

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | | | | | | | | |
|  | | Library Management System | | | | |  | |
|  |  | | | | | | |  |
|  | | | |  |  | | | |
|  | | | | Muhammad Raza Madni |  | | | |
|  | | | | Date08-07-2022 |  | | | |
|  | | |  | | |  | | |

**Project**

* **Name:**
* Library Management System
* **Description:**
* Library Management System is a system which maintains the information about the books present in the library, their authors, the members of library to whom books are issued. This is very difficult to organize manually. Maintenance of all this information manually is a very complex task.This computerization of library helps in many instances of its maintenances. It reduces the workload of management as most of the manual work done is reduced.
* **Functionalities:**
* Add Books
* Display All Books
* Search a Particular Book
* Issue Books
* Display Issued Book Data
* Add Students
* Display All Students
* **Topics Used:**
* Classes
* Inheritance
* Files Handling
* Pointers
* Functions
* **Pseudocode:**
  + - Start.
    - Globally defined variables.
    - Prototype of the functions.
    - Base class starts.
      * Publicly defined char type arrays.
      * Publicly defined files.
      * 1st member function stars to display book.
        + LMS file opens
        + Char defined to get data from this file
        + While stats to read this till end

Replace ‘|’ from file with 25 space

Display data stored in char variable

* + - * + File closes
      * Member function ends.
      * 2nd member function starts to add books in library.
        + LMS file opens
        + Getting all data from the user
        + Storing it in file (opened)
        + File closes
      * Member function ends.
      * 3rd member function starts to find Book by id.
        + LMS file opens
        + Defined variable to store id names
        + Getting input
        + While loop starts to read file till end
        + Getting data from file into fin (char type array) till ‘|’
        + Returning pointer of first occurrence of input id to id stored
        + Getting data from file into ftitle till ‘|’
        + Getting data from file into fname till ‘|’
        + if condition starts

bool true then

display all data that is found

* + - * + Stopping extraction of data from file
      * Member function ends.
      * 4th member function starts to find book by title/name.
        + Work similarly as 3rd member function
      * Member function ends.
      * 5th member function stars to find book by author’s name.
        + Works similarly as 3rd member function
      * Member function ends.
      * Member function to find Books starts.
        + Do-while loop start
        + Taking input
        + Switches starts

Case 1: call 3rd member function to search by id

Case 2: call 4th member function to search by title

Case 3: call 5th member function to search by author’s name

Case 4: to exit

* + - * + Switches ends
        + Do while ends
    - Base class ends.
    - Derived class named student starts.
      * 1st member function starts to add/register students.
        + Taking input from user
        + Std named file opens
        + Storing all the data to file
        + Std file close
      * Member function ends.
      * 2nd member function starts to display all data.
        + Work similarly as 1st member function of base class LMS
      * Member function Ends.
    - Derived class named student ends.
    - Derived class named issuebook starts.
      * 1st function to display all issued books
        + Work similarly as 1st member function of base class LMS
      * Member function ends.
      * 2nd member function starts.
        + Calling isid to see whether the book you want to issue is already issued or not
        + If starts

For isbk=2

File named issue opens

Taking input

Store all input to file

Close file

* + - * + If ends
      * Member function ends.
      * 3rd member function starts.
        + Works similarly as 3rd member function of class
      * Member function ends.
    - Derived class named issuebooks ends.
    - Function to display heading.
    - Function to display header of books in library.
    - Function to display header of students in file.
    - Function to display header of issued books.
    - Function to display choices.
    - Main function starts.
      * Base class object declaration.
      * Derived class (issuebooks) object declaration.
      * Derived class (student) object declaration
      * Different types of variable arrays declaration.
      * While loop starts to keep program running ( in it ) till user exit.
        + Re-enter to return to man menu
        + Taking input
        + System clear command
        + Calling heading function
        + If starts

Defining char term n to keep program running (in it)

