
        cout<<"The book name . "<<s2<<" not found...\n\n";
    }
    cout<<"\n\nData Updated Successfully Done in File...\n";
}
```

DELETE BY ID

```
void deletes(int n)
{
    int flag=0;
    ifstream fin;
    ofstream fout;

    fin.open("books.dat",ios::in|ios::binary);
    fout.open("Temp.dat",ios::out|ios::app|ios::binary);
    while(fin.read((char*)&obj,sizeof(obj)))
    {
        if(n!=obj.retid())
        {
            fout.write((char*)&obj,sizeof(obj));
        }
        else{
            cout<<"The Following book id. "<<n<<" has been deleted:\n";
            obj.showdata();
            flag++;
        }
    }
    fout.close();
    fin.close();
    if(flag==0)
        cout<<"The book id. "<<n<<" not found...\n\n";
    remove("books.dat");
    rename("temp.dat","books.dat");
}
```

DELETE BY NAME

```
void deletes(char s3[])
{
    int flag=0;
    ifstream fin;
    ofstream fout;

    fin.open("books.dat",ios::in|ios::binary);
    fout.open("Temp.dat",ios::out|ios::app|ios::binary);

    while(fin.read((char*)&obj,sizeof(obj)))
    {
        if(strcmp(s3,obj.name)!=0)
        {
            fout.write((char*)&obj,sizeof(obj));
        }
        else{
            cout<<"The Following book name. "<<s3<<" has been deleted:\n";
            obj.showdata();
            flag++;
        }
    }
    fout.close();

    fin.close();
    if(flag==0)
        cout<<"The book id. "<<s3<<" not found...\n\n";
    remove("books.dat");
    rename("temp.dat","books.dat");
}
```

SWITCH

```
void booksmanage()
{
    int num;
    do{
        system("cls");
        cout<< "0 for exit "<<endl;
        cout<< "1 for Add new record of books "<<endl;
        cout<< "2 for Show record of books "<<endl;
        cout<< "3 for Search record of books "<<endl;
        cout<< "4 for Modify record of books "<<endl;
        cout<< "5 for Delete record of books "<<endl;
        cout<< "Enter Your Choice : ";
        cin>>num;
        system("cls");
    }
```

EXTRA FEATURES

```
switch(num)
{
    case 1:write();break;
    case 2:read();break;
    case 3:

        { int n,num2;
          char s1[30];
          do{
              system("cls");
              cout<< "0 for go back "<<endl;
              cout<<"1 For search records by book id "<<endl;
              cout<<"2 For search records by book name "<<endl;
              cout<< "Enter Your Choice : ";
              cin>>num2;
              system("cls");
              switch(num2)
              {
                  case 1: {
                      cout<<"Enter Book id to search : ";
                      cin>>n;
                      searchdata(n);
                      break;
                  }

                  case 2: {
                      cout<<"Enter Book name to search : ";
                      cin.ignore();
                      cin.getline(s1,30);
                      capital(s1);
                      searchdata(s1);
                      break;
                  }
                  break;
              }

          }while(num2);

          system("pause");
        }
}
```

SWITCH

```
case 4:
{
    int n1,num3;
    char s2[30];
    do{
        system("cls");
        cout<< "0 for go back "<<endl;
        cout<<"1 For modify records by book id "<<endl;
        cout<<"2 For modify records by book name "<<endl;
        cout<< "Enter Your Choice : ";
        cin>>num3;
        system("cls");
        switch(num3)
        {
            case 1: {
                cout<<"Enter Book id to modify : ";
                cin>>n1;
                modify(n1);
                break;
            }

            case 2: {
                cout<<"Enter Book name to modify : ";
                cin.ignore();cin.getline(s2,30);
                capital(s2);
                modify(s2);
                break;
            }

        }
        system("pause");

    }while(num3);

}

case 5:
{
    int n2,num4;
    char s3[30];
    do{
        system("cls");
        cout<< "0 for go back "<<endl;
        cout<<"1 For delete records by book id "<<endl;
        cout<<"2 For delete records by book name "<<endl;
        cout<< "Enter Your Choice : ";
        cin>>num4;
        system("cls");
        switch(num4)
        {
            case 1: {
                cout<<"Enter Book id to delete : ";
                cin>>n2;
                deletes(n2);
                break;
            }

        }

    }

}
```


MAIN FUNCTION

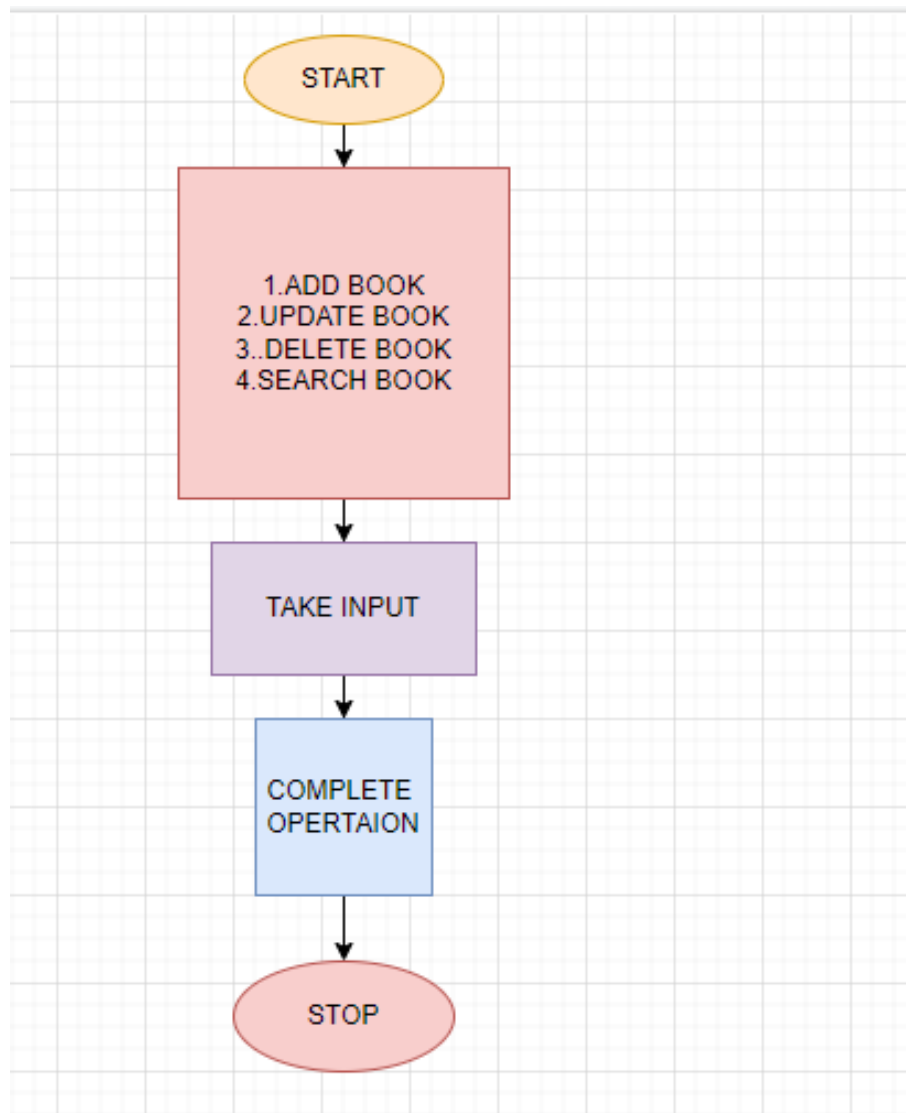
```
        cin>>n2;
        deletes(n2);
        break;
    }

    case 2: {
        cout<<"Enter Book name to delete : ";
        cin.ignore();cin.getline(s3,30);
        capital(s3);
        deletes(s3);
        break;
    }
    break;
}
system("pause");

