**CO101-PROGRAMMING FUNDAMENTALS**

### REPORT ON INNOVATIVE PROJECT

### Topic: LIBRARY MANAGEMENT SYSTEM

### Submitted by: Arjun Manoj Kumar 2K20/B1/25 Karan Gupta 2K20/B1/24

### Submitted to: Dr. Rajeev Kumar



## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

###### DELHI TECHNOLOGICAL UNIVERSITY

###### (Formerly Delhi College of Engineering) Shahbad Daulatpur, Bawana Road, Delhi 110042

# A CKNOWLEDGEMENT

### The completion of this project work gives us much pleasure. We would like to express our sincere gratitude to our Professor of Computer Fundamentals Dr. Rajeev Kumar under whose supervision and guidance we could successfully complete our project. His valuable suggestions throughout the duration helped us improvise our work.

### We also take this opportunity to thank our fellow classmates for their valuable comments regarding our project topic. This project gave us a golden opportunity to learn and understand the subject more.

# I NTRODUCTION

**C++** is a general-purpose programming language created by Bjarne Stroustrup as an extension of the C programming language, or "C with Classes". The language has expanded significantly over time, and modern C++ now has object-oriented and functional features in addition to facilities for low-level memory manipulation.

The Library Management System Application has been created in C++. It is mainly based on classes and data files.

This application helps the librarian to keep track of the various books in the library and also assists in the issuing and returning of books. The application is also secured by a custom password which keeps all the records safe. The application consists of 8 options available to the user.

The user can input the serial number of the option to open the desired section.

**CONTENTS**

[Acknowledgement ii](#_TOC_250004)

[Introduction iii](#_TOC_250003)

Concepts Used in the Application v

[Features of the Application vii](#_TOC_250002)

[Program Code](#_TOC_250001) x

[Output Screens xv](#_TOC_250000)ii

**CONCEPTS USED**

## VARIOUS HEADER FILES:

### #include <stdio.h> #include <stdlib.h> #include <string.h>

## FLOW OF CONTROL:

**If-Else Statements:** It is used to evaluate an expression and execute the true value.

### **switch Case:** A switch statement tests the value of a variable and compares it with multiple cases. Once the case match is found, a block of statements associated with that particular case is execution.

## ITERATION STATEMENTS:

**while Loop and for Loop:** They are entry-controlled loops i.e., there is control over entry in them.

## ARRAYS:

### It is a collection of variables of the same data type that are referenced by a common name.

**CLASSES:**

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.

### **Example**

Create a class called "MyClass":

class MyClass {       // The class  
  public:             // Access specifier  
    int myNum;        // Attribute (int variable)  
    string myString;  // Attribute (string variable)  
};

## DATA FILE HANDLING:

### The concept of Data File Handling is used in a program to store the information permanently.

# 

# FEATURES OF THE APPLICATION

## LOGIN SCREEN:

The First screen is the login screen, wherein the user types the password (1234 in this case) and is taken to the next screen. If incorrect password is typed, the user is notified that it is incorrect and the application closes.

## MAIN MENU:

### The Application has a MENU which displays all the eight options available to the user. The user can input the serial number of the option to open the desired section.

### w

### **INVENTORY:**

### Here you can add or delete a book. When adding a book, the user is prompted to input the name of the book along with its author, if it is a new entry. If it is an already existing book, then the number of copies is incremented by 1.

## AVAILABILITY MENU:

The user can type the name of the book they want to search for, the program returns the number of copies available currently in the library.

## ISSUING:

In this menu, the user can issue a book. The name of the book is typed along with the membership ID of the person issuing the book.

### **RETURNING:**

In this menu, a book can be returned. The user is prompted to type the name of the book and their membership ID.

**NEW MEMBER:**

In this menu, a new member can be added. Only the name of the user is required, the membership ID is automatically assigned to them.

## Display All Books:

This option displays all the books in the library in the order of when it was added.

## Display All Members:

This option displays all the members registered in the system along with their membership IDs.