2nd while starts

System clear command

Variable/array declaration

Calling choice function

Taking choice

Switches start

Case 1: calling addbook function from base class

Case 2: calling bk\_header function + allbooks function (from base class)

Case 3: calling findbook function from base class

Case 4: calling issbooks function from derived class (issuebook)

Case 5: calling isbk\_header function + All\_issbook function from derived class (issuebook)

Case 6: calling reg-std function from derived class (student)

Case 7: calling st\_header function + Allstd function from derived class (student)

Case 8: taking input

If starts

to exit choice menu

else to goto plan\_change

switch ends

system clear command

2nd while ends

Else invalid (login info)

* + - * + Else if (selects 2)

1st while loop ends.

* + - * + Else goto re\_enter
        + If ends
    - End.
* **Code**

#include<iostream>

#include<fstream>

#include<iomanip>

#include<string>

#include <windows.h>

#include <string.h>

using namespace std;

//////////////Global variable declaration///////////////

int opt, isbk;;

char fid[10], fed[10];

char fname[25], fTitle[25];

char\* ptr;

char space[2] = { '|','\0'};

//////////////Prototypes of function////////////////////

void choice();

void heading();

void bk\_header ();

void std\_header();

void isbk\_header();

class LMS //base class

{

public:

char stdID[10],stdclass[10],stdname[25]; //char arrays to store student data

char BookID[5],BookName[25],AuthorsName[25]; //char arrays to store book data

//////////////files decleration/////////////////////////

fstream file;

fstream stdfile;

fstream issue;

//////////////class member functions////////////////////

void AllBooks() // to print all books + details

{

file.open("LMS.txt", ios::in); //file opens

char ch; //char declar to stor data from the file

while (!file.eof()) //while condition to read file till end

{

file.get(ch); //storing ata from file (LMS)

if (ch == '|') //if condition to replace |

{

cout << setw(15); //replacing |with setw(15)

continue; //to continue same conditon

}

cout << ch; //print the data stored

}

file.close(); //closing file (LMS)

system("pause");

}

void addBooks() // to add new book to library

{

cout<<"\nAdding Books\n";

file.open("LMS.txt",ios::app); //LMS named file opens,getting data from the user.

cout<<"\nBook ID: ";

cin>>BookID;

cout<<"Book Name: ";

cin.ignore(); //to ignore spaces/null charactors

cin.getline(BookName,25);

cout<<"Authors Name: ";

cin.getline(AuthorsName,25);

file<<BookID<<space<<BookName<<space<<AuthorsName<<space<<endl; //storing all data to file (LMS)

file.close(); //File(LMS) closes

cout<<"\nBook is registered.\n";

system("pause");

}

//////////////search function////////////////////

void byid() //to search by id

{

file.open("LMS.txt", ios::in); //LMS named file opens

char fidd[10]; //char define to get id to be searched

cout << "Enter ID You Want To Search: ";

cin >>fidd;

while (!file.eof()) //while condition to read file till end

{

file.getline(fid, 10, '|'); //getting data from the file in fid till |

ptr = strstr(fid, fidd); //returing pointer to first occurence of fidd to fid

file.getline(fTitle, 25, '|'); //getting data from the file in fTitle till

file.getline(fname, 25, '|'); //getting data from the file in fname till

if (ptr) //if starts (for true, proceed)

{

cout << "Book Found, printing details\n"; //print all the data found

cout << "\nBook name: " << fTitle;

cout << "\nBook's Author: " << fname;

cout << "\nBood id: " << fidd;

break;

}

file.ignore(); //stop extraction of charactors

}

cout <<endl<<endl;

system("pause");

file.close(); //File(LMS) closes

}

void byTitle() //to search by title

{

file.open("LMS.txt", ios::in); //LMS named file opens

char fTitle1[25]; //char define to get title to be searched

cout << "Enter Title You Want To Search: ";

cin.ignore(); //ignoring null charactors

cin.getline(fTitle1,25); //getting input

while (!file.eof()) //while condition to read file till end

{

file.getline(fid, 10, '|'); //getting data from the file in fid till |

file.getline(fTitle, 25, '|'); //getting data from the file in fTitle till |

file.getline(fname, 25, '|'); //getting data from the file in fname till |

ptr = strstr(fTitle, fTitle1); //returing pointer to first occurence of fTile1 to fTitle

if (ptr) //if starts (for true, proceed)