}while(num4);
}
system("pause");
}
}while(num);
}
```

```
int main()
{
    booksmanage();
    return 0;
}
```

FLOW CHART

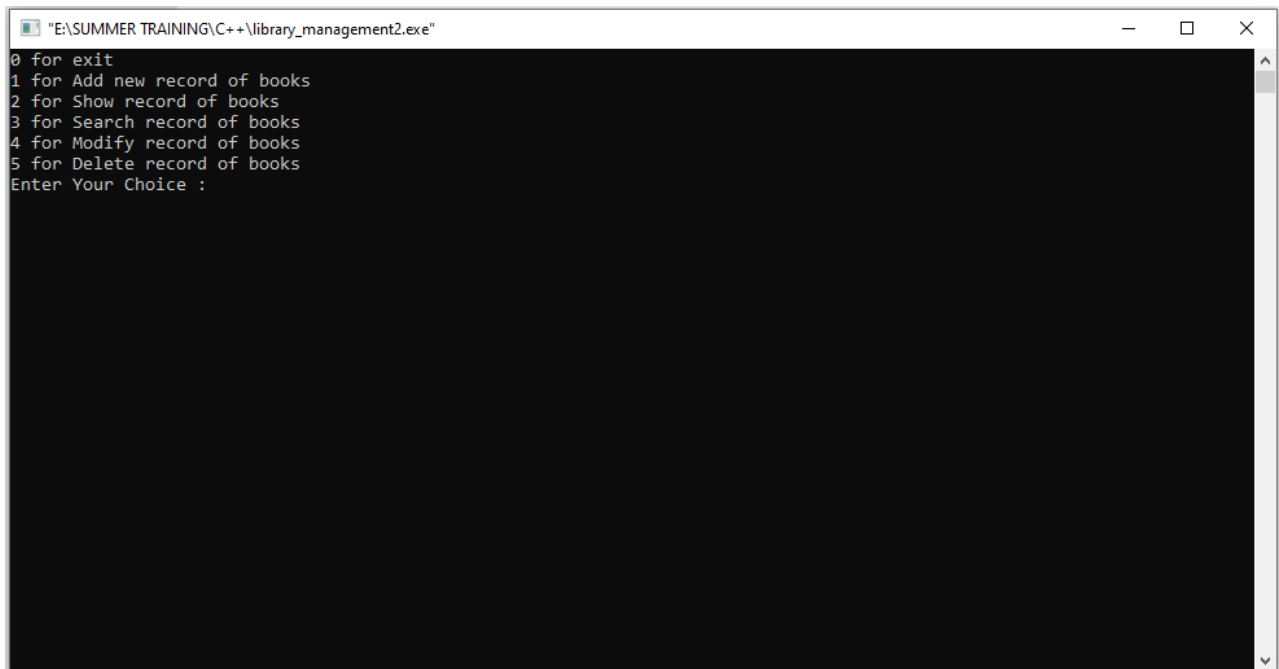


Testing

The project will run fine with zero errors and warning on any platform but in only one condition platform have c++ environment setup

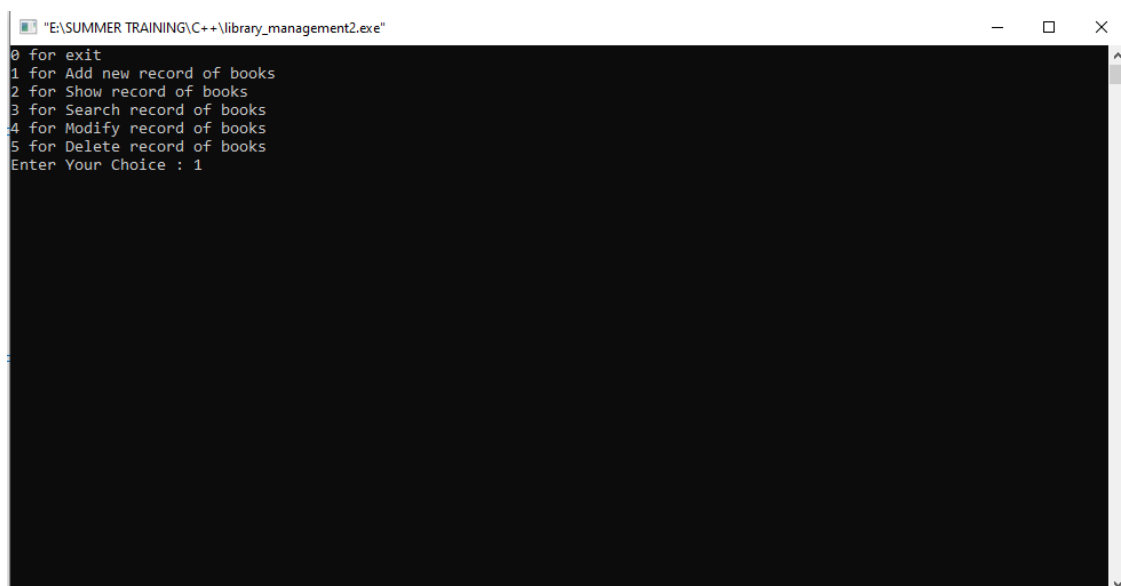
IMPLEMENTATION

LANDING PAGE

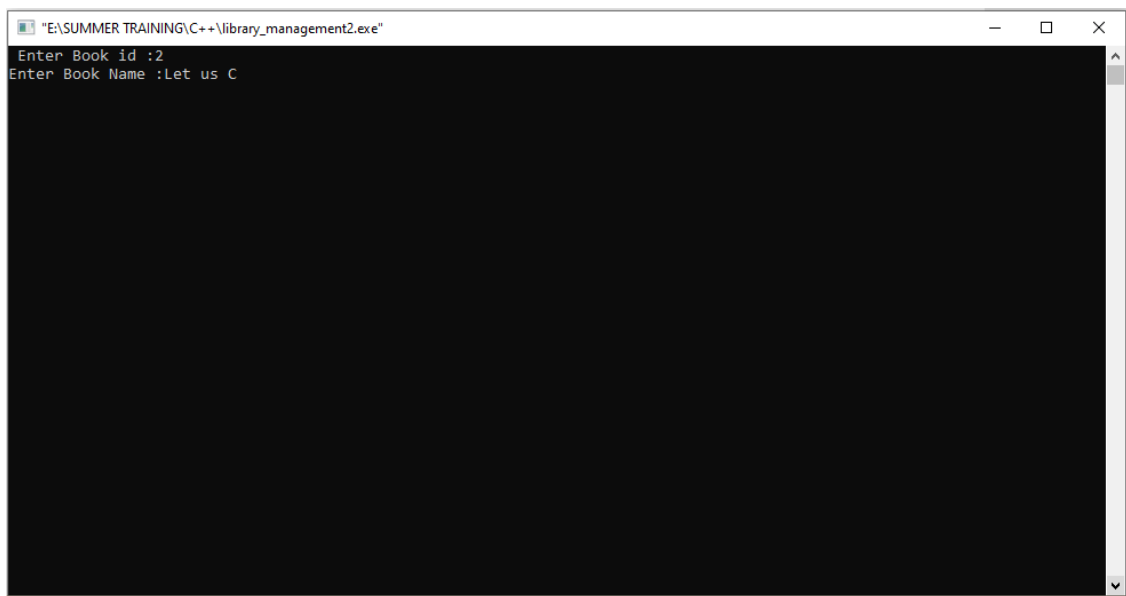


