NEW HORIZON COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade. Accredited by NBA

nirf TOP ENGINEERING COLLEGE OF INDIA as per NATIONAL INSTITUTIONAL RANKING FRAMEWORK, 2018
Ministry of Human Resource Development, Government of India

A

MINI PROJECT REPORT

ON

"Book Store Maintenance"

Submitted in the partial fulfillment of the requirements in the IV semester of

BACHELOR OF ENGINEERING

IN

INFORMATION SCIENCE AND ENGINEERING

BY

Varna Murali - [1NH17IS126]

FOR

COURSE NAME: MINI PROJECT

COURSE CODE: ISE46

Under the guidance of

Prof. Prakruthi S.T
Assistant Professor,
Dept. of ISE, NHCE

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

NEW HORIZON COLLEGE OF ENGINEERING

(Autonomous College Permanently Affiliated to VTU, Approved by AICTE, Accredited by NAAC with 'A' Grade & NBA)
Ring Road, Bellandur Post, Near Marathahalli,
Bengaluru-560103, INDIA

NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College Permanently Affiliated to VTU, Approved by AICTE & UGC
Accredited by NAAC with 'A' Grade. Accredited by NBA
nirf TOP ENGINEERING COLLEGE OF INDIA as per
NATIONAL INSTITUTIONAL RANKING FRAMEWORK, 2018
Ministry of Human Resource Development, Government of India

CERTIFICATE

Certified that the project work entitled "Book Store Maintenance System" carried out by Mr./Ms. Varna Murali, USN 1NH17IS126, a bonafied student of IV sem in partial fulfillment for the award of Bachelor of Engineering in Information Science and Engineering of the Visvesvaraya Technological University, Belgaum during the year 2018-19. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

Signature of the Guide	Signature of the HOD	Signature of the Principal
[Guide Name]	Dr. R.J Anandhi	Dr.Manjunatha

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. A number of personalities, in their own capacities have helped me in carrying out this mini project. I would like to take this opportunity to thank them all.

I thank the management, **Dr. Mohan Manghnani,** Chairman, New Horizon Educational Institutions for providing necessary infrastructure and creating conducive environment for effective learning.

I also record here the constant encouragement, support and facilities extended to us by **Dr. Manjunatha,** Principal, New Horizon College of Engineering, Bengaluru.

I extent sincere gratitude for constant encouragement and facilities provided to us by **Dr. R.J Anandhi**, Professor and Head of the Department, Department of Information Science and Engineering, New Horizon College of Engineering, Bengaluru.

I sincerely acknowledge the encouragement, timely help and guidance to me by **Prof. Prakruthi S T**, Assistant Professor, Department of Information Science and Engineering, New Horizon College of Engineering, Bengaluru, to complete the mini project within stipulated time successfully.

Finally, a note of thanks to the teaching and non-teaching staff of Information Science and Engineering Department for their cooperation extended to us and our friends, who helped me directly or indirectly in the successful completion of this mini project.

Varna Murali

TABLE OF COTENTS

Abstract	i
Acknowledgement	ii
Table of Contents	iii
Chapters	**Page Number**
Chapter 1: Introduction	
1.1Motivation of Project	02
1.2 Problem Statement	02
Chapter 2: System Requirement Specification	
2.1Hardware System Configuration	08-10
2.2Software System Configuration	08-10
Chapter 3: Methodology	
3.1 Module	11-12
3.2 Flowchart	13-15
3.3 Algorithm	16-17
3.4 Code Implementation	17-23
Chapter 4: Results and Discussion	
4.1 Use of table and explanation	23
4.2 Output (Snapshots)	24-33
Conclusion and Future Enhancements	34
References	35

ABSTRACT

Today it is becoming very difficult to maintain records manually. Software system easily does the job of maintaining daily records as well as the transaction according to the user requirements. Only basic knowledge of computers is required for operations. The software system consists of all information of books and sold to the customer. The proposed system provides lots of facility to the user to store information of the books and it provide information in quick time in a systematic manner. The processing time on the data is very fast. It provides required data quickly to the user and also in specified manner to the user. There is lot of duplicate woks, and chance of mistake when the records are changed they need to update each and every excel file. There is no option to find and print previous saved records there is any security; anybody can access any report and sensitive data. This bookshop management system is used to overcome entire problem which they are facing currently, and making complete atomization of manual system to computerized system.

TABLE OF FIGURES

FIGURE NUMBER	FIGURE NAME	PAGE NUMBER
Figure 1.1	Polymorphism	5
Figure 1.2	Polymorphism	7
Figure 1.3	Inheritance	7
Figure 3.2	Flowchart	13-15
Figure 4.1.1	Output Snapshot 1	24
Figure 4.1.2	Output Snapshot 2	25
Figure 4.1.3	Output Snapshot 3	26
Figure 4.1.4	Output Snapshot4	27
Figure 4.1.5	Output Snapshot5	28
Figure 4.1.6	Output Snapshot6	29
Figure 4.1.7	Output Snapshot 7	30
Figure 4.1.8	Output Snapshot 8	31
Figure 4.1.9	Output Snapshot 9	32
Figure 4.1.10	Output Snapshot10	33

CHAPTER 1

INTRODCTION

It is an online web application where the customer can purchase book online. This software is developed to maintain records of sales, purchase and staff records. The online book shop website provides customer with online shopping through a web browser. The purpose of this software is to manage the books in the bookshop. It includes the order processing, stock management and we developed this software to maintain records of sales, purchase. Here, we try to develop this system which is providing the automation on the any type of the bookshop. In this system all records are saved in the database for report generation. In present system during issuing order of more stock, the product register is required to check to availability of stock in hand, and it takes time to check records. In each process whether it is product management, maintaining customer records, payment management, report generation, user has to pay attention to a greater extent while performing the tasks. Book shop management system should help the customers query whether a book in a stock the user can query the availability of a book either by using the book title. Bookshop management software for monitoring and controlling the transactions in a bookshop. Book Store maintenance is a project in which people can search books, buy books, details of the books. Admin can add new books to the store whenever required. It is a simple and user friendly project.

Book store provides quality information about books. It provides user with catalog of different books available for purchase in the store. The user can choose a book based on its views. He can check the list of books present in the book store and choose the book of his choice.

Book store can lead to error free and fast management system. It reduces manual effort. Searching a book becomes easier as the book can be searched by its name. And all the details of the book along with the number of views of that book will be displayed. The user can choose a book based on the number of views. It is a more efficient way of maintaining books.

1.1Motivation of Project

Book Store Maintenance is the system in which all the details of the book are computerized which will reduce manual effort and will save time. It is a more efficient way of maintaining records.

When a customer asks for a book, the book should be manually searched and there are chances that the book may not be there in the store. In this system a book can be searched easily by entering its name and even the number of views of that particular book will be displayed which will help to user to choose a book without confusion. New books can be added to the store. This is more efficient way of maintaining books.

The above mentioned problem can be reduced by making it an online system. It reduces time, is user friendly and reduces manual effort.

1.5 Problem Statement:

The aim of the project is to develop a system that is meant to computerize the work like maintaining the details of the book, adding new book to the store, searching the details of the book and displaying the details. This reduces the manual effort of searching a book and saves time. This is more methodical way of maintaining books.

1.2 Introduction to oops concepts

Object oriented programming is based on the concept of "objects", which contains data in the form of fields and code in the form of procedures. In oop computer programs are designed by making them out of objects that interact with one another. Oop languages are divers, but most popular ones are class-based, meaning the object is instance of classes which also determine their types.

Many of the most widely used programming languages are multi-paradigm and they support object-oriented programming to a greater or lesser degree, typically in combination with imperative, procedural programming. The core of the pure object oriented programming is to create an object, in code, that has certain properties and methods.

Class:

In object oriented programming, a class a collection of similar objects. It is a blue print of data and functions. The complete object can be defined in a single datatype called class. It is a blue print or set of instructions to build a specific type if object.

Syntax: class class_name

```
{
  Private: data members;
  Public: member functions;
 }
```

Object:

An object is nothing but a self-contained component which consists of methods and properties to make a particular type of data useful. Object determines the behavior of the class. When you send a message to an object, you are asking the object to invoke or execute one of its methods. 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.