{

cout << "Book Found, printing details\n"; //print all the data found

cout << "\nBook name: " << fTitle;

cout << "\nBook's Author: " << fname;

cout << "\nBood id: " << fid;

break;

}

file.ignore(); //stop extraction of charactors

}

cout <<endl<<endl;

system("pause");

file.close(); //File(LMS) closes

}

void byAuthor() //to search by Authors name

{

file.open("LMS.txt", ios::in); //LMS named file opens

char fname1[25]; //char define to get author's name to be searched

cout << "Enter Author's Name: ";

cin.ignore(); //ignoring null charactors

cin.getline(fname1, 25); //getting input

while (!file.eof()) //while condition to read file till end

{

file.getline(fid, 10, '|'); //getting data from the file in fid till |

file.getline(fTitle, 25, '|'); //getting data from the file in fTitle till |

file.getline(fname, 25, '|'); //getting data from the file in fname till |

ptr = strstr(fname, fname1); //returing pointer to first occurence of fname1 to fname

if (ptr) //if starts (for true, proceed)

{

cout << "Book Found, Printing Details\n\n"; //print all the data found

cout << "\nBook name: " << fTitle;

cout << "\nBook's Author: " << fname;

cout << "\nBood id: " << fid;

break;

}

file.ignore(); //stop extraction of charactors

}

cout <<endl<< endl;

system("pause");

file.close(); //File(LMS) closes

}

void FindBooks()

{

do //Do-while loop start

{

system("cls");

heading(); //calling function named heading

cout << "1) Search by id \n2) Search by Title\n"; //asking for input

cout << "3) Search by Author \n4) exit\n\n";

cout << "option: ";

cin >> opt;

switch (opt) //switches start

{

case 1:

byid(); //calling (byid) member function

break;

case 2:

byTitle(); //calling (bytitle) member function

break;

case 3:

byAuthor(); //caling (byauthor) member function

break;

}

}

while (opt != 4); //condition for loop to end

}

};

//////////////////////////////////////////////////////////////////////////

class student : public LMS //derived class

{

public:

//////////////class member functions////////////////////

void Reg\_std() // to register students

{

cout<<"\nAdding a new student\n"; //taking data

cout<<"\n ID: ";

cin>>stdID;

cout<<"\n Name: ";

cin.ignore();

cin.getline(stdname,25);

cout<<"\n Class: ";

cin.getline(stdclass,10);

stdfile.open("Std.txt",ios::app); //stdfile opens

stdfile<<stdID<<space<<stdname<<space<<stdclass<<space<<"\n"; //storing data in stdfile

stdfile.close(); //stdfile closes

cout <<"\nStudent is registered.\n";

system("pause");

}

void Allstd() // to print all students + details

{

stdfile.open("Std.txt",ios::in); //stdfile opens

char Ch; //char defined to store data in stdfile

while(!stdfile.eof()) //while condition to read file till end

{

stdfile.get(Ch); //storing ata from file (LMS)

if (Ch == '|') //if condition to replace |

{

cout << setw(15); //replacing |with setw(15)

continue; //to continue same conditon

}

cout << Ch; //print the data stored

}

stdfile.close(); //closing file (std)

system("pause");

}

};

//////////////////////////////////////////////////////////////////////////

class issuebooks :public LMS

{

public:

//////////////class member functions////////////////////

void All\_issBooks() // to print all books + details

{

issue.open("issue.txt", ios::in); //file opens

char ich; //char declar to stor data from the file

while (!issue.eof()) //while condition to read file till end

{

issue.get(ich); //storing ata from file

if (ich == '|') //if condition to replace |

{

cout << setw(15); //replacing |with setw(15)

continue; //to continue same conditon

}

cout << ich; //print the data stored

}

issue.close(); //closing file (issue)

system("pause");

}

void IssBooks() // to issue books

{

isid(); //calling function named isid

if (isbk==2) //if comdition

{

cout<<"\nIssuing Books\n"; //taking data

string isu\_date,rtn\_date;

issue.open("issue.txt",ios::in | ios::app);

cout<<"\nStudent ID: ";

cin>>stdID;

cout<<"Student Name: ";

cin.getline(stdname,25);

cin.getline(stdname,25);

cout<<"Enter Book id: ";

cin.getline(BookID,5);

cout<<"Book title: ";

cin.getline(BookName,25);

cout<<"Enter issue date: ";

getline(cin,isu\_date);

cout<<"Enter return date: ";

getline(cin,rtn\_date);

issue<<BookID<<space<<BookName<<space<<stdname<<space<<stdID<<space<<isu\_date<<space<<rtn\_date<<space<<endl; //storing data to file

issue.close(); //file closes

cout <<"\nBook is issued.\n";

system("pause");

}

}

void isid() //to search issued book

{

issue.open("issue.txt", ios::in); //file opens

char fid1[10]; //char to get id to be searched

cout << "Enter Book id: ";

cin >>fid1;

while (!issue.eof()) //loop to read file til end

{

issue.getline(fid, 10, '|'); //getting data from file till | to fid

ptr = strstr(fid, fid1); //returing pointer to first occurence of fidd to fid

if (ptr) //if starts (for true, proceed)

{

cout << "\nBook is already issued...\n"; //if found

isbk=1; //isbk=1

break;

}

else

{

cout<<"Book is not issued yet...\n"; //if not

isbk=2; //isbk=2

break;

}

issue.ignore(); //stop extraction of charactors

break;

}

cout << endl;

system("pause");

issue.close(); //file closes

}

};

//////////////////////////////////////////////////////////////////////////

void heading() //function to print hading

{

cout<<"\n\t\t\*\*\*\*\* LIBRARY MANAGEMENT SYSTEM \*\*\*\*\*\n\n";

}

void bk\_header () //function to print header of books details

{

cout<<"I.D"<<setw(20)<<"Title"<<setw(20)<<"Author"<<endl;

for(int i=0; i<=60; i++ )

cout<<"-";

cout<<endl;

}

void std\_header() //function to print header of student details

{

cout<<"I.D"<<setw(25)<<"Name"<<setw(25)<<"Class"<<endl;

for(int i=0; i<=60; i++ )

cout<<"-";

cout<<endl;

}

void isbk\_header() //function to print header of issue book details

{

cout<<"Book\_Name"<<setw(17)<<"Book\_ID"<<setw(23)<<"Student\_Name"<<setw(20)<<"Student\_ID"<<setw(20)<<"Issue\_Date"<<setw(20)<<"Return\_Date"<<endl;

for(int i=0; i<=120; i++ )

cout<<"-";

cout<<endl;

}

//////////////////////////////////////////////////////////////////////////

void choice() //function to display choice

{

cout<<"How can we help you.\n";

cout<<"\n\t1) Add Books"<<setw(37)<<"2) All Books";

cout<<"\n\t3) Search Books"<<setw(36)<<"4) Issue Books";

cout<<"\n\t5) All Issued Books"<<setw(36)<<"6) Adding Students";

cout<<"\n\t7) All Students"<<setw(29)<<"8)Exit\n";

cout<<"\nchoose any.\n==> ";

}

int main() //main start

{

///////////////variable declaration///////////////

LMS obj; //base class objective

issuebooks oobj; //derived class (issuebooks) objective

student ooobj; //derived class (student) objective

char done = 'n';

string username;

string password;

int select;

char confirm;

system("color 0b");

while(done!='y') //while loop to start keep the funtion(in it) runing till done=n