# PROGRAM CODE

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

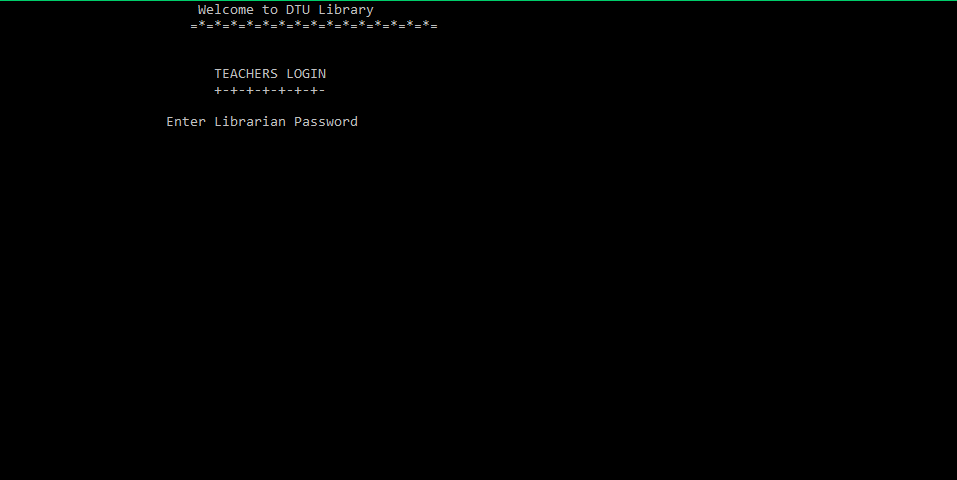
//LIBRARY MANAGEMENT SYSTEM CREATED BY KARAN AND ARJUN

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

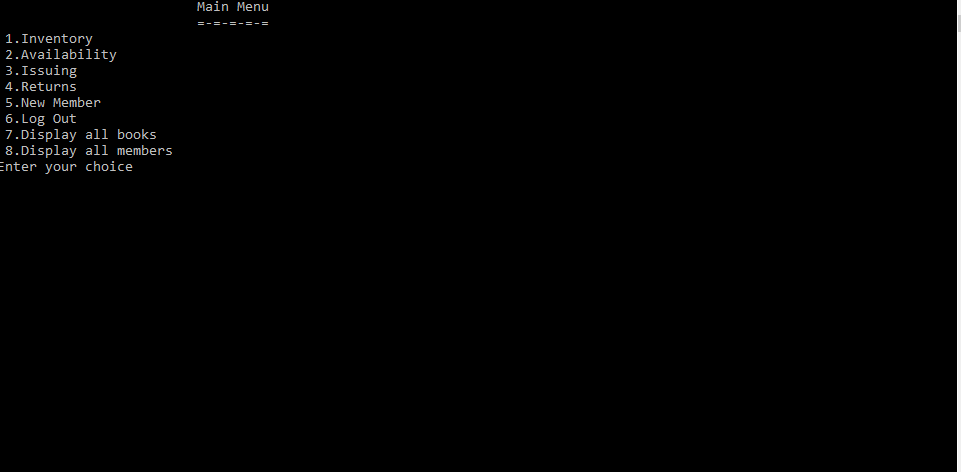
#include<iostream>  
#include<stdlib.h>  
#include<conio.h>  
#include<process.h>  
#include<fstream>  
#include<iostream>  
#include<stdio.h>  
#include<ctype.h>  
#include<cstdio>  
#include<string.h>  
#include<ios>  
#include<limits>  
using namespace std;  
  
    // begin class books  
    class books{  
        char name[50];  
        char author[50];  
        int bookcode,i;  
        public :  
        int copies;  
  
 //accept tjedetails of new book  
        void getval (){  
            ("cls");  
        cout<<"Enter Books Name\n";  
        cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
        cin.get(name, 50);  
        cout<<"\nEnter Author\n";  
        cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
        cin.get(author,50);  
        cout<<"\nBook Added\n";  
        bookcode=code();  
        copies = 1;  
        system("pause");  
        }  
  
        char \*bookname(){return name;}  
//assign code number  
        int code(){  
            books b1;  
            int c=0;  
            fstream a;  
            a.open("Library.dat",ios::in|ios::binary);  
            while (a.read((char\*)&b1,sizeof(books))){  
                c++;  
                if(!a)  
                    break;  
                }  
            a.close();  
            return(c+1);  
           }  
  
        int showcode(){return bookcode;}  
  
        void showval(){ cout<<name<<"\t\t"<<author<<"\t\t"<<bookcode; }  
      };  
  
// begin class member  
     class member{  
        char membername[50];  
        public :  
        int membership;  
        char bookname[50];  
        void insertmem(){  
            system("cls");  
            cout<<"\t\t New Members"<<endl;  
            cout<<"\t\t =\*=\*=\*=\*=\*="<<endl;  
            cout<<"Enter Members Name"<<endl;  
            cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
            cin.get(membername,50);  
            membership=re();  
            cout<<"Member added\n";  
            system("pause");  
                }  
  
        int re(){  
        member m1;  
        int c=0;  
        fstream a;  
        a.open("Member.dat",ios::in|ios::binary);  
        while (a.read((char\*)&m1,sizeof(member))){  
                c++;  
                if(!a)  
                    break;  
  