Declaring Objects: When a class is defined, only the specification for the object is defined no memory or storage is allocated. To use the data and access functions defined in the class, you need to create objects.

Syntax: classname objectname;

Encapsulation:

The process of wrapping up of data and functions into a single unit called class is encapsulation. This prevents direct access of data; the access to them is given through functions of the class. It helps in data hiding.

How Encapsulation is achieved in a class
To do this:
1) Make all the data members private.
2) Create public setter and getter functions for each data member in such a way that the set function set the value of data member and get function get the value of data member. The finance section handles all the financial transactions and keeps records of all the data related to finance. Similarly the sales section handles all the sales related activities and keeps records of all the sales. Now there may arise a situation when for some reason an official from finance section needs all the data about sales in a particular month. In this case, he is not allowed to directly access the data of sales section. He will first have to contact some other officer in the sales section and then request him to give the particular data. This is what encapsulation is. Here the data of sales section and the employees that can manipulate them are wrapped under a single name "sales section".

Abstraction:

Data abstraction refers to the act of providing essential details without including the background details. Class is called abstract datatype because it has the feature of data abstraction. The advantage of abstraction is we can change implementation at any point, users of complex class won't be affected as out method interface remains the same. Data abstraction is a programming (and design) technique that relies on the separation of interface and

implementation. Let's take one real life example of a TV, which you can turn on and off, change the channel, adjust the volume, and add external components such as speakers, VCRs, and DVD players, BUT you do not know its internal details, that is, you do not know how it receives signals over the air or through a cable, how it translates them, and finally displays them on the screen.

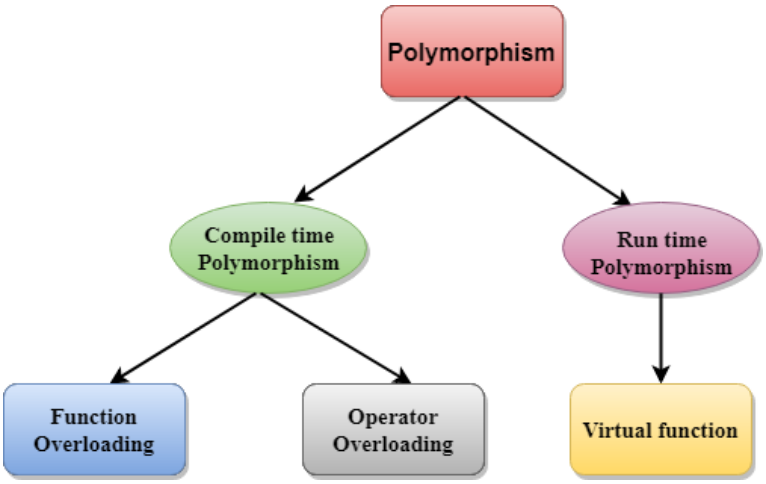
Thus, we can say a television clearly separates its internal implementation from its external interface and you can play with its interfaces like the power button, channel changer, and volume control without having any knowledge of its internals.

Polymorphism:

Ability to take more than one form is called polymorphism. Polymorphism can be implemented in two ways: Function overloading and operator overloading. Polymorphism occurs when there is a hierarchy of classes and they are related by inheritance.

Function overloading: when two or more functions have the same name but differ in type and number of arguments is called function overloading. A function is choosed based on the number or type of argument.

Operator overloading: The process of making an operator to work on operands of different types is called operator overloading.



1.1 Polymorphism

Inheritance:

Inheritance is one of the key features of object oriented programming in c++. It allows user to create a new class (derived class) from an existing class(base class). The derived class inherits all the features of the base class and can have additional features of its own. There are many types of inheritance. Since, all of the characters are persons, they can walk and talk. However, they also have some special skills. If you want to add a new feature - eat, you need to implement the same code for each character. This can easily become error prone (when copying) and duplicate codes.

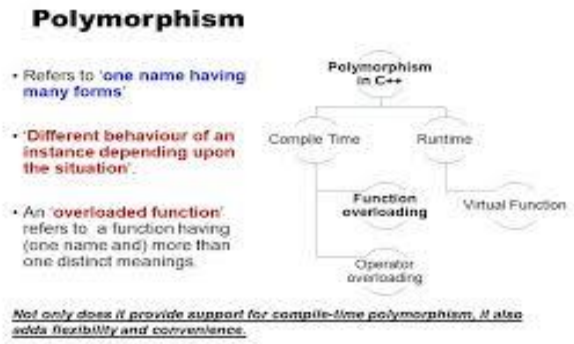
It'd be a lot easier if we had a Person class with basic features like talk, walk, eat, sleep, and add special skills to those features as per our characters. This is done using inheritance.

Using inheritance, now you don't implement the same code for walk and talk for each class. You just need to inherit them.

So, for Math's teacher (derived class), you inherit all features of a Person (base class) and add a new feature TeachMaths. Likewise, for a footballer, you inherit all the features of a Person and add a new feature Play Football and so on. This makes your code cleaner, understandable and extendable.

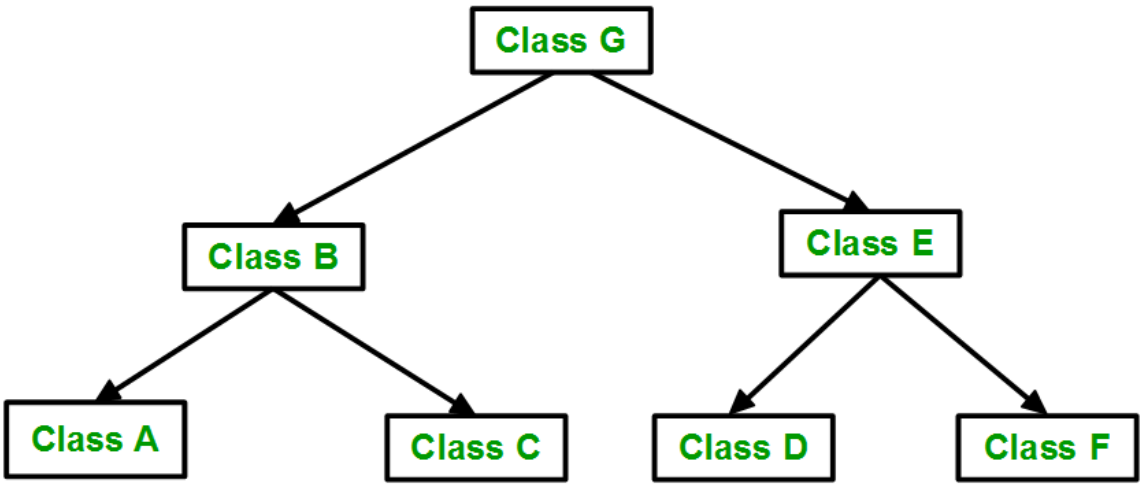
1.3To implement the proposed system polymorphism and inheritance are used

Polymorphism: It helps programmers reuse the code and classes once written, tested and implemented. They can be reused in many ways. Single variable name can be used to store variables of multiple datatypes. The length of the code will be reduced. It helps programmers reuse the code and classes once written, tested and implemented. They can be reused in many ways. Single variable name can be used to store variables of multiple data types (Float, double, Long, Int etc.).Polymorphism helps in reducing the coupling between different functionalities.



1.2 Figure on polymorphism

- Inheritance: The most frequent use of inheritance is for deriving classes using existing classes using existing classes, which provides code reusability. The existing classes remain unchanged. By reusability, the development time of software is reduced. One of the key benefits of inheritance is to minimize the amount of duplicate code in an application by sharing common code amongst several subclasses. Where equivalent code exists in two related classes, the hierarchy can usually be refactored to move the common code up to a mutual superclass. This also tends to result in a better organization of code and smaller, simpler compilation units.
- Inheritance can also make application code more flexible to change because classes that inherit from a common superclass can be used interchangeably. If the return type of a method is superclass



1.3 Inheritance

Chapter 2

SYSTEM REQUIREMENTS SPECIFICATION

2.1 Hardware system configuration

Processor: Intel Core i3-380M dual-core processor

Ram: 1GB (32 bit) o2 3GB (64 bit)

Hard disk: 16GB

2.2 Software requirements

Operating System - windows

Programming Language - C++

Compiler - Code blocks

Code blocks Is a free, open-source cross-platform IDE that supports multiple compilers including GCC, Clang and Visual C++. It is developed in C++ using wxWidgets as the GUI toolkit. Using plugin architecture, its capabilities and features are defined by the provided plugins. Currently Code::Blocks is oriented towards C, C++ and FORTRAN. IT has a custom build system and Make support. It has custom build system and optional.