```
"E:\SUMMER TRAINING\C++\library_management2.exe"
0 for exit
1 for Add new record of books
2 for Show record of books
3 for Search record of books
4 for Modify record of books
5 for Delete record of books
Enter Your Choice :
```

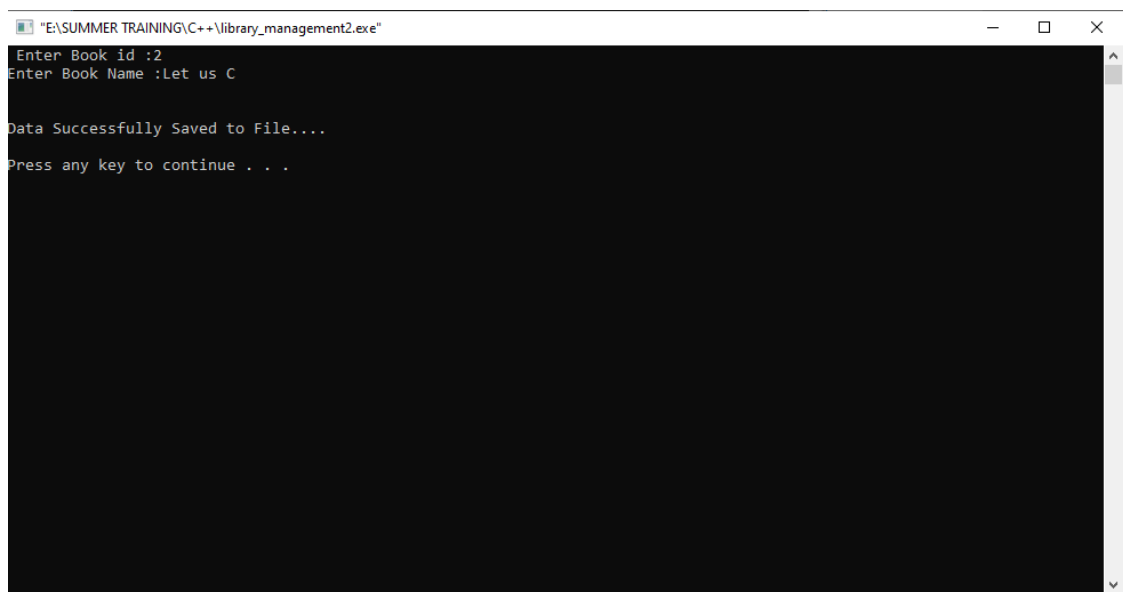
ADD BOOK



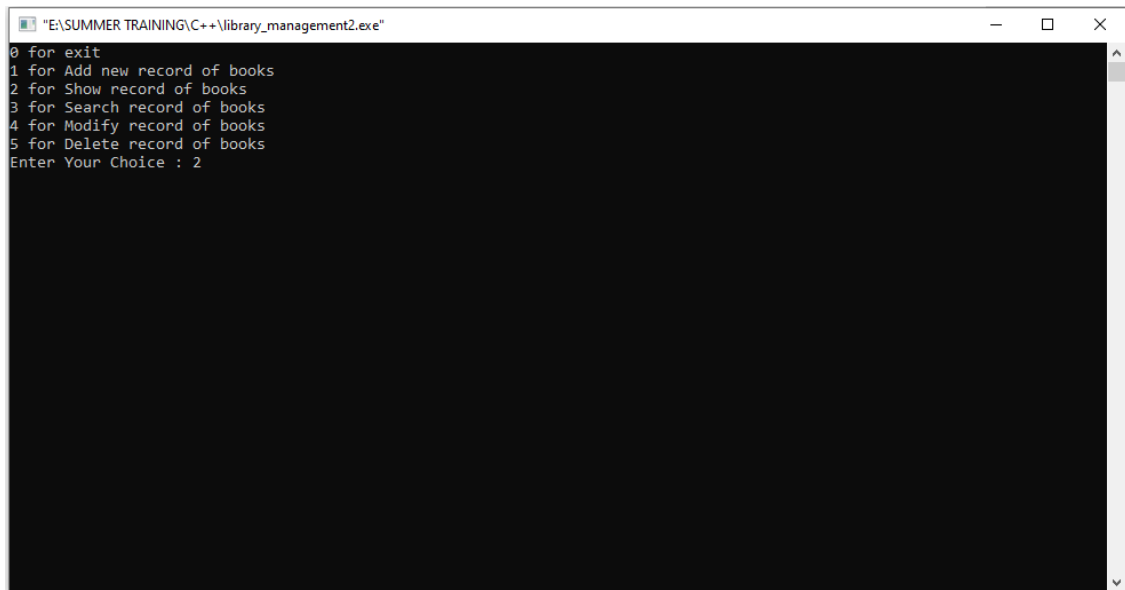
```
"E:\SUMMER TRAINING\C++\library_management2.exe"
0 for exit
1 for Add new record of books
2 for Show record of books
3 for Search record of books
4 for Modify record of books
5 for Delete record of books
Enter Your Choice : 1
```



ADD SUCCESS

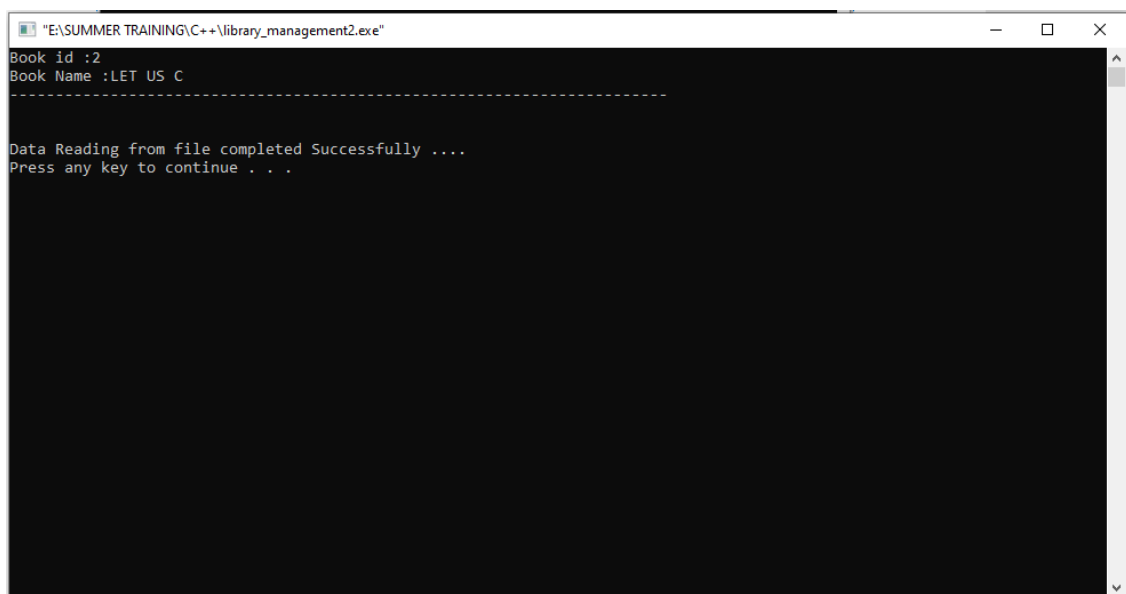


SHOW ALL BOOKS



