NAZARBAYEV UNIVERSITY

School of Engineering and Digital Sciences

Library Management System

A project submitted
in partial fulfilment of the requirements for the ENG101 course in SEDS

by

Daniyar Zhakyp
(ID: 201774605)

Kaisar Tuguzbayev
(ID: 201810778)

Alisher Turganbay
(ID: 201860075)

Aisar Abdrakhmanov
(ID: 201868460)

Evaluated by

Vipin Kinzhepatt
Amin Zollanavari
Carlo Molardi
Aresh Dadlani

Fall 2019

Abstract

The main purpose of this project was to develop a console-based Library management system (LMS) in the C++ programming language. The basic concepts of Object-Oriented Programming (OOP) taught in ENG 101 course lessons were used to create this program. The project was created on cross-platform IDE "CodeBlocks", no other software was used. As a database, this program uses "lmsdb.xlx" file provided by the course instructors which contains data about some in NU library. Almost all of the project objectives were achieved because the program performs all the functions described in the project manual. The possible drawback is the absence of GUI design. This is due to the complexity of the development of design in this programming language. However, the program has decent functionality and works without any errors.

Outline

1. OOP concepts

Object-oriented programming has many advantages over functional-oriented and other types of programming. OOP provides us with the ability to create objects that combine properties and behaviour into an independent union, which can be reused many times with the same program code. Instead of focusing on writing functions, we can focus on defining the objects which have a clear set of behaviours. Here the main OOP concepts:

Class and objects:

Class and objects are the basic components of any OOP program. Our program contains 2 classes: “Book” and “Library”. Every single book is defined as an object.

Encapsulation

If the code has classes and objects it should also have the encapsulation. Encapsulation is an important concept in OOP, as it helps us to combine the data as a single unit.
The class “Book” is a very good example of how encapsulation was performed in our code. This class contains several variables and methods. All the properties of the book are set as variables in this class. Here the list of variables:

- AuthorName
- BookTitle
- YearPublished
- ISBN
- Publisher
- LLC
- InStock (number of books available in the stock)

The methods used are the actions that can be performed with books in the library. They are:

- Issue
- Return
- Print

Inheritance

Our code does not contain the features of inheritance. For simplicity and our convenience, we made the variables of the Book class public and the minimum list of simple functions were written in there.

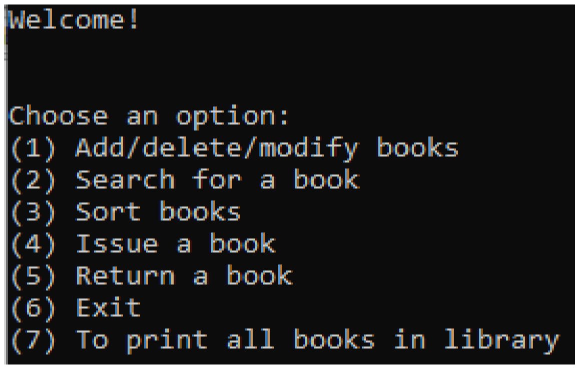
Polymorphism

Polymorphism is tightly connected to the concept of inheritance. We did not use it as we have its own special role for each function. Each of the objects like books and mainLibrary uses the set of functions which is pertinent to them.

2. Content

2.1 General overview

In this part simple interface displayed as in Figure 2.1.1 which welcomes user and provides with seven different options from which to choose, “Add/delete/modify books”, “Search for a book”, “Sort books”, “Issue a book”, “Return a book”, “Exit”, “To print all books in library”. User should enter the number of the option and then press “Enter” to continue.

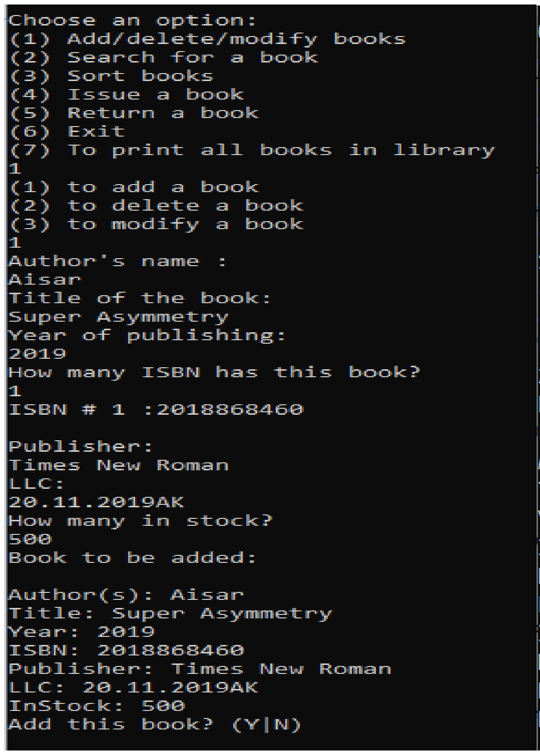


```
Welcome!


Choose an option:
(1) Add/delete/modify books
(2) Search for a book
(3) Sort books
(4) Issue a book
(5) Return a book
(6) Exit
(7) To print all books in library
```