Code::Blocks supports multiple compilers, including GCC, MinGW, Digital Mars, Microsoft Visual C++, Borland C++, LLVM Clang, Watcom, LCC and the Intel C++ compiler. Although the IDE was designed for the C++ language, there is some support for other languages, including FORTRAN and D. A plug-in system is included to support other programming languages.

The Code::Blocks debugger has full breakpoint support. It also allows the user to debug their program by having access to the local function symbol and argument display, user-defined watches, call stack, disassembly, custom memory dump, thread switching, CPU registers and GNU Debugger Interface.

2.3 Programming Language:

C++ is a general purpose programming language that was developed by Bjarne Strostrup as an extension of C language, or C with classes. It has imperative, object-oriented and generic programming features, while also providing facilities of low-level memory manipulation. C++ is standardized by International Organization for Standardization with the latest standard version ratified and published by ISO. The motivation for creating a new language originated from Strostrup's experience in programming for his Ph.D. thesis.

Throughout C++'s life, its development and evolution has been guided by a set of principles:

- It must be driven by actual problems and its features should be immediately useful in real world programs.
- Every feature should be implementable (with a reasonably obvious way to do so).
- Programmers should be free to pick their own programming style, and that style should be fully supported by C++.
- Allowing a useful feature is more important than preventing every possible misuse of C++.
- It should provide facilities for organizing programs into separate, well-defined parts, and provide facilities for combining separately developed parts.
- No implicit violations of the type system (but allow explicit violations; that is, those explicitly requested by the programmer).
- User-created types need to have the same support and performance as built-in types.
- Unused features should not negatively impact created executable (e.g. in lower performance).

- There should be no language beneath C++ (except assembly language).
- C++ should work alongside other existing programming languages, rather than fostering its own separate and incompatible programming environment.
- If the programmer's intent is unknown, allow the programmer to specify it by providing manual control.

Features of c++

- Objects
- Data Encapsulation which ensures data hiding
- Inheritance which ensures code reusability
- C++ is case sensitive
- It is platform dependent
- Huge library function
- Fast and efficient

Chapter 3

SYSTEM MODULE

3.1 Module

- Void display ();

  User-defined function

  To display the title and price of the books

- Void increaseviews();

  To increase the number of views of the book when it is searched for

- Void book ::display();

  To display author views and publication

- Displayall();

  To display all the available books in the book store

- Void search();

  Enter the title

  If found call increaseviews()

  Else

  Display entry not found

- Add();

  Enter title

  Enter author name

  Enter publications

Enter price

- Book();

  It contains a this pointer which points to authors and publications

  It has views initialized to zero

- Mainscr();

  Switch case

  Case1:

  Enter the title, publication, author and price

  Case 2:

  To search for a book

  Enter the title of the book

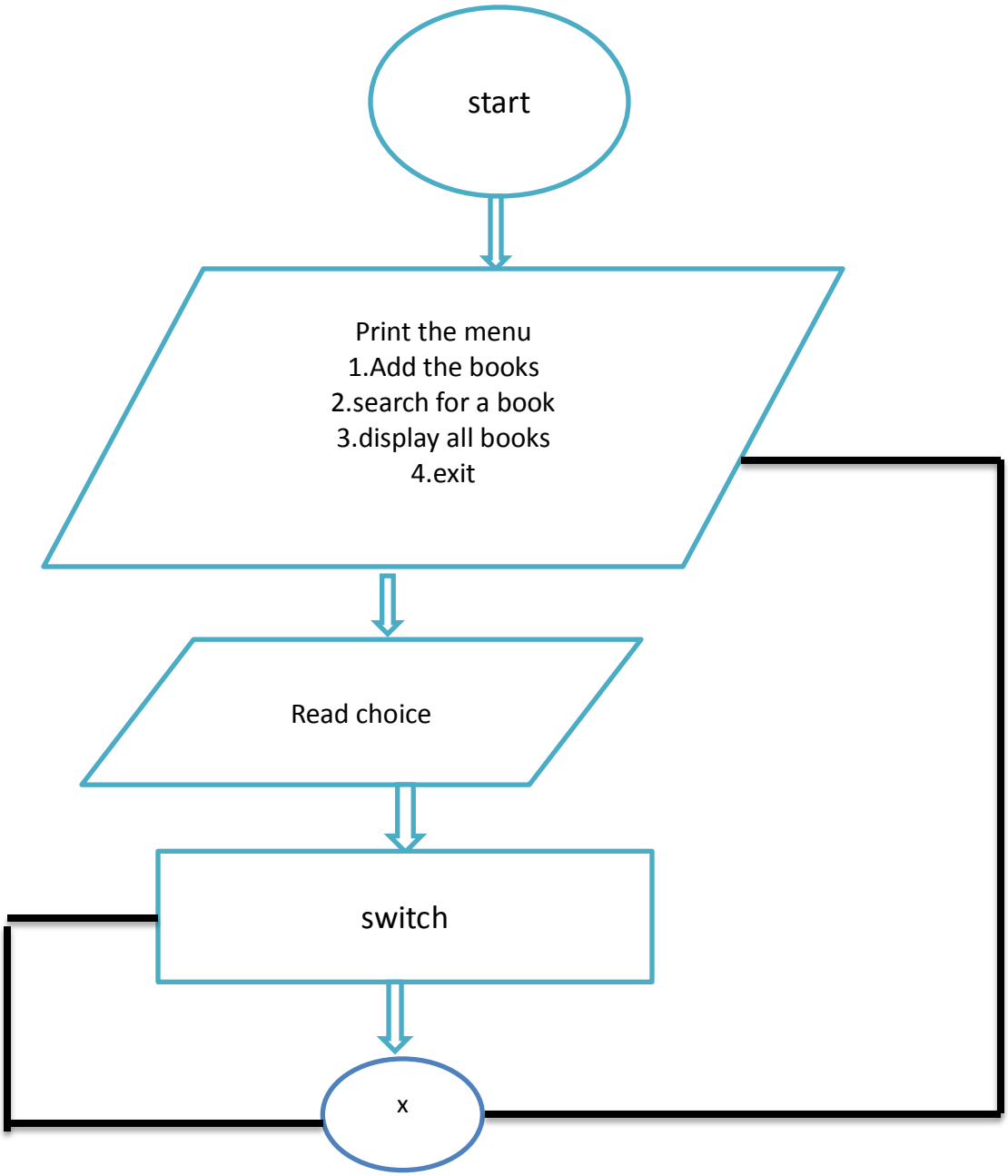
  Case 3:

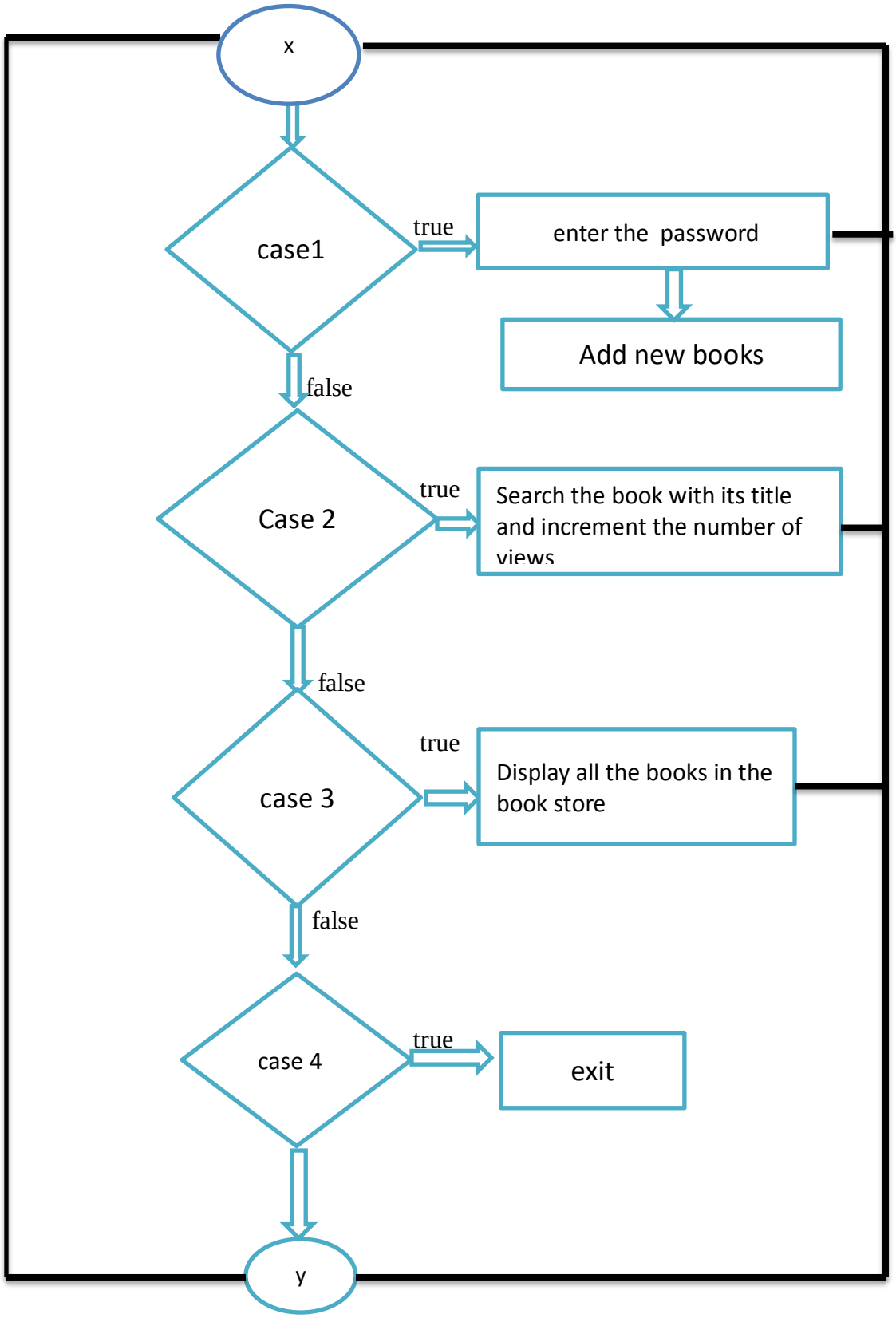
  To display all the books present the book store

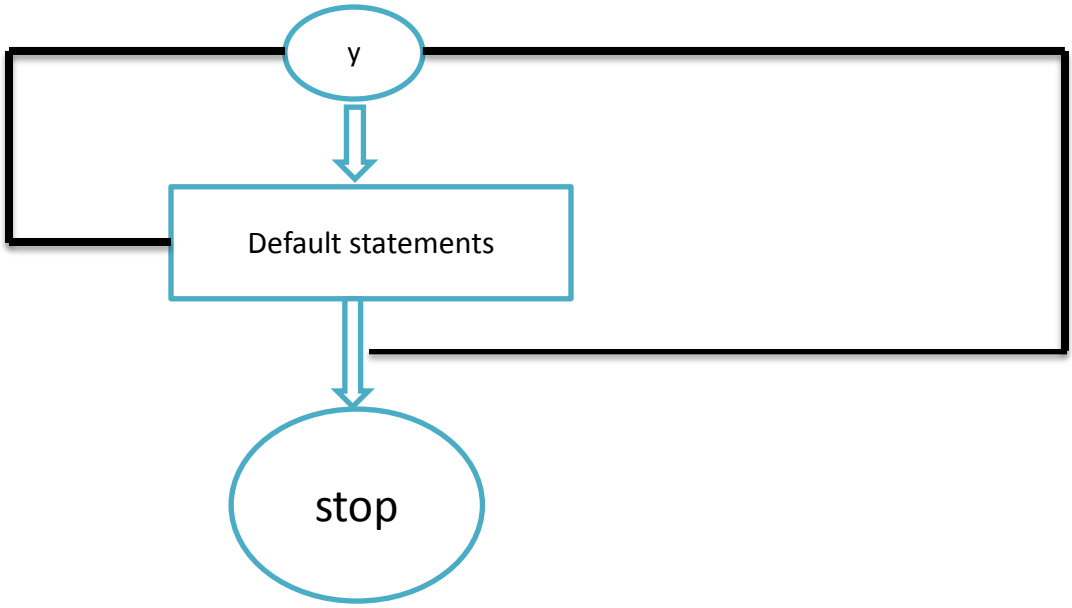
  Case 4:

  To exit

3.2 Flow Chart




x
case1
true
enter the password
Add new books
false
Case 2
true
Search the book with its title and increment the number of views
false
case 3
true
Display all the books in the book store
false
case 4
true
exit
y



3.2 Flowchart

3.3 Algorithm

Step 1: START

Step 2: Ask the user to choose:

1. to add new book

2. to display

3. to search

4. Exit

if input 1 goto step 3

if input 2 goto step 4

if input 3 goto step 5

Step 3 : If password is true

Input details such as entering the title, author name, publication and Price .

Else

Print

Wrong password

Step 4 : input the title of the book to be searched for

If present

Increment the views of the book searched for

Else

Print entry not found

Step 5: Display all the books available in the book store

Step 6: Stop

3.4 Code and Implementation

```
#include<iostream>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

using namespace std;
static int no=0;
void clrscr()
{
        system("@cls||clear");
}
class item
{
public:
   char title[20];
int price;
public:
   item()
   {
      strcpy(title,"");
      price=0;
   }
   item(char title[20],int price)
{
      strcpy(this->title, title);            //this pointer
      this->price=price;
   }

   void Display()
   {
```

```
        cout<<"Title : "<<title<<endl;
        cout<<"Price : Rs. "<<price<<endl;
    }
};
class book: public item          //public inheritance
{
public:
        char author[20],publication[20];
        int views;

public:

inline void IncreaseViews()          //inline function
{
    ++views;
}
book()
{
    strcpy(this->author,"");
    strcpy(this->publication,"");
    views=0;
}
book(char author[20],char publication[20],char title[20],int price):item(title, price)
{
    strcpy(this->author,author);
    strcpy(this->publication,publication);
    views=0;
}
void Display();
}bk[50];
void book::Display()       //Function Overriding OR RunTime Polymorphism
```

```
{
        cout<<"**********************************************************\n";
                        cout<<"VIEWS : "<<views<<endl;
        cout<<"Author : "<<author<<endl;
                        cout<<"Publication : "<<publication<<endl;
        item::Display();
        cout<<"**********************************************************\n\n\n";
}

void Add(char auth[20],char pub[20],char titl[20],int pric)
{
   int x=no++;
 book b(auth,pub,titl,pric);
   bk[x]=b;
}
void DisplayAll()
{
   cout<<"AVAILABLE BOOKS : -\n";
   for(int i=0;i<no;i++)
   {
        cout<<bk[i].title<<endl;
   }
}

void Search(char titl[20])
{
   for(int i=0;i<no;i++)
   {
      if (stricmp(bk[i].title,titl)==0)
      {
         bk[i].IncreaseViews();
```

```
        cout<<"         ENTRY FOUND:-\n\n";
        bk[i].Display();
                return;
      }
   }
        cout<<"NO SUCH ENTRY WAS FOUND.\n\n";
   return;
}

void mainscr()
{

 char titl[20],auth[20],pub[20];int pric, ch;

 cout<<"        BOOKSTORE MAINTENANCE SYSTEM:-\n\n";
 cout<<"1. Add A Book\n";
 cout<<"2. Search For a Book\n";
 cout<<"3. List All Books\n";
 cout<<"4. Exit\n\n\n";
 cout<<"        Enter your choice : ";
 cin>>ch;
switch(ch)

 {

        case 1:

   clrscr();

        cout<<"Enter Title of the Book : "<<endl;

        cin>>titl;

        cout<<"Enter Author of the Book : "<<endl;
```

```
        cin>>auth;
    cout<<"Enter Publications of the Book : "<<endl;
        cin>>pub;
    cout<<"Enter Price of the Book : "<<endl;
        cin>>pric;
        Add(auth,pub,titl,pric);
        cout<<"\n\n\nENTRY SUCCESSFULLY ADDED.\n\n\n";
    cout<<"Enter any number to exit to main screen : ";
        cin>>ch;
        clrscr();
        mainscr();
        break;
      case 2:
        cout<<"\nEnter Title Of the Book : "<<endl;
        cin>>titl;
        Search(titl);
    cout<<"\n\nEnter any number to exit to main screen : ";
        cin>>ch;
        clrscr();
        mainscr();
```