            }  
        a.close();  
        return(c+1);  
        }  
     void show()  
     { cout<<membername<<"\t\t"<<membership<<"\t\t"<<bookname; }  
  
  
    };  
// Create a new book or add existing book  
     void newb(){  
        fstream afile;  
        char na[50];  
        books b1;  
        int b;  
  
        cout<<"Do You want to add a new book or add a copy of existing book\n1.New Book 2.Existing Book"<<endl;  
        cin>>b;  
        switch (b){  
            case 2:  
                system("cls");  
                afile.open("Library.dat",ios::in|ios::binary|ios::out);  
                cout<<"Enter the name of book"<<endl;  
                cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
                cin.get(na,50);  
                while(afile.read((char\*)&b1,sizeof(books)))  
                {  
                    if(strcmp(na,b1.bookname())==0){  
                        b1.copies++;  
                        afile.seekp(-sizeof(books),ios::cur);  
                        afile.write((char\*)&b1,sizeof(books));  
                        cout<<"Book added\n";  
                    }  
                }  
                system("pause");  
                break;  
            case 1:  
                afile.open("Library.dat",ios::app|ios::binary);  
                b1.getval();  
                afile.write((char\*)&b1,sizeof(books));  
                break;  
            default :  
                cout<<"Wrong Choice";  
                break;  
 }  
 afile.close();  
}  
  
// Deleting a book  
        void deleteb(){  
            books b1;  
            int bid ;  
            char bname[50];  
        fstream a;  
        fstream b;  
        cout<<"Enter the book name"<<endl;  
        cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
        cin.get(bname,50);  
        a.open("Library.dat",ios::in|ios::binary);  
        b.open("i.dat",ios::out|ios::binary);  
        while(a.read((char\*)&b1,sizeof(books))){  
                if(strcmp(bname,b1.bookname())==0){  
                    cout<<"Record has been deleted"<<endl;  
                    system("pause");}  
                else  
                    b.write((char\*)&b1,sizeof(books));  
        }  
       a.close();  
       b.close();  
       remove("Library.dat");  
       rename("i.dat","Library.dat");  
      }  
//Inventory - to add a new/existing book or deleting a book  
    void inventory(){  
        system("cls");  
        int a;  
        cout<<"\t\t INVENTORY"<<endl;  
        cout<<"\t\t +-+-+-+-+"<<endl;  
        cout<<"1.New Book \n2.Delete Record"<<endl<<"Enter your choice  ";  
        cin>>a;  
        switch(a){  
            case 1:  
            newb();  
            break;  
            case 2:  
            deleteb();  
            break;  
            default:  
            cout<<"\nwrong choice";  
            }  
  
        }  
// checking availability  
    void avail(){  
        system("cls");  
        books b1;  
        char name[50];  
        cout<<"\t\t\tAvailability"<<endl;  
        cout<<"\t\t\t=-=-=-=-=-=-"<<endl;  
        cout<<"Enter The Book Name to be searched for "<<endl;  
        cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
        cin.get(name,50);  
        fstream a;  
        a.open("Library.dat",ios::in|ios::binary);  
        while(a.read((char\*)&b1,sizeof(books))){  
            if(strcmp(name,b1.bookname())==0){  
            cout<<"\nThe No of Copies Available "<<b1.copies<<endl;  
            system("pause");}  
        }  
      }  
  