```
"E:\SUMMER TRAINING\C++\library_management2.exe"
0 for exit
1 for Add new record of books
2 for Show record of books
3 for Search record of books
4 for Modify record of books
5 for Delete record of books
Enter Your Choice : 2
```

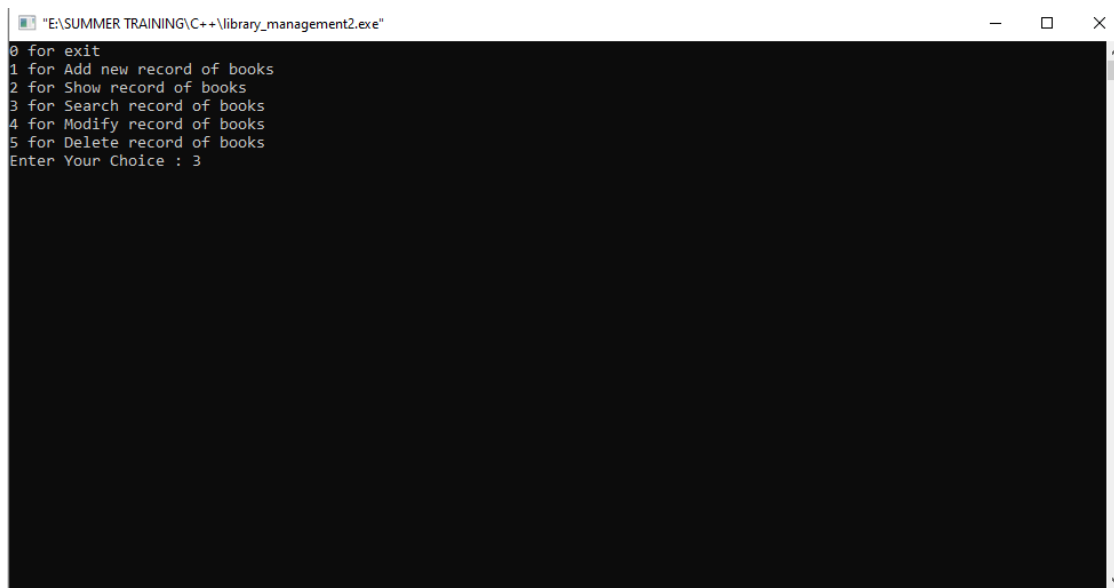
SHOW SUCCESS



```
"E:\SUMMER TRAINING\C++\library_management2.exe"
Book id :2
Book Name :LET US C
-----

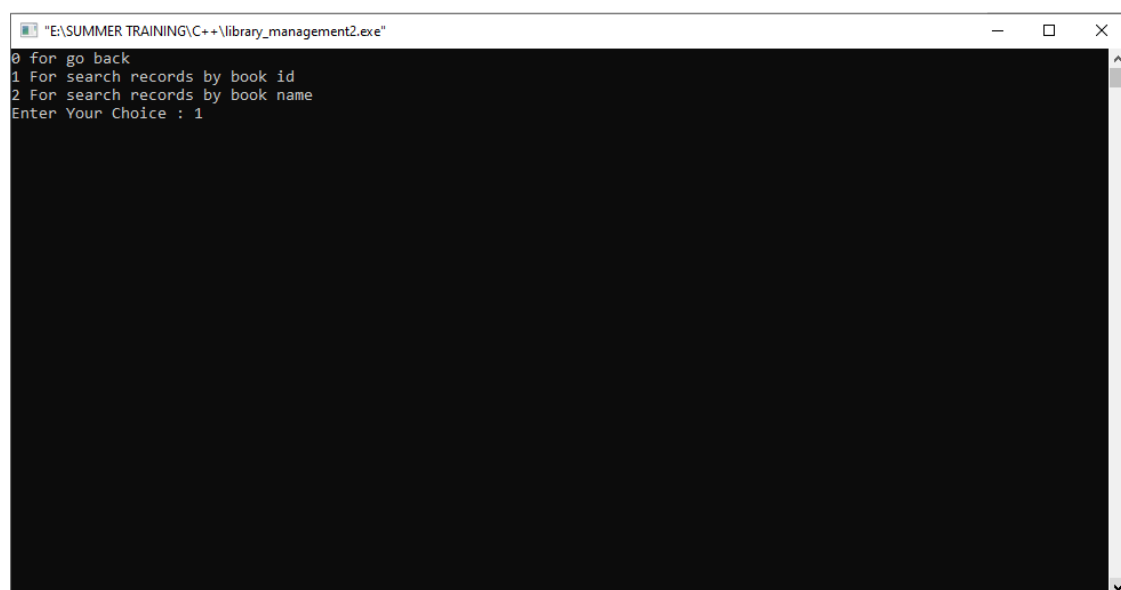
Data Reading from file completed Successfully ....
Press any key to continue . . .
```

SEARCH BOOK



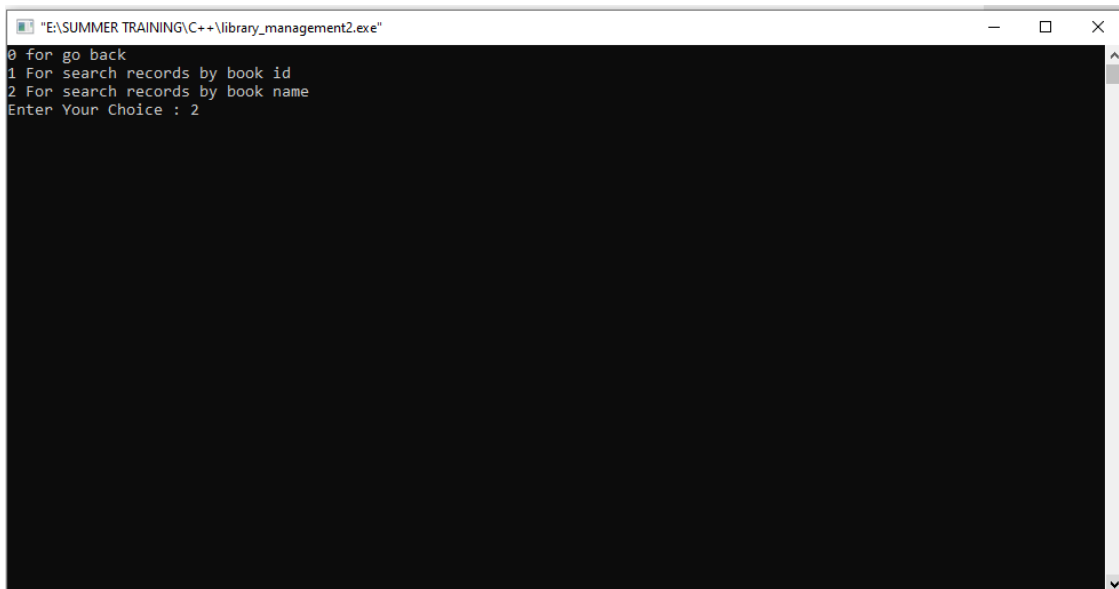
```
"E:\SUMMER TRAINING\C++\library_management2.exe"
0 for exit
1 for Add new record of books
2 for Show record of books
3 for Search record of books
4 for Modify record of books
5 for Delete record of books
Enter Your Choice : 3
```

SEARCH BY ID



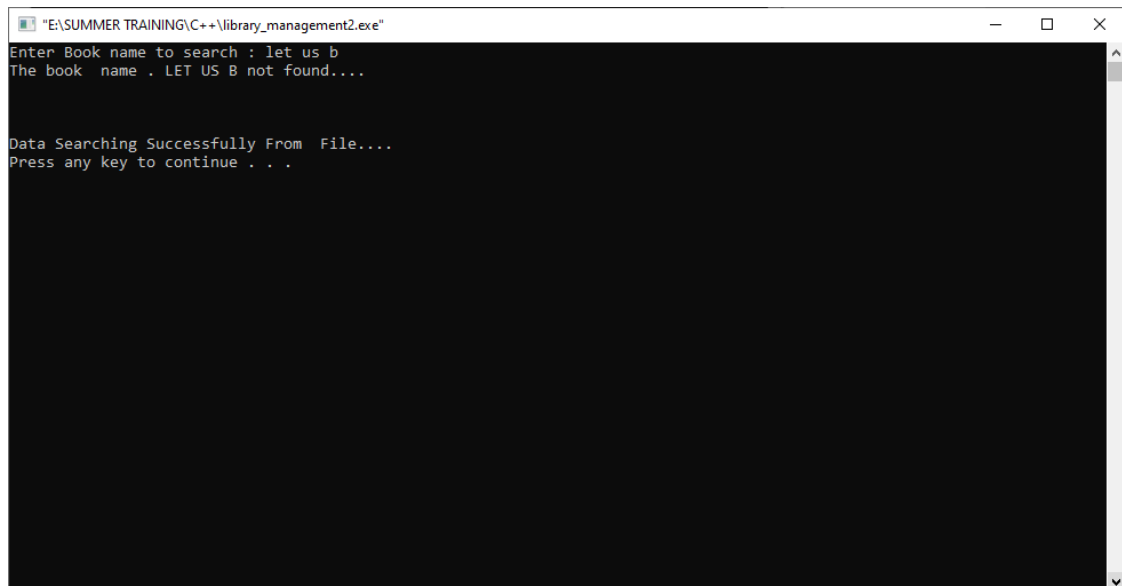
```
"E:\SUMMER TRAINING\C++\library_management2.exe"
0 for go back
1 For search records by book id
2 For search records by book name
Enter Your Choice : 1
```

SEARCH BY NAME