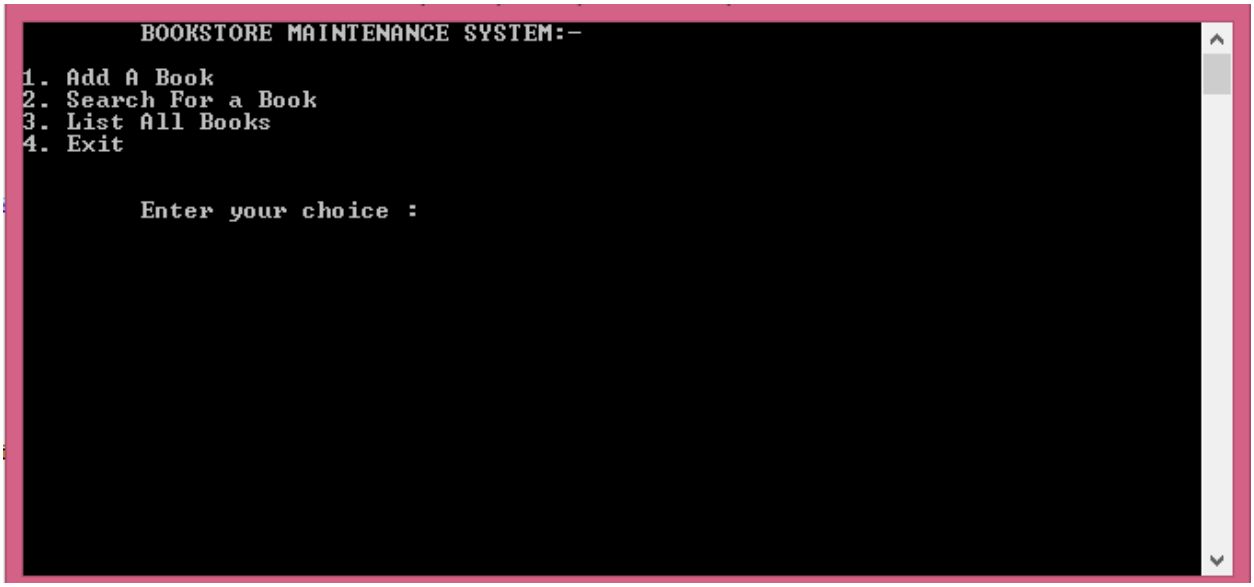
```
            break;
    case 3:
    DisplayAll();
    cout<<"\n\nEnter any number to exit to main screen : ";
            cin>>ch;
            clrscr();
            mainscr();
            break;
           case 4:exit(0);
          default:clrscr();mainscr();;
 }
}
int main()
{


            clrscr();
mainscr();
}
```

CHAPTER 4

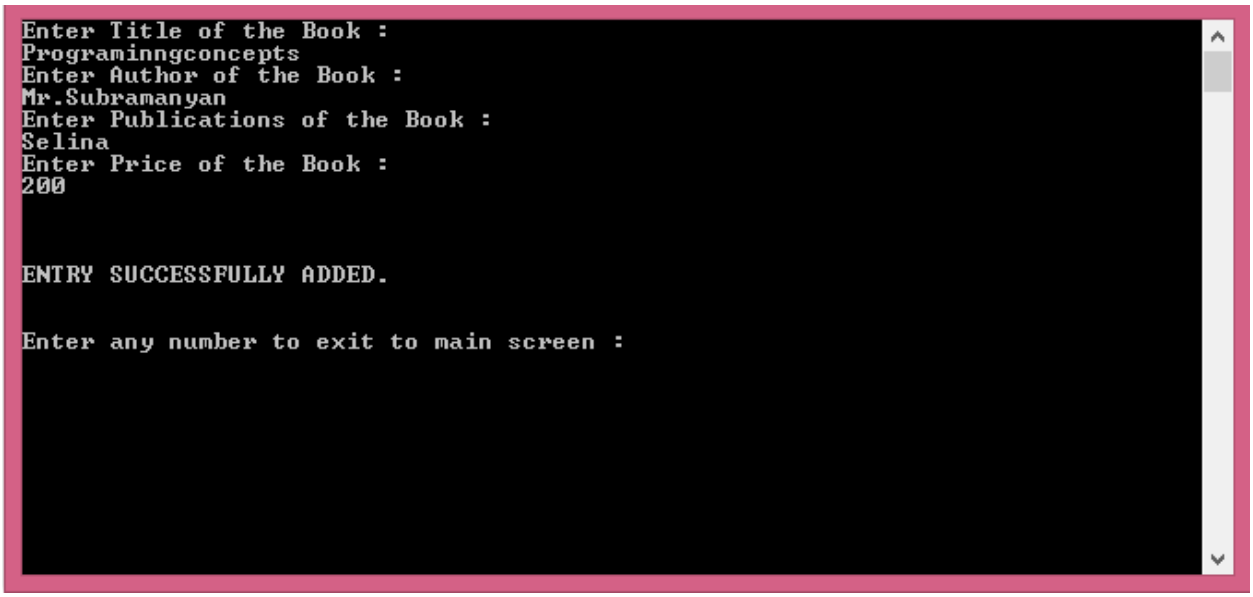
RESULTS AND DISCUSSION

4.1 Output Snapshots:



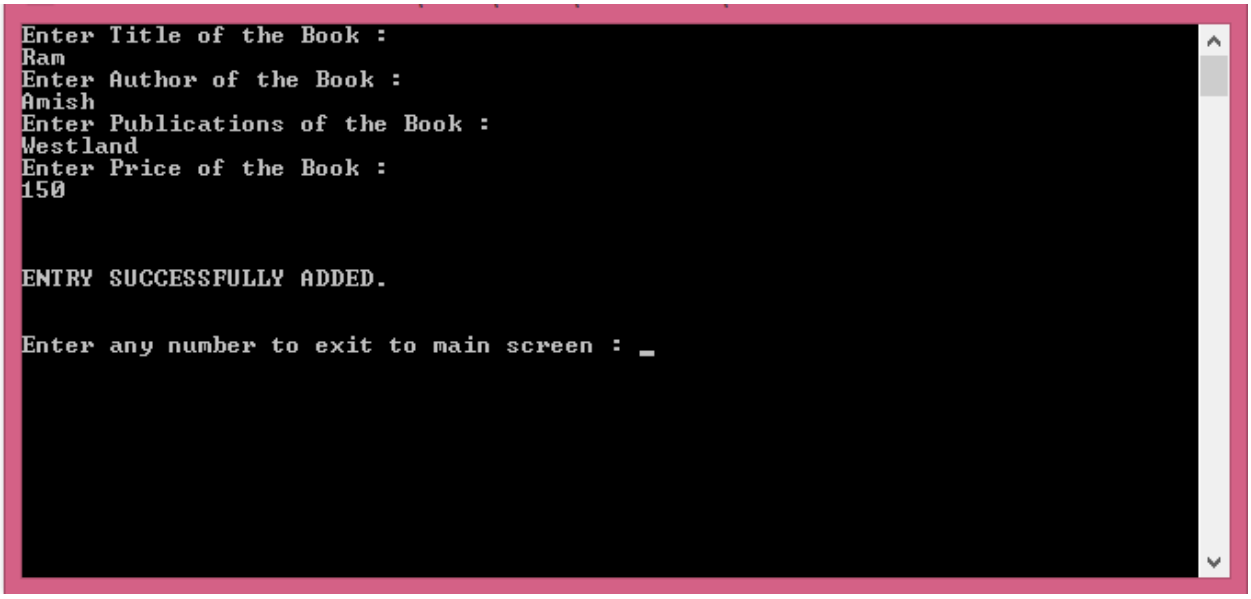
4.1.1 The first screen of output.