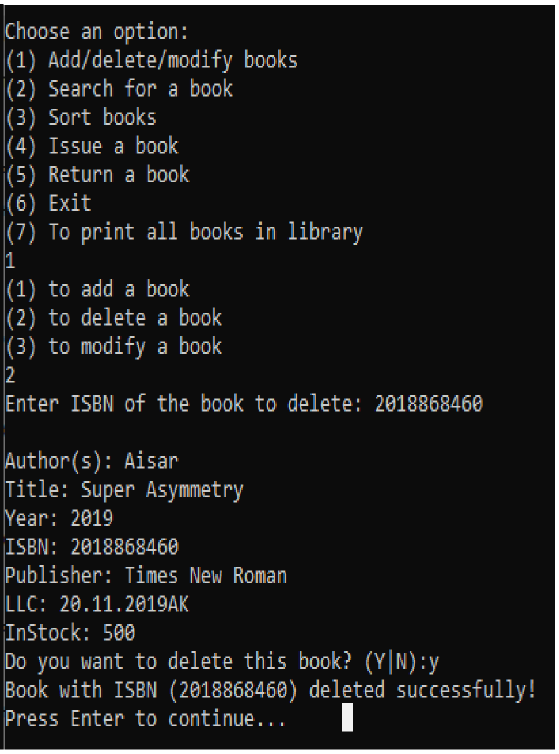
Figure 2.1.1 Main Menu

2.2 Add/delete/modify books

User is able to add/delete/modify books. First, to add books user should press “1” in the main menu, then press “1” again. After that, there are some fields that should be filled. As shown in figure 2.2.1, “Author's name”, “Title of book”, “Year of publishing”, “How many ISBN has this book?”, “ISBN”, “Publisher”, “LLC”, “In stock”, after all the fields are filled user should press “y” or “n” to add the book or not. Moreover, it is possible to delete a book from the library. The procedure is the same as adding the book. First, to delete the book user should press “1” in the main menu, then press “2” as in figure 2.2.2. It is compulsory to know the ISBN to delete the book. User should input ISBN of the book he wants to delete. After that user should press “y” or “n” to delete the book or not.



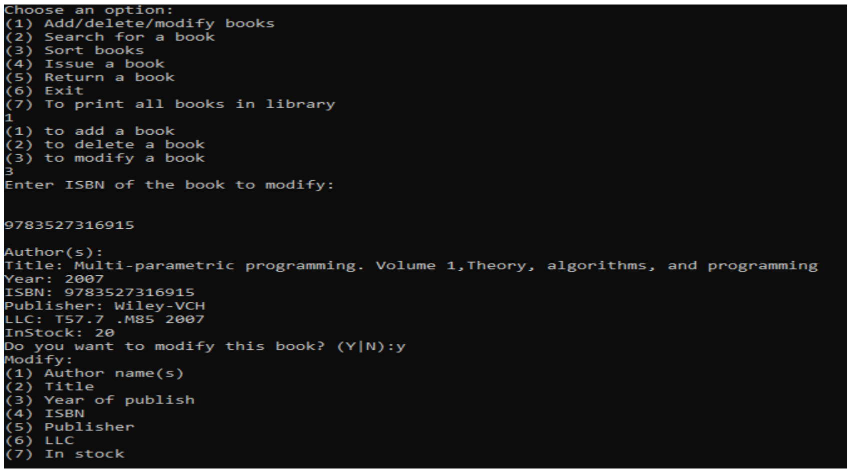
```
Choose an option:
(1) Add/delete/modify books
(2) Search for a book
(3) Sort books
(4) Issue a book
(5) Return a book
(6) Exit
(7) To print all books in library
1
(1) to add a book
(2) to delete a book
(3) to modify a book
1
Author's name :
Aisar
Title of the book:
Super Asymmetry
Year of publishing:
2019
How many ISBN has this book?
1
ISBN # 1 :2018868460

Publisher:
Times New Roman
LLC:
20.11.2019AK
How many in stock?
500
Book to be added:

Author(s): Aisar
Title: Super Asymmetry
Year: 2019
ISBN: 2018868460
Publisher: Times New Roman
LLC: 20.11.2019AK
InStock: 500
Add this book? (Y|N)
```



```
Choose an option:
(1) Add/delete/modify books
(2) Search for a book
(3) Sort books
(4) Issue a book
(5) Return a book
(6) Exit
(7) To print all books in library
1
(1) to add a book
(2) to delete a book
(3) to modify a book
2
Enter ISBN of the book to delete: 2018868460

Author(s): Aisar
Title: Super Asymmetry
Year: 2019
ISBN: 2018868460
Publisher: Times New Roman
LLC: 20.11.2019AK
InStock: 500
Do you want to delete this book? (Y|N):y
Book with ISBN (2018868460) deleted successfully!
Press Enter to continue...
```