{

re\_enter: //goto function part

heading(); //calling heading

cout<<"Please select any.\n\n"; //taking input

cout<<"\t1) Login";

cout<<"\t\t2) Exit \n\n";

cout<<"==> ";

cin>>select;

system("cls"); //system clear command

heading();

if(select == 1) //if start for the specific selection

{

cout<<"\n\tUSERNAME: "; //taking input

cin>>username;

cout<<"\tPASSWORD: ";

cin>>password;

if(username=="admin" && password=="pass") //if condition for correct password and id

{

char term='n';

while(term!='y') //while loop to run or display coices again and again till find

{

plan\_change: //part of goto funtion command

system("cls");

int elect = 0;

char con;

heading(); //calling function

choice(); //calling function

cin>>elect;

system("cls");

heading(); //calling function

switch (elect)

{

case 1:

obj.addBooks(); //calling function

break;

case 2:

bk\_header(); //calling function

obj.AllBooks(); //calling function

break;

case 3:

obj.FindBooks(); //calling function

break;

case 4:

oobj.IssBooks(); //calling function

break;

case 5:

isbk\_header(); //calling function

oobj.All\_issBooks();

break;

case 6:

ooobj.Reg\_std(); //calling function

break;

case 7:

std\_header(); //calling function

ooobj.Allstd(); //calling function

break;

case 8:

cout<<"\nConfirm Termination! with 'y': ";

cin>>con;

if(con=='y')

term = 'y'; //to end up the loop that displays choices

else

goto plan\_change; //to go to its part up there somewher

}

system("cls");

}

}

else

{

cout<<"Invalid Entry \n";

system("pause");

system("cls");

}

}

else if(select == 2) //if you want to exit

{

cout<<"Confirm Exit! (press y to confirm) ";

cin>>confirm;

if(confirm == 'y')

done = 'y'; //to end up the loop

else

goto re\_enter;

}

else

{

cout<<"Invalid Entry \n";

system("pause");

system("cls");

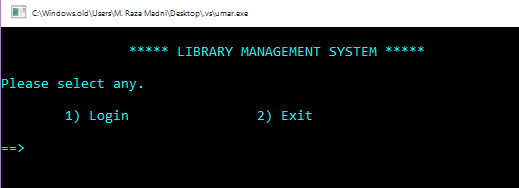
goto re\_enter; //to go to its part up there somewher

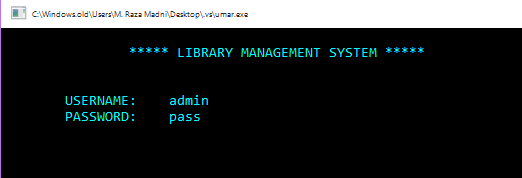
}

}

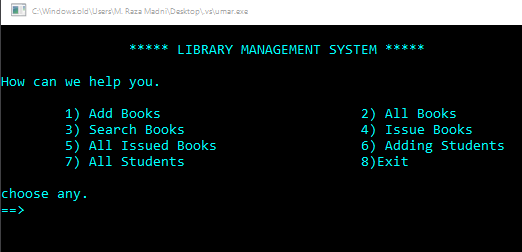
}

* **Result:**
* Log in.

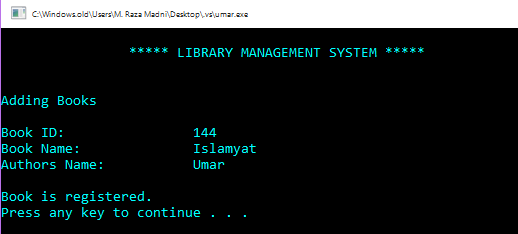




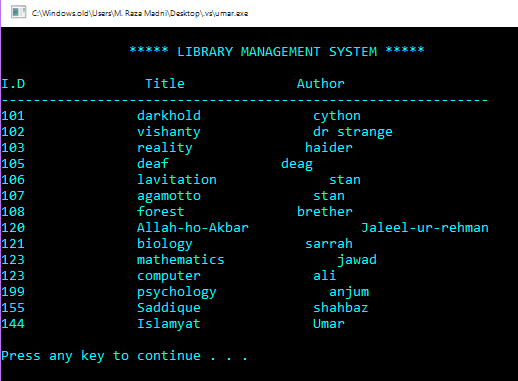
* Main menu.