The user is asked to enter his choice from the list of choices given above. When the user enters his choice he is redirected to next page. He can choice of adding a book, searching for a book or for listing all the books which are present in the store.



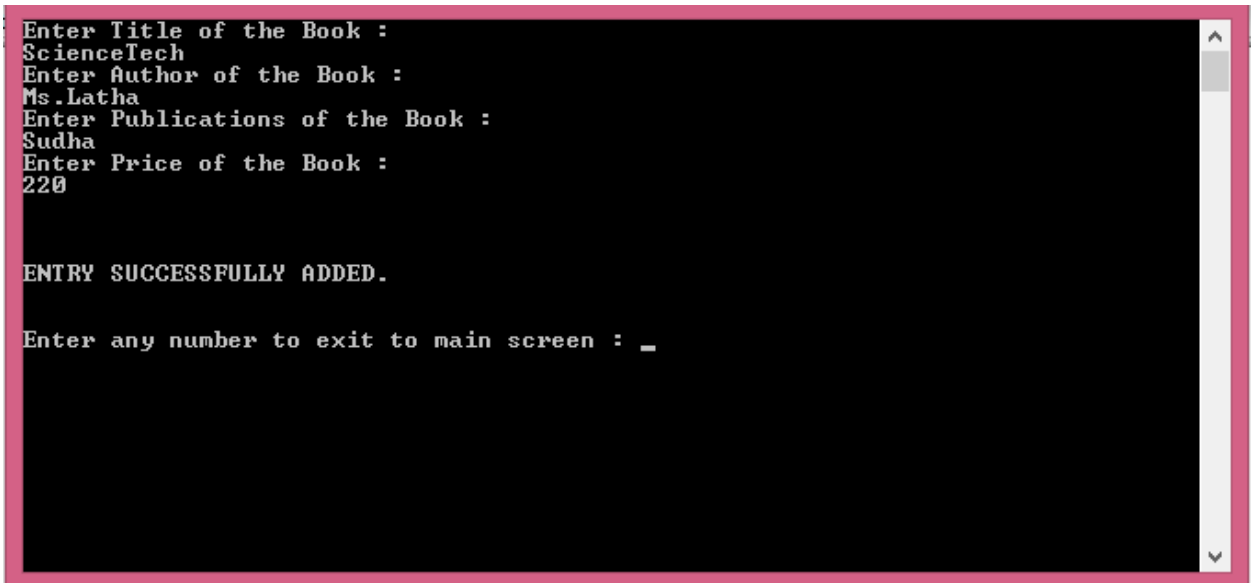
4.1.2 Entering the details of the book to be added

If option one is choosed then the user can add new book to the store by entering the title author publications and the price of the book. The details of the book which is entered will be successfully added to the store. The user can add how many ever books he wants to by entering all these details of the book. In this example the book added is Programming concepts



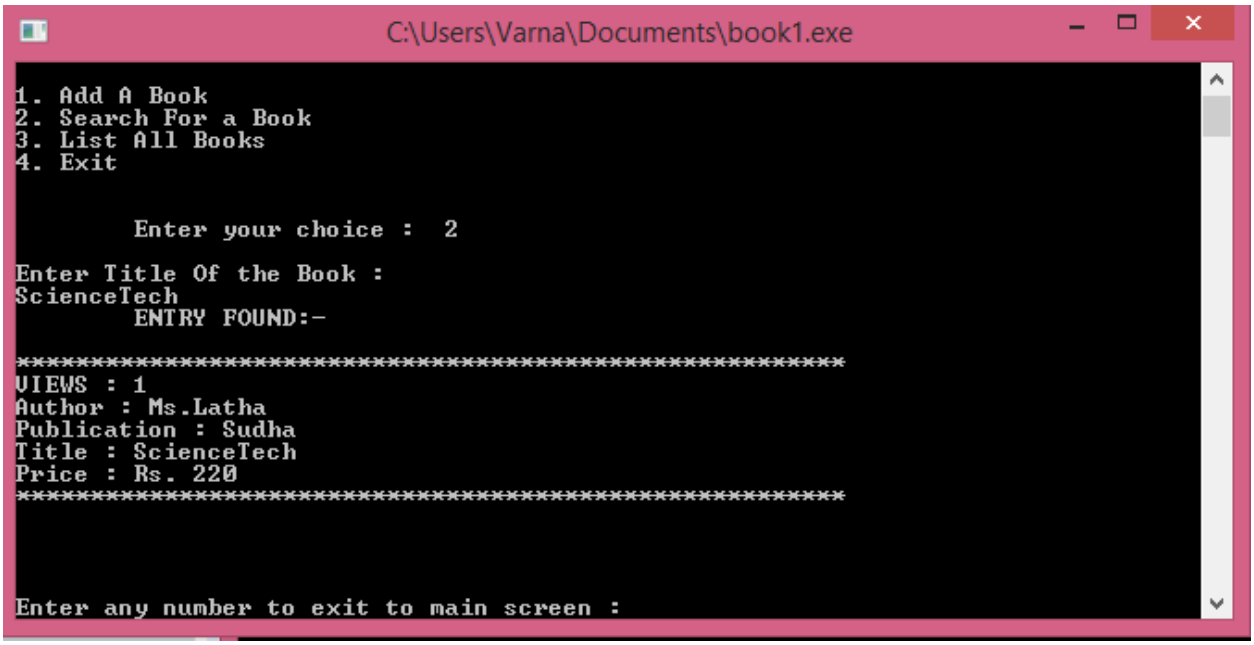
4.1.3 Entering the details of the book to be added

A new book is added to the book store. The author's name, the publications and the price of the books should be given manually. The details will be stored in the book and can be later searched for with the help of the title of the book. If option one is choosed then the user can add new book to the store by entering the title author publications and the price of the book. The details of the book which is entered will be successfully added to the store.



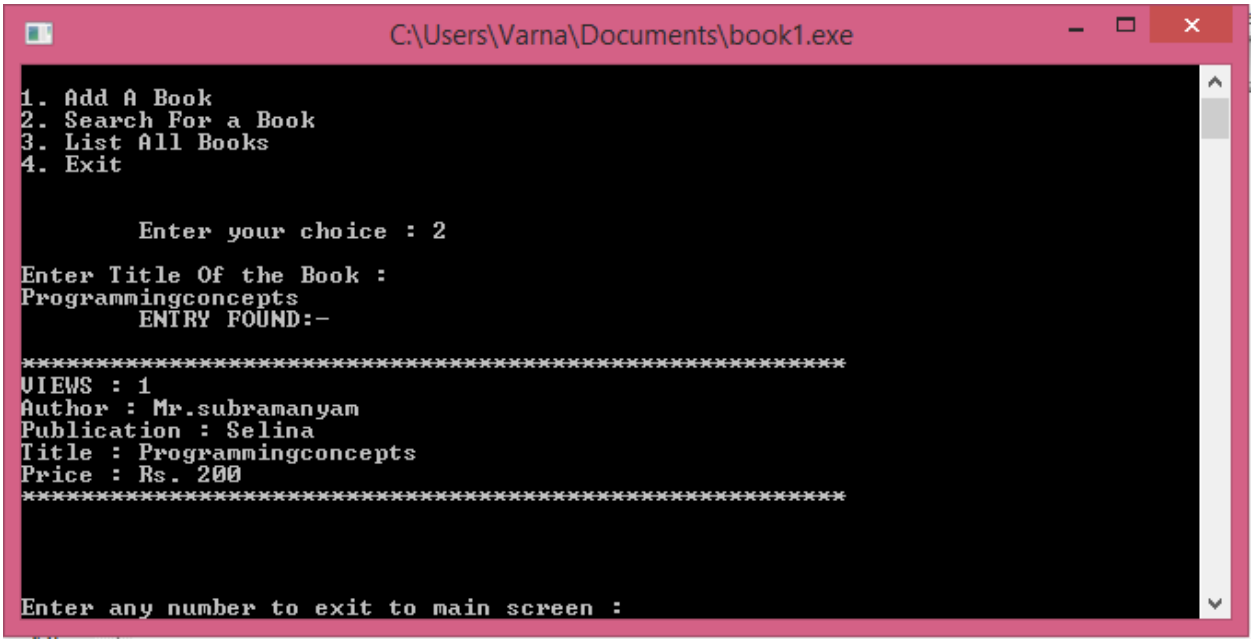
4.1.4 Entering the detail of the book to be stored

A new book can be added easily added by entering the details of the book. A new book is added to the book store. The author's name, the publications and the price of the books should be given manually. The details will be stored in the book and can be later searched for with the help of the title of the book. If option one is choosed then the user can add new book to the store by entering the title author publications and the price of the book. The details of the book which is entered will be successfully added to the store.