Figure 2.2.1 Add a book　　　　*Figure 2.2.2 Delete a book*

Also, the program allows to modify books. First, to modify books user should press “1” in the main menu, then press “3”. It is compulsory to know the ISBN to modify the book. User should input ISBN of the book he wants to modify. Then, the program asks “Do you want to modify this book?” after that user should press “y” or “n” to modify the book or not. As shown in figure 2.2.3, user can modify “Author's name”, “Title of book”, “Year of publishing”, “How many ISBN has this book?”, “ISBN”, “Publisher”, “LLC”, “In stock”. After that user should press “y” or “n” to modify the book or not.

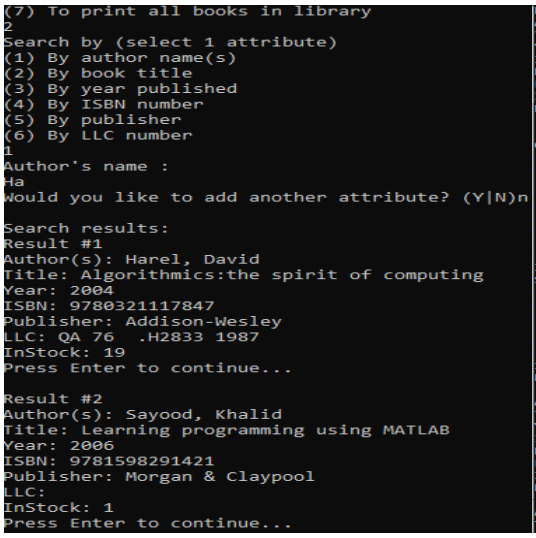


```
Choose an option:
(1) Add/delete/modify books
(2) Search for a book
(3) Sort books
(4) Issue a book
(5) Return a book
(6) Exit
(7) To print all books in library
1
(1) to add a book
(2) to delete a book
(3) to modify a book
3
Enter ISBN of the book to modify:


9783527316915

Author(s):
Title: Multi-parametric programming. Volume 1,Theory, algorithms, and programming
Year: 2007
ISBN: 9783527316915
Publisher: Wiley-VCH
LLC: T57.7 .M85 2007
InStock: 20
Do you want to modify this book? (Y|N):y
Modify:
(1) Author name(s)
(2) Title
(3) Year of publish
(4) ISBN
(5) Publisher
(6) LLC
(7) In stock
```