// adding a new member  
    void newmember(){  
        member m1;  
        m1.insertmem();  
        fstream a;  
        a.open("Member.dat",ios::app|ios::binary);  
        a.write((char\*)&m1,sizeof(member));  
        }  
// Issuing a book  
    void issue(){  
        char name[50];  
        books b1;  
        member m1;  
        int no;  
        fstream a,b;  
        system("cls");  
        cout<<"\t\t ISSUING"<<endl;  
        cout<<"\t\t =\*=\*=\*="<<endl;  
        cout<<"Enter the book to be Issued"<<endl;  
        cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
        cin.get(name,50);  
        cout<<"\nEnter The Membership No"<<endl;  
        cin>>no;  
        a.open("Library.dat",ios::out|ios::binary|ios::in);  
        b.open("Member.dat",ios::out|ios::binary|ios::in);  
        while (a.read((char\*)&b1,sizeof(books))){  
                if (strcmp(name,b1.bookname())==0){  
                    b1.copies--;  
                    a.seekp(-sizeof(books),ios::cur);  
                    a.write((char\*)&b1,sizeof(books));  
      }}  
        while (b.read((char\*)&m1,sizeof(member))){  
                if (no==m1.membership){  
                    strcpy(m1.bookname,name);  
                    b.seekp(-sizeof(books),ios::cur);  
                    b.write((char\*)&b1,sizeof(books));  
                    }  
       }  
     cout<<"Book Issued";  
     getch();  
       }  
//book return  
    void returns(){  
        char name[50];  
        books b1;  
        member m1;  
        int no;  
        fstream a,b;  
        system("cls");  
        cout<<"Enter the book to be Returned"<<endl;  
        cin.ignore(numeric\_limits<streamsize>::max(), '\n');  
        cin.get(name,50);  
        cout<<"\nEnter The Membership No"<<endl;  
        cin>>no;  
        a.open("Library.dat",ios::out|ios::binary|ios::in);  
        b.open("Member.dat",ios::out|ios::binary|ios::in);  
        while (a.read((char\*)&b1,sizeof(books))){  
                if(strcmp(name,b1.bookname())==0){  
                    b1.copies=b1.copies+1;  
                    a.seekp(-sizeof(books),ios::cur);  
                    a.write((char\*)&b1,sizeof(books));}  
      }  
        while (b.read((char\*)&m1,sizeof(member))){  
                if(no==m1.membership){  
                    m1.bookname=="\o";  
                    b.seekp(-sizeof(books),ios::cur);  
                    b.write((char\*)&b1,sizeof(books));  
    }}  
    cout<<"Book Returned\n";  
    system("pause");  
    getch();  
   }  
//display books  
    void book(){  
        fstream a;  
        books b1;  
        a.open("Library.dat",ios::in|ios::binary);  
        system("cls");  
        cout<<"List:\n";  
        while (a.read((char\*)&b1,sizeof(books))){  
            b1.showval();  
            cout<<"\n";  
      }  
      getch();  
    }  
//display members  
    void members(){  
        fstream a;  
        member m1;  
        system("cls");  
        cout<<"List of members:\n";  
        a.open("Member.dat",ios::in|ios::binary);  
        while(a.read((char\*)&m1,sizeof(member))){  
            m1.show();  
            cout<<"\n";  
        }  
        getch();  
        }  
  
   int main (){  
       system("cls");  
       int j,N,choice;  
       j=1;  
  
   cout<<"\t\t\t Welcome to DTU Library"<<endl;  
   cout<<"\t\t\t=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*=\*="<<endl<<endl<<endl;  
   cout<<"\t\t\t   TEACHERS LOGIN"<<endl;  
   cout<<"\t\t\t   +-+-+-+-+-+-+-"<<endl;  
   cout<<"\n\t\t     Enter Librarian Password  ";  
   cin>>N ;  
   if(N==1234){  
   while(j==1){  
    system("cls");  
     cout<<"\t\t\t Main Menu"<<endl;  
     cout<<"\t\t\t =-=-=-=-="<<endl;  
     cout<<" 1.Inventory \n 2.Availability \n 3.Issuing \n 4.Returns \n 5.New Member \n 6.Log Out\n 7.Display all books\n 8.Display all members"<<endl;  
     cout<<"Enter your choice  ";  
     cin>>choice;  
     switch (choice){  
        case 1:  
        inventory();  
        break;  
        case 2:  
        avail();  
        break;  
        case 3:  
        issue();  
        break;  
        case 4:  
        returns();  
        break;  
        case 5:  
        newmember();  
        break;  
        case 6:  
        exit(0);  
        case 7:  
        book();  
        break;  
        case 8:  
        members();  
        break;  
        default:  
        cout<<"Wrong Option";  
        break;  
}  
system("cls");  
    cout<<"Do you Want to use again 1.Yes 2. No    ";  
cin>>j;  
}}  
else  
{cout<<"\t\t\t\a    Incorrect Password\n  ";  
     system("pause");  
     exit(0);  
}  
system("cls");  
cout<<"Thanks For Using";  
getch();

# OUTPUT SCREENS

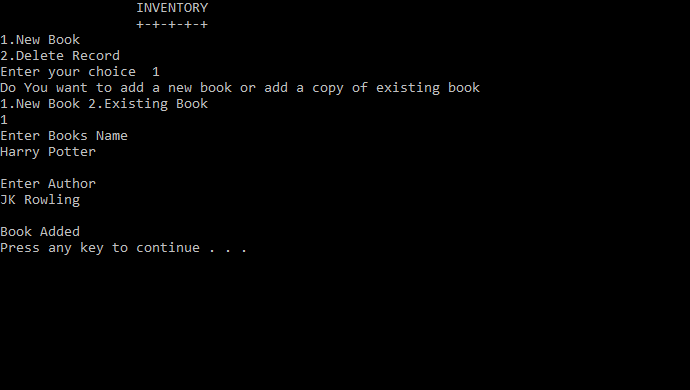
##### LOGIN SCREEN:

****

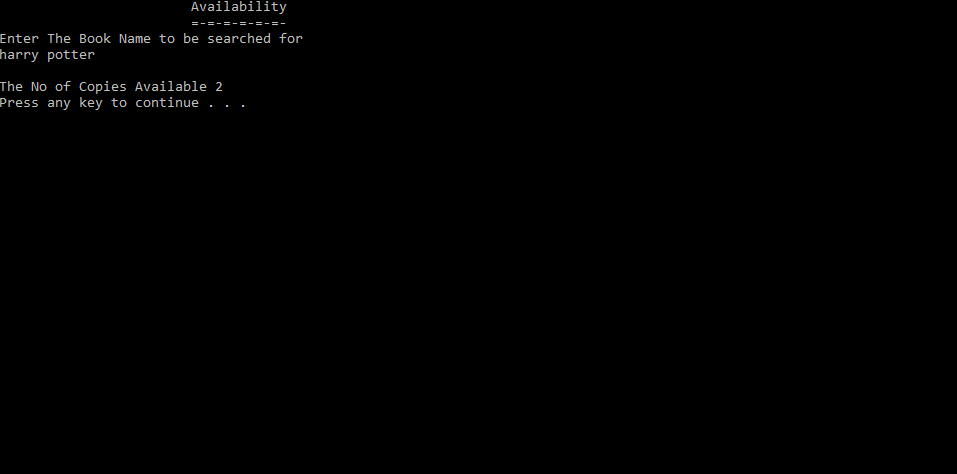
##### MAIN MENU:



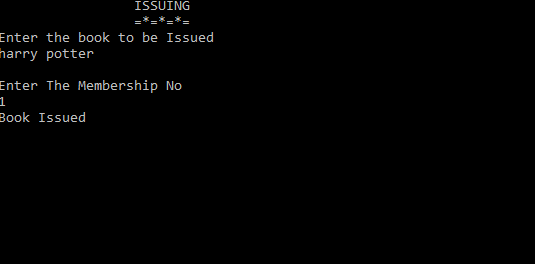
##### INVENTORY:

****

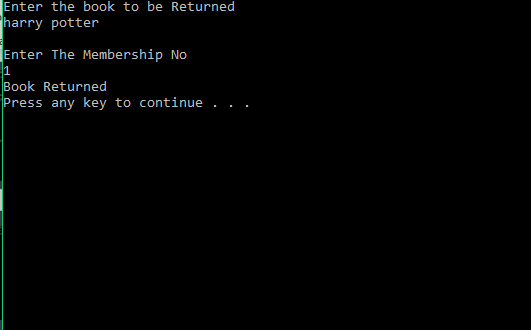
**AVAILABILTY MENU:**

****

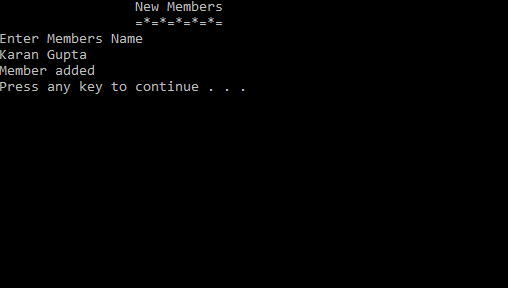
##### ISSUING:

****

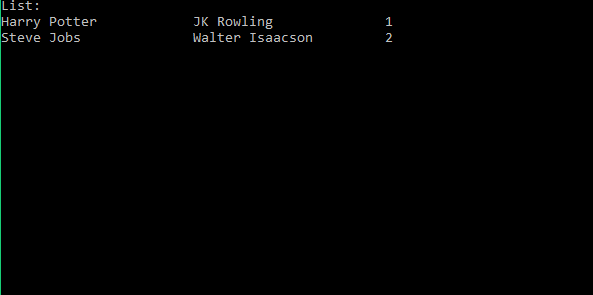
##### RETURNING:

****

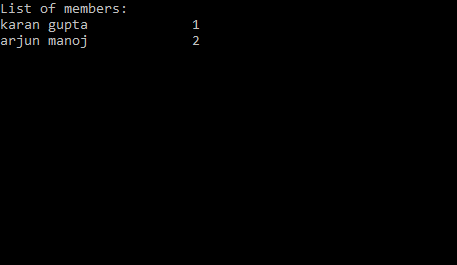
##### New Member:

****

##### Display All Books:



##### Display All Members:

****