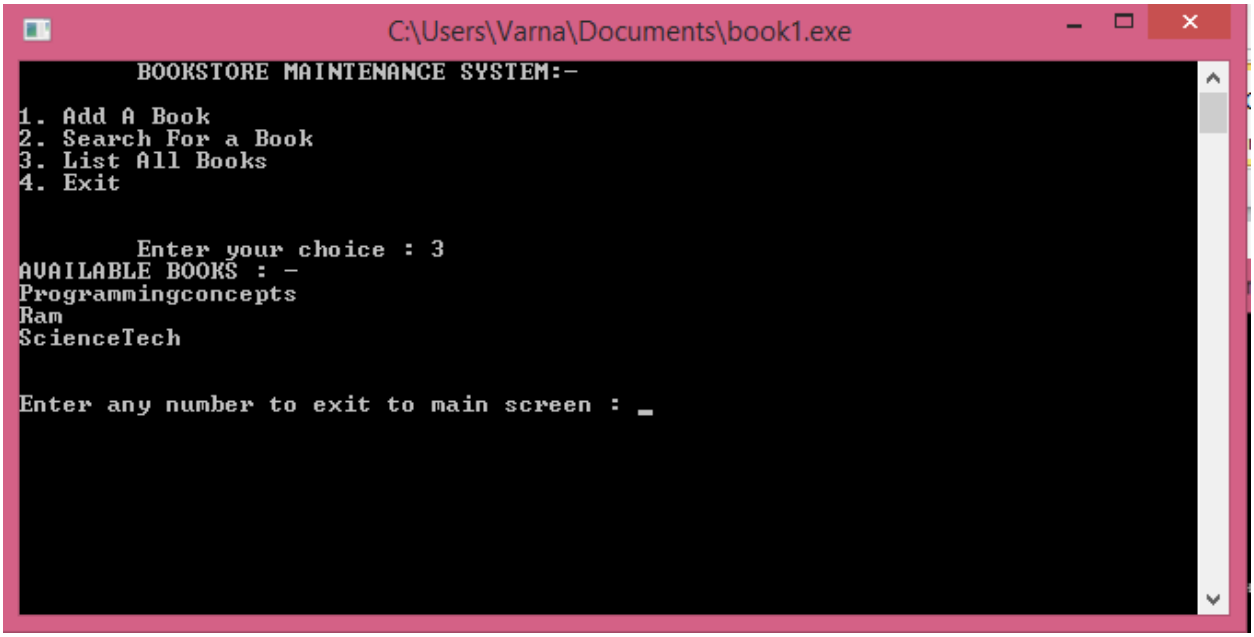
4.1.5 Searching for a book

By choosing option 2 from the main menu the user can search for a book which he wants to search. When the user enters 2 as his choice, he is asked to enter the title of the book. All the details of the book will be displayed. The author's name, its publications, title of the book will be displayed. The number of views of that particular book will be displayed.



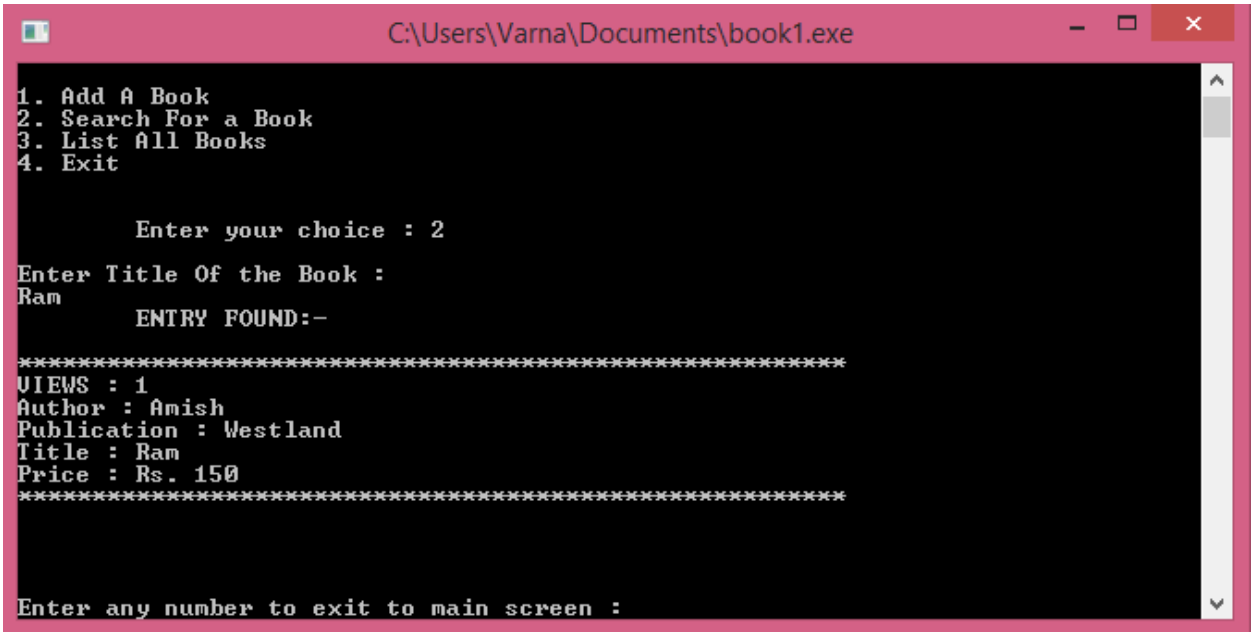
4.1.5 Searching for a book

If the user wants to search a book then he can do that by entering the title of the book .If the book is present then all the details of the book with the number of views of that book will be displayed. Based on the number of views the user can decide whether to buy that book or not. In this example the book searched for is Programming Concepts. All the details of that book with the number of views of that book is displayed. The number of views of that book presently is 1.



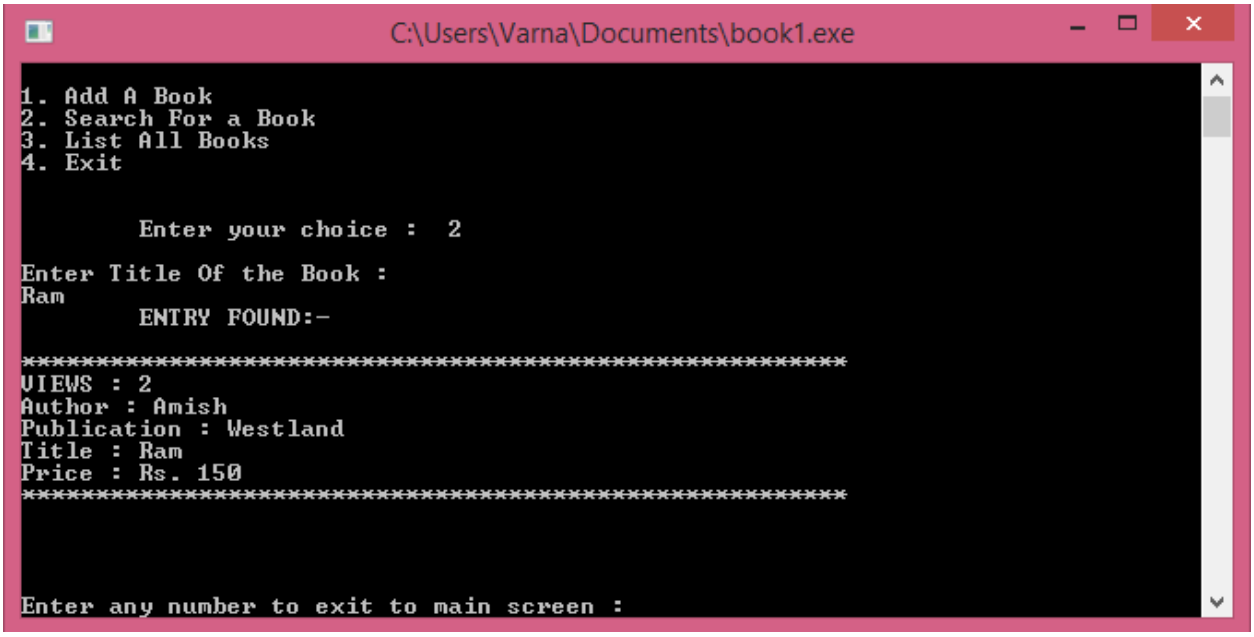
4.1.6 Displaying all the books

All the books which are present is the book store will be displayed when this option is choosed. By the option the user can get to know about the varieties of the books which are present in the book store and can decide which book he has to read from the books which are present in the bookstore. In this example the number of books is 3. All the books in the system are displayed to the user.