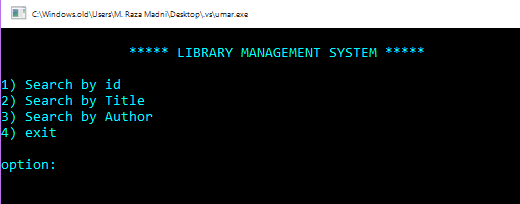
* + Add Books



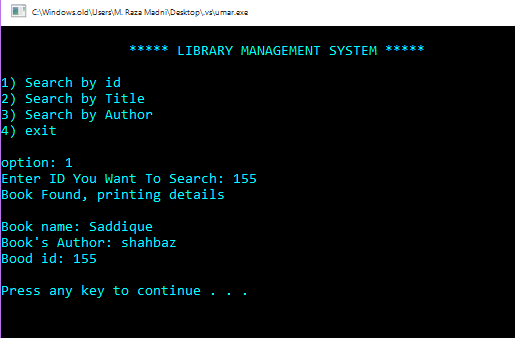
* + All Books



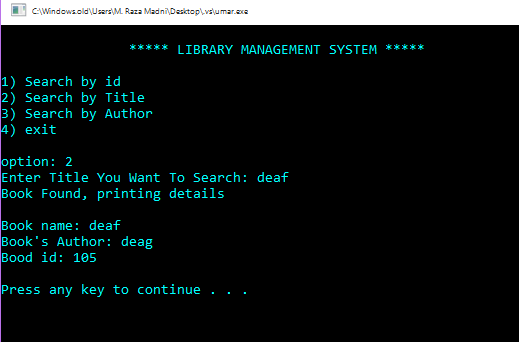
* + Search Books



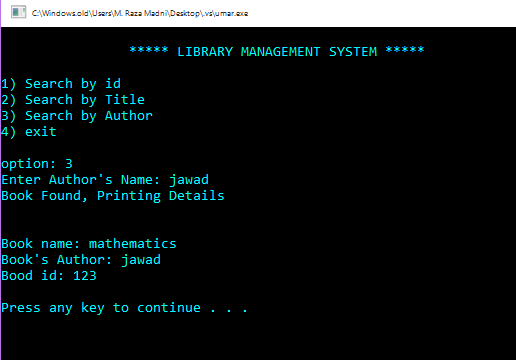
* + - By ID



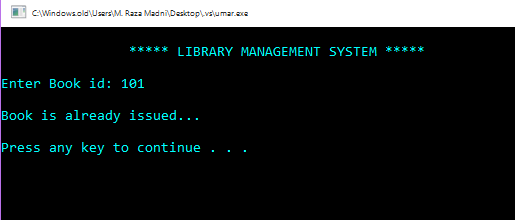
* + - By Title



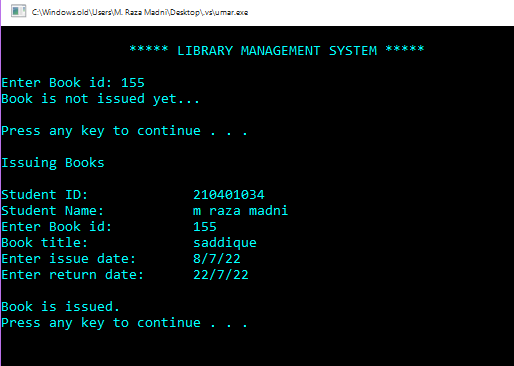
* + - By Author



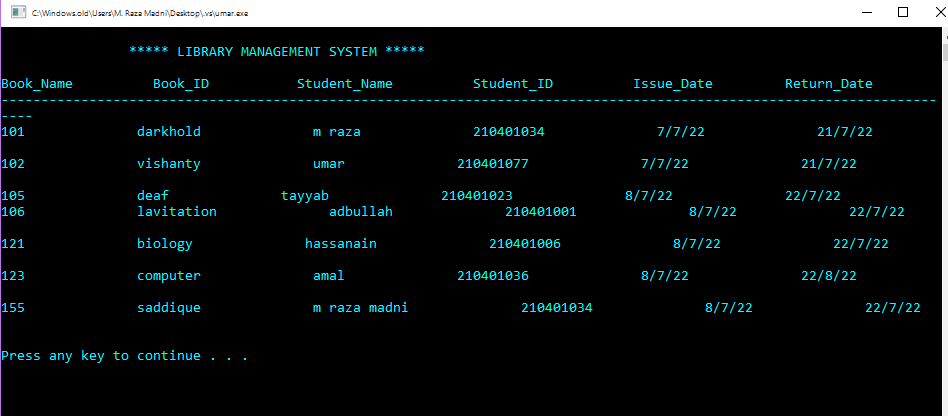
* + Issue Books