```
"E:\SUMMER TRAINING\C++\library_management2.exe"
0 for go back
1 For search records by book id
2 For search records by book name
Enter Your Choice : 2
```

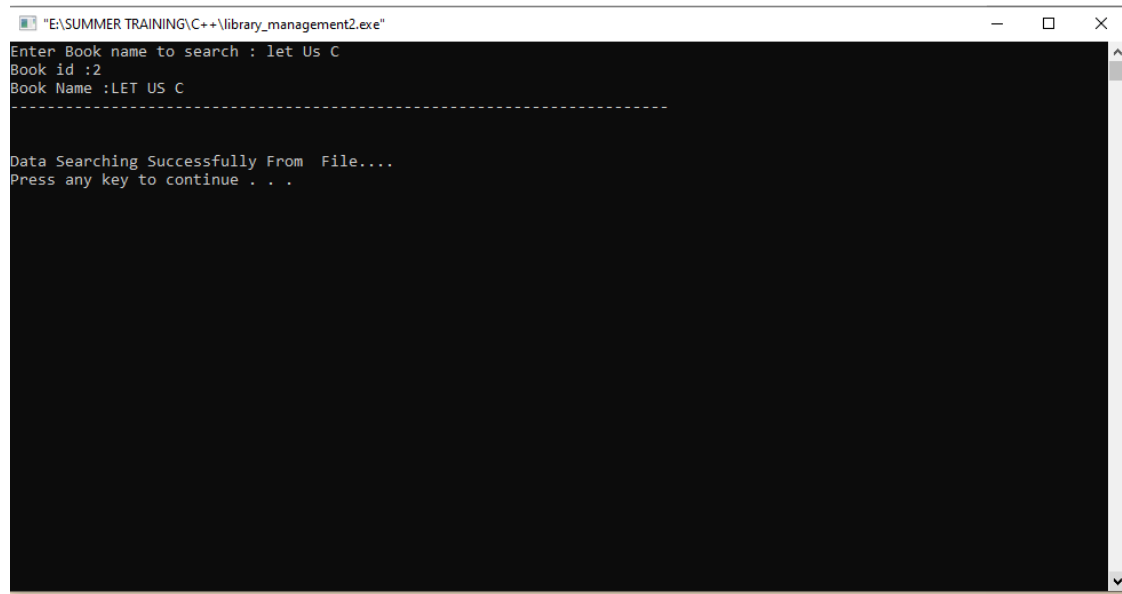
NOT FOUND



```
"E:\SUMMER TRAINING\C++\library_management2.exe"
Enter Book name to search : let us b
The book name . LET US B not found....