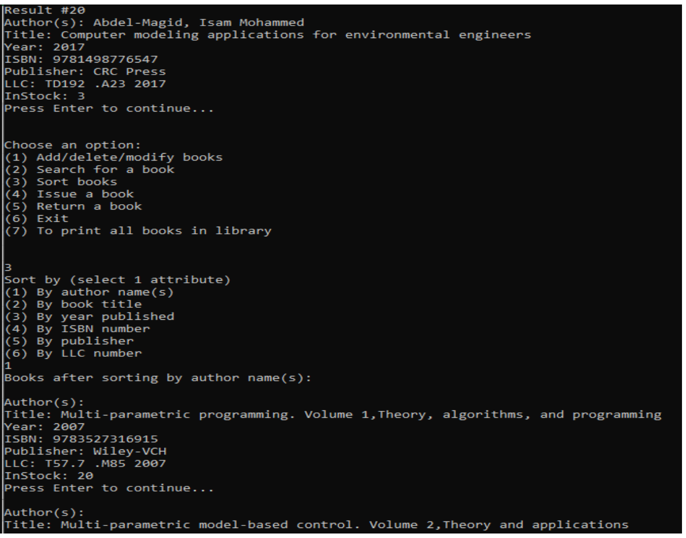
Figure 2.2.3 Modify books

2.3 Search a book/Sort books/ /Issue a book/ Return a book

User can search a book by 6 parameters: “ by author's name”, “ by title of book”, “ by year of publishing”, “by ISBN number”, “ by publisher”, “by LLC number”. If a user chooses to search by author’s name or title and input some letters, the program will find books by keys as shown in figure 2.3.1. Otherwise, the user should input whole ISBN number, LLC number, year of publishing and publisher. Also, is possible to sort a book by 6 parameters. First, to do that user should press “2” in the main menu, and choose 6 different parameters. “ by author's name”, “ by title of book”, “by year of publishing”, “by ISBN number”, “ by publisher”, “by LLC number”. If the user chooses to sort by “by author's name”, it will be sorted in alphabetical order and those books that don’t have an author would appear first as shown in figure 2.3.2. As in regular library, you can issue or return a book. To do that user should press “4” to issue a book or “5” to return a book and then choose a book by ISBN as shown in figure 2.3.3

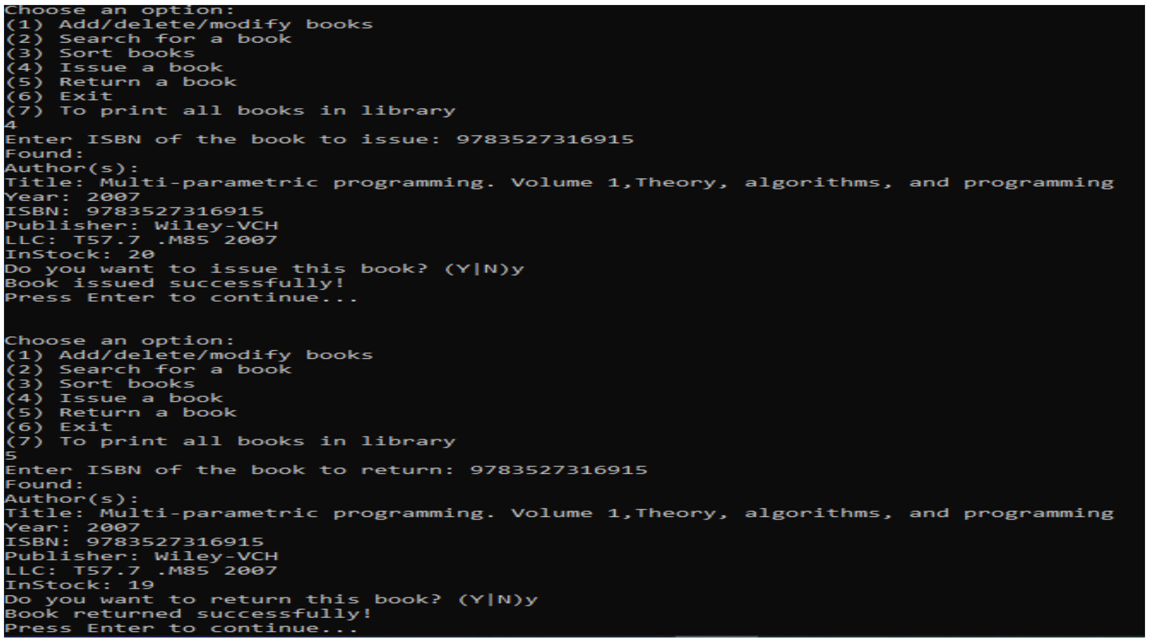


```
(7) To print all books in library
2
Search by (select 1 attribute)
(1) By author name(s)
(2) By book title
(3) By year published
(4) By ISBN number
(5) By publisher
(6) By LLC number
1
Author's name :
Ha
Would you like to add another attribute? (Y|N)n

Search results:
Result #1
Author(s): Harel, David
Title: Algorithmics:the spirit of computing
Year: 2004
ISBN: 9780321117847
Publisher: Addison-Wesley
LLC: QA 76  .H2833 1987
InStock: 19
Press Enter to continue...

Result #2
Author(s): Sayood, Khalid
Title: Learning programming using MATLAB
Year: 2006
ISBN: 9781598291421
Publisher: Morgan & Claypool
LLC:
InStock: 1
Press Enter to continue...
```



```
Result #20
Author(s): Abdel-Magid, Isam Mohammed
Title: Computer modeling applications for environmental engineers
Year: 2017
ISBN: 9781498776547
Publisher: CRC Press
LLC: TD192 .A23 2017
InStock: 3
Press Enter to continue...


Choose an option:
(1) Add/delete/modify books
(2) Search for a book