    - If already issued



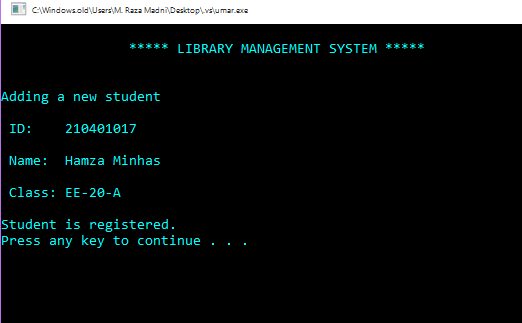
* + - If not



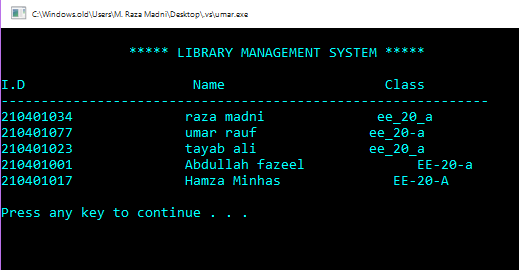
* + All Issued Books



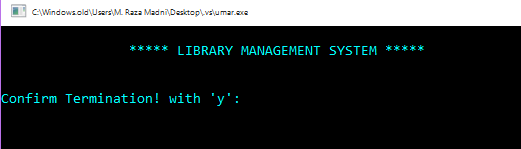
* + Add students

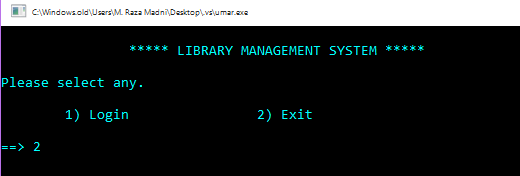


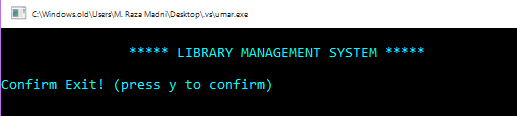
* + All students

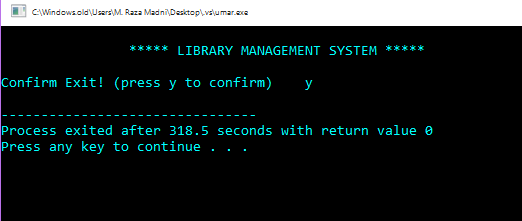


* + Exit









<https://replit.com/@RazaMadni/UnsteadyLovableAbility#main.cpp>