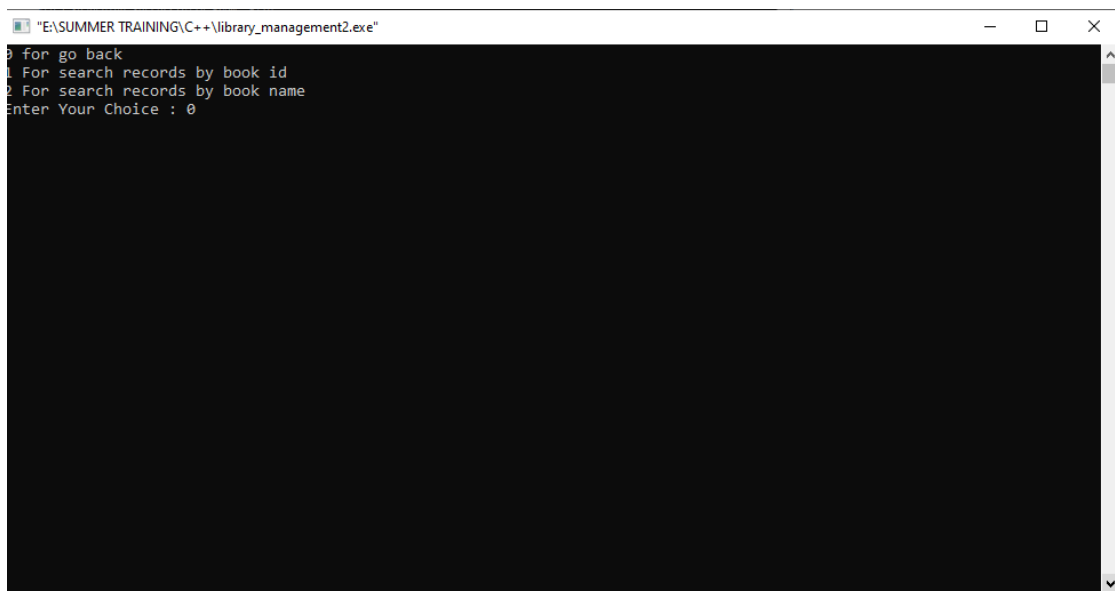
Data Searching Successfully From File....
Press any key to continue . . .
```


SEARCH BY NAME SUCESS



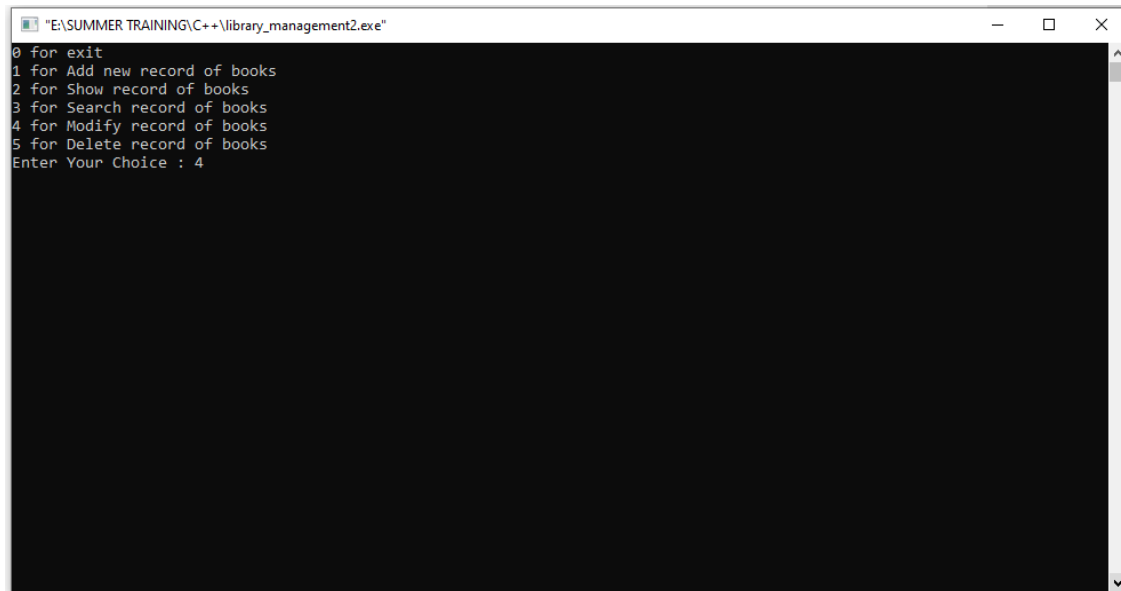
```
"E:\SUMMER TRAINING\C++\library_management2.exe"
Enter Book name to search : let Us C
Book id :2
Book Name :LET US C
-----
Data Searching Successfully From File....
Press any key to continue . . .
```

GO BACK BUTTON



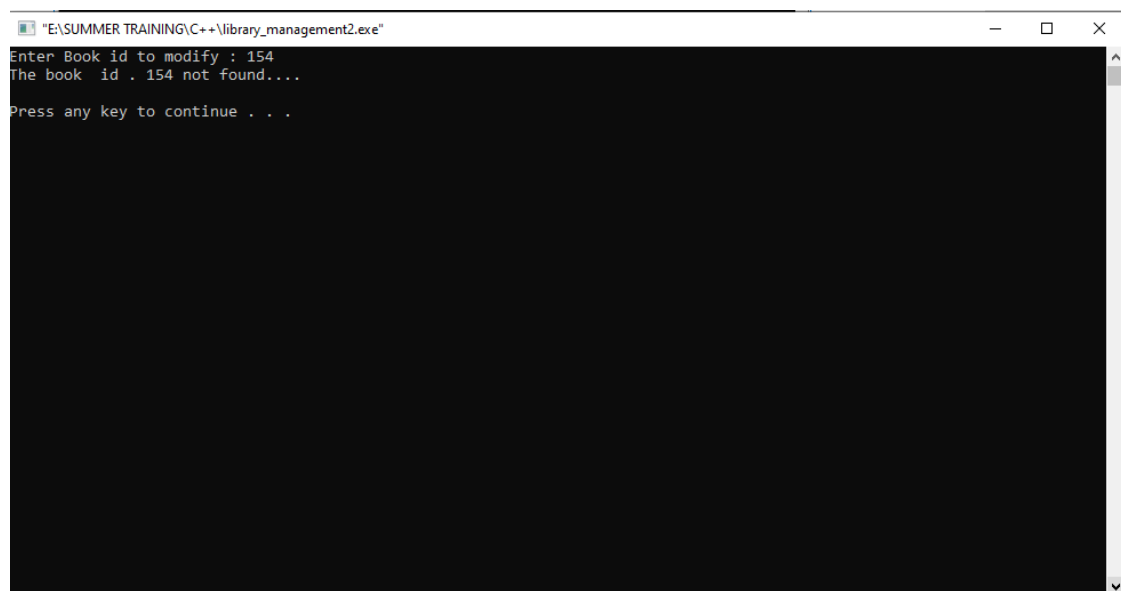
```
"E:\SUMMER TRAINING\C++\library_management2.exe"
0 for go back
1 For search records by book id
2 For search records by book name
Enter Your Choice : 0
```

UPDATE BOOK



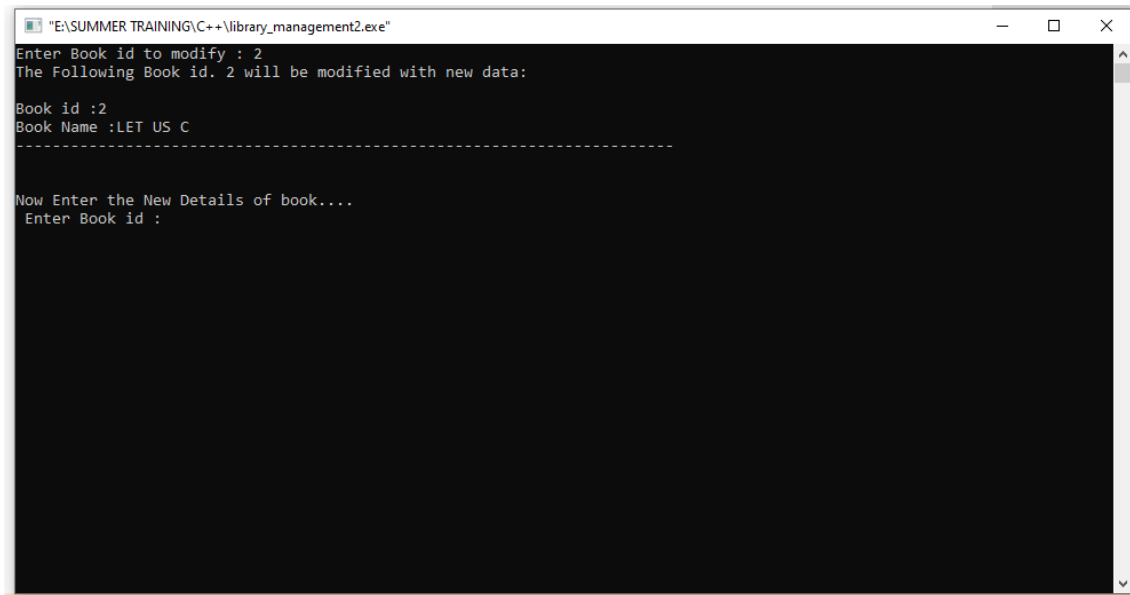
```
"E:\SUMMER TRAINING\C++\library_management2.exe"
0 for exit
1 for Add new record of books
2 for Show record of books
3 for Search record of books
4 for Modify record of books
5 for Delete record of books
Enter Your Choice : 4
```

ERROR HANDLING