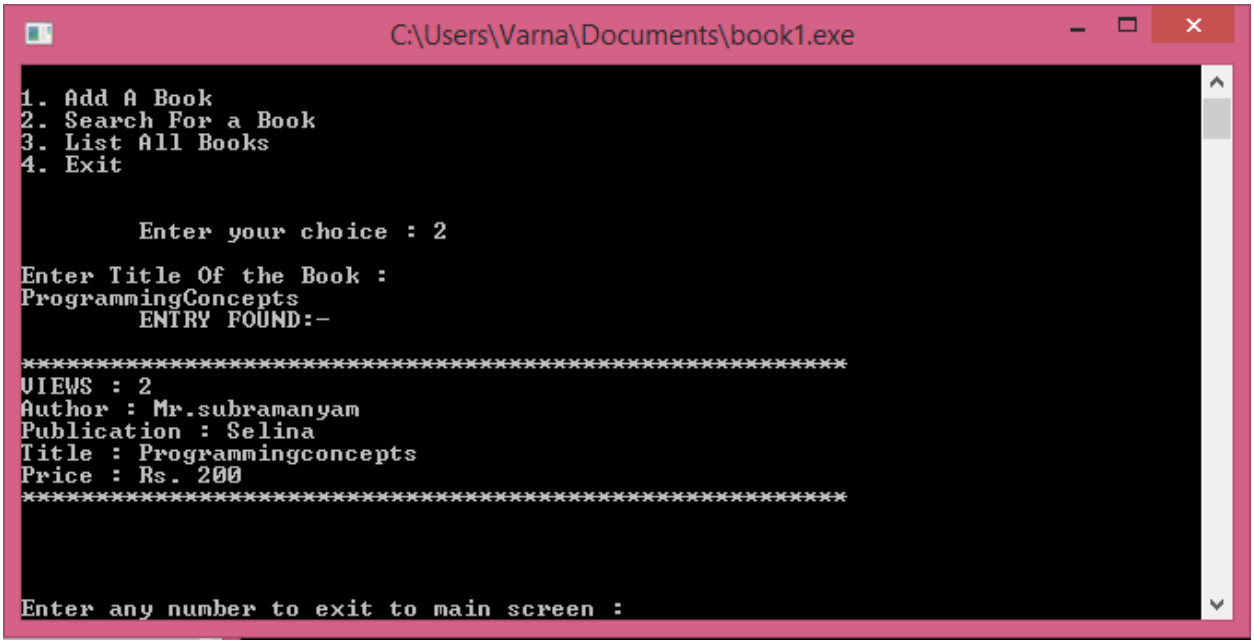
4.1.7 Searching for a book

If the user wants to search a book then he can do that by entering the title of the book .If the book is present then all the details of the book with the number of views of that book will be displayed. Based on the number of views the user can decide whether to buy that book or not. In this example the book searched for is Programming Concepts. All the details of that book with the number of views of that book is displayed. The number of views of that book presently is 1.



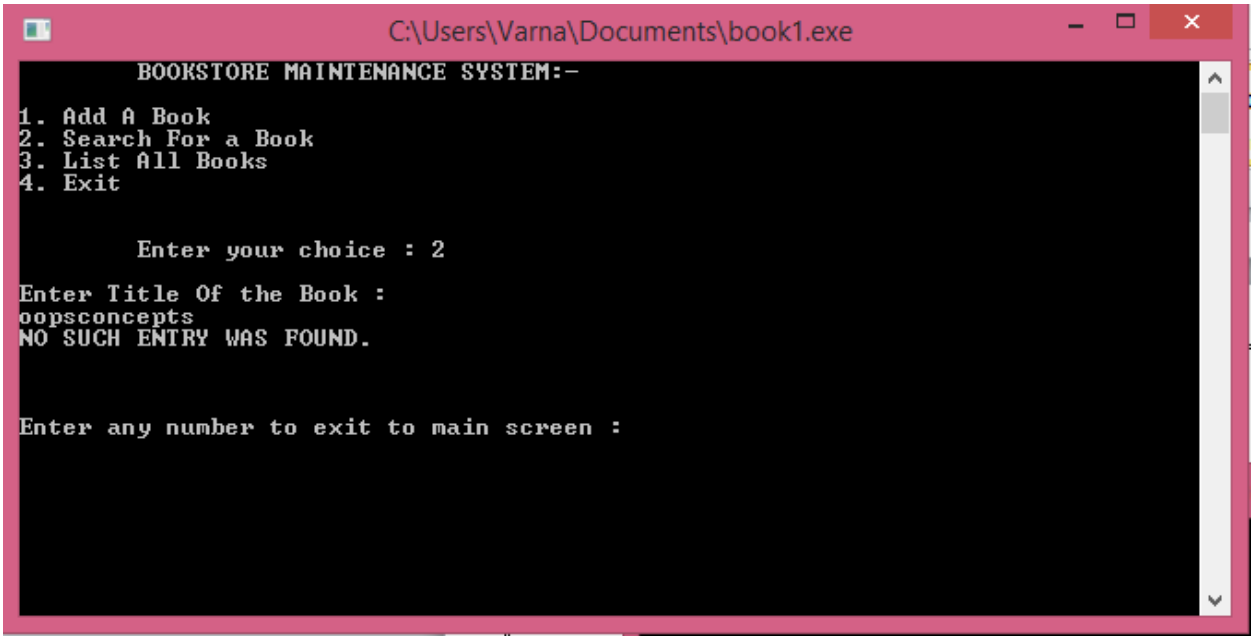
4.1.8 Incrementing the number of views

This option is to search for a book If the user wants to search a book then he can do that by entering the title of the book .If the book is present then all the details of the book with the number of views of that book will be displayed. Based on the number of views the user can decide whether to buy that book or not. Each time a book is searched for the number of views of that book gets incremented. In the previous snapshot the number of views of RAM was 1. Since another person searched for the book, number of views of that particular book increased.



4.1.9 Incrementing the views

This option is to search for a book If the user wants to search a book then he can do that by entering the title of the book .If the book is present then all the details of the book with the number of views of that book will be displayed. Based on the number of views the user can decide whether to buy that book or not. Each time a book is searched for the number of views of that book gets incremented. In the previous snapshot the number of views of Programming Concepts was 1. Since another person searched for the book, number of views of that particular book increased.



4.1.10 searching for a book

By choosing option 2 from main menu the user can search for a book which he wants to read by entering the title of the book. If the book is present in the book store the details of that book is displayed. In this snapshot the book which is searched is not present in the store, so the message "no such entry was found" is displayed.

CONCLUSION

This application project reduces manual effort and saves time. All the details of the books are computerized. In this project searching for a book becomes easier, the user can search for a book by entering its title and all the details of the book along with the number of views of that particular book will be displayed. It is a methodical way of arranging data. It is efficient and user friendly project. Book shop management system is an attempt to overcome the present inefficient and time consuming process of locating reserving and purchasing quality reading materials available in the shop. Through automated book shop solution, provide an easy way of searching reserving and purchasing of books. It's worth analyzing and identifying the benefits as it would directly influence the productivity of the shop.

REFERENCES

Books:

- Programming in c++ by Balaguru Swamy
- The C++ programming Language
- Modern C++ Design
- Principles and Practice using C++

Websites:

- https://www.quora.com
- Cppprojectcode.blogspotcom
- http://getprojects.blogspot.com
- http://www.proramiz.com