```
"E:\SUMMER TRAINING\C++\library_management2.exe"
Enter Book id to modify : 154
The book id . 154 not found....
Press any key to continue . . .
```

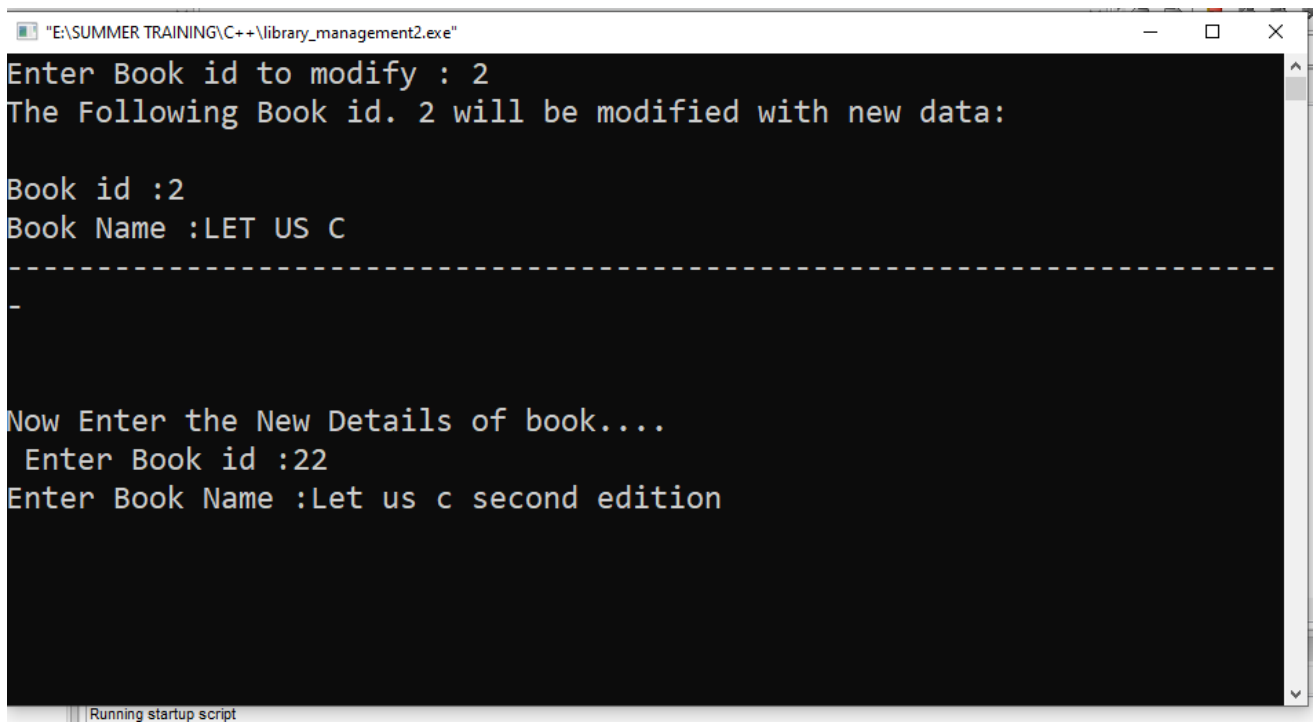
UPDATE



```
"E:\SUMMER TRAINING\C++\library_management2.exe"
Enter Book id to modify : 2
The Following Book id. 2 will be modified with new data:

Book id :2
Book Name :LET US C
-----

Now Enter the New Details of book....
Enter Book id :
```



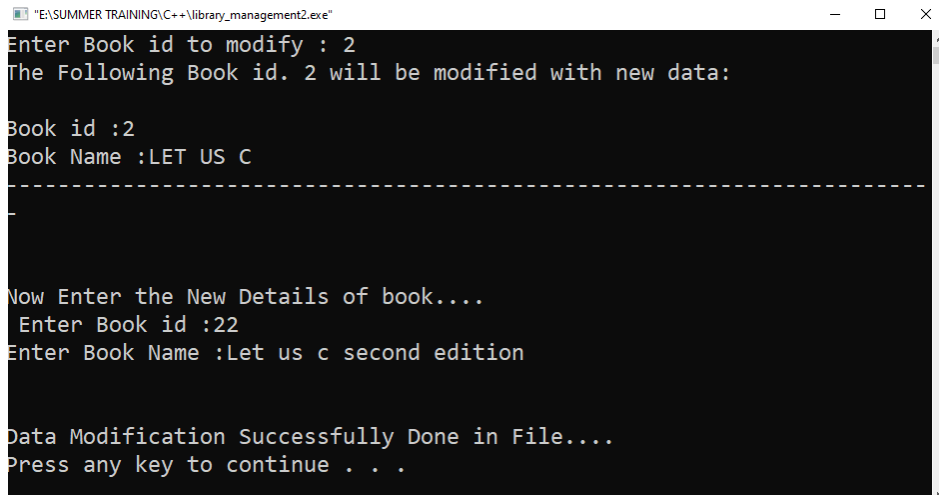
```
"E:\SUMMER TRAINING\C++\library_management2.exe"
Enter Book id to modify : 2
The Following Book id. 2 will be modified with new data:

Book id :2
Book Name :LET US C
-----

Now Enter the New Details of book....
Enter Book id :22
Enter Book Name :Let us c second edition
```

Running startup script

UPDATE SUCCESS



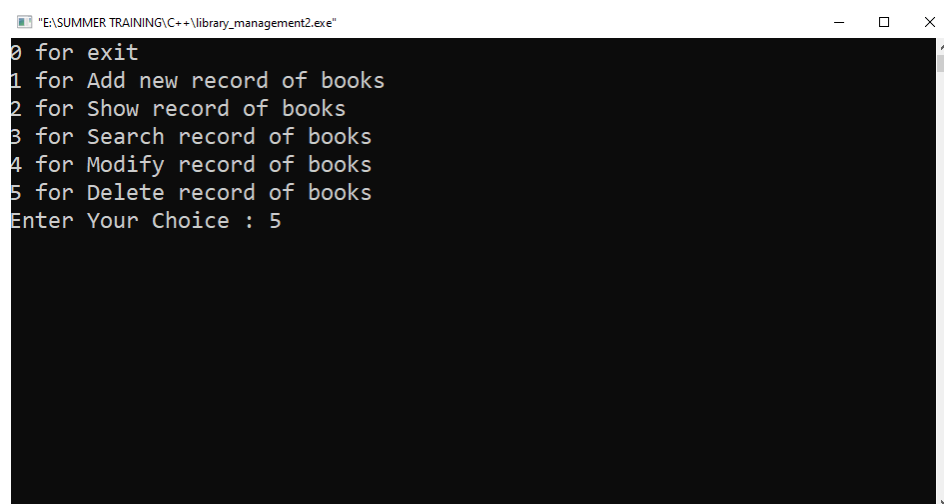
```
"E:\SUMMER TRAINING\C++\library_management2.exe"
Enter Book id to modify : 2
The Following Book id. 2 will be modified with new data:

Book id :2
Book Name :LET US C
-----

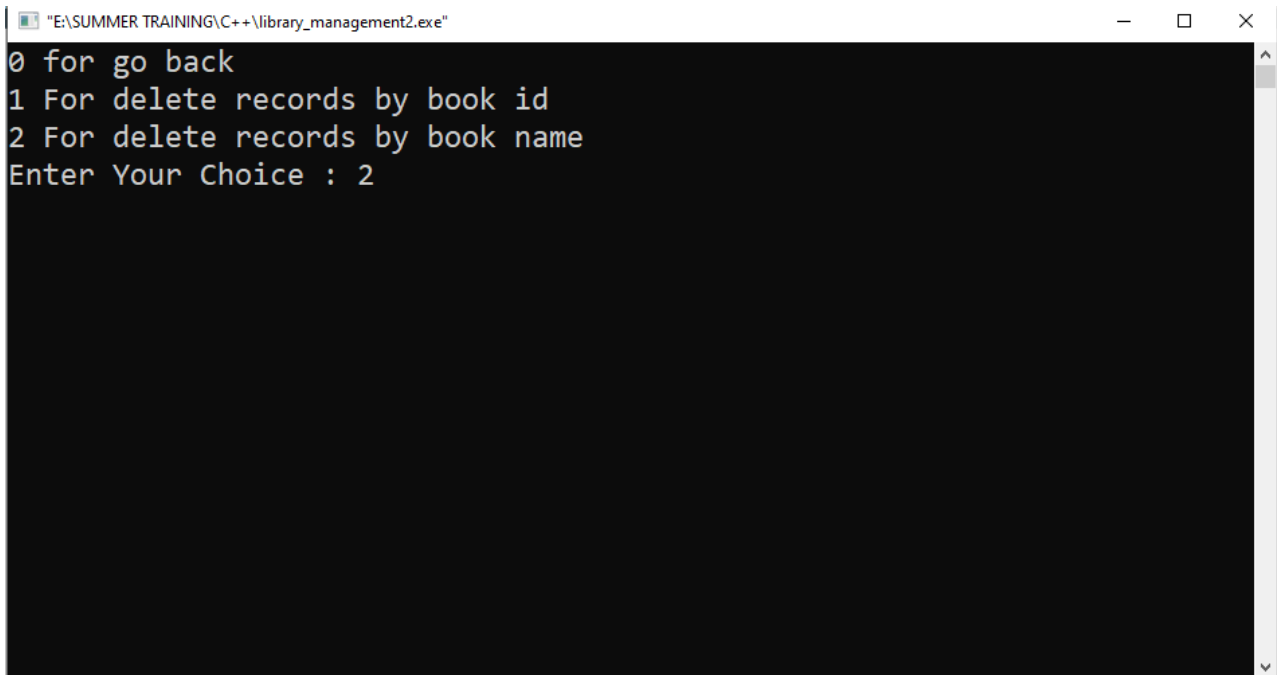
Now Enter the New Details of book....
Enter Book id :22
Enter Book Name :Let us c second edition

Data Modification Successfully Done in File....
Press any key to continue . . .
```

DELETE BOOK

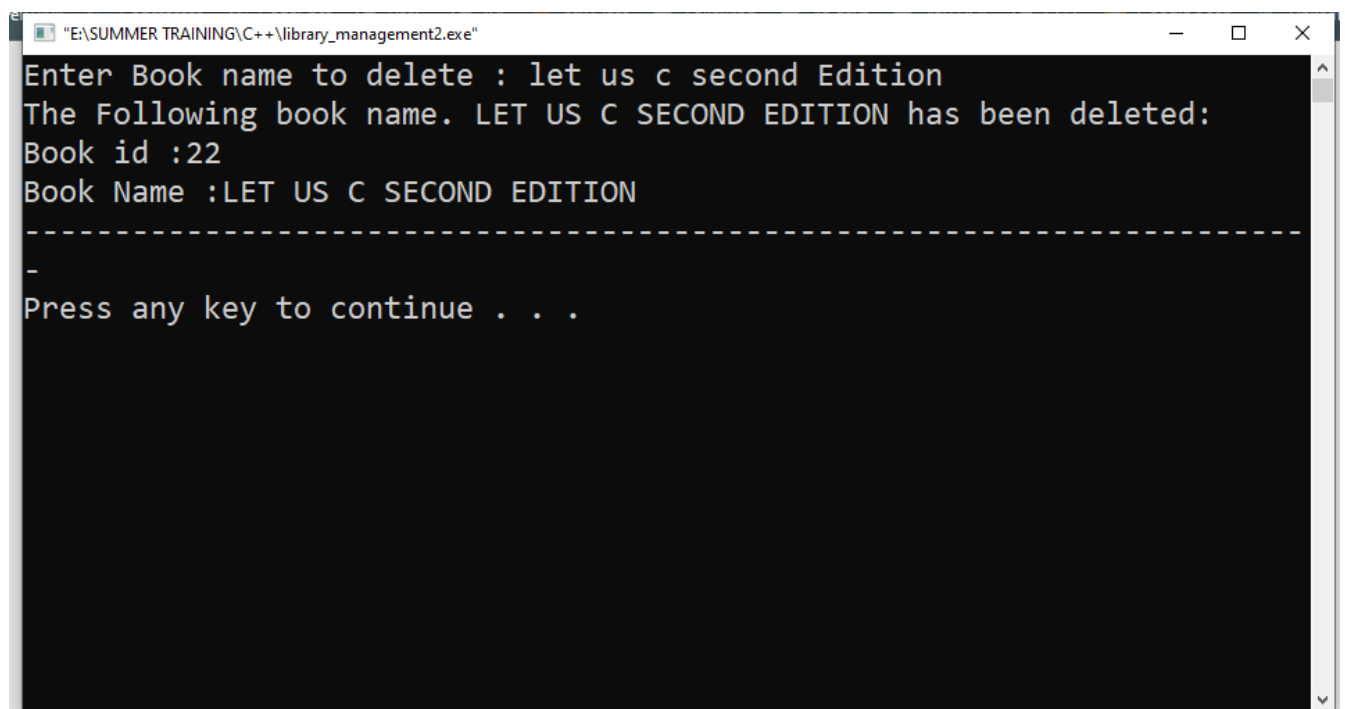


```
"E:\SUMMER TRAINING\C++\library_management2.exe"
0 for exit
1 for Add new record of books
2 for Show record of books
3 for Search record of books
4 for Modify record of books
5 for Delete record of books
Enter Your Choice : 5
```



```
"E:\SUMMER TRAINING\C++\library_management2.exe"
0 for go back
1 For delete records by book id
2 For delete records by book name
Enter Your Choice : 2
```

DELETE SUCCESS



```
"E:\SUMMER TRAINING\C++\library_management2.exe"
Enter Book name to delete : let us c second Edition
The Following book name. LET US C SECOND EDITION has been deleted:
Book id :22
Book Name :LET US C SECOND EDITION
-----
Press any key to continue . . .
```

Learning Outcomes from training/technology learnt

This project will be able to be understanding the way that how C++ code works after making this programming as well as I learned about data structures and algorithms and also get to know about all object oriented programming concepts.

Project Legacy

- > I learned to always prepare ahead for team meetings by having the goal and the objective of the meeting in mind. I also planned an agenda and questions that can be asked during meeting as well. It helped in getting value out of team meetings.
- > And, I learned to give actionable, specific and kind feedback and to highlight what the programmers did well each week and also, I could improve on.
- > I learned to always try to find out what is going on with my team members asides work and being patient with each and everyone.
- > I learned how to schedule my day and manage my time effectively by always planning my day ahead and sticking to the plan for the day. I was more productive and accomplished more of my goals this way.

Bibliography

- In my opinion, I have successfully completed my project by taking the help from online websites and from YouTube.
- Mainly I had taken help of faculty from the Fifth Force Academy, by his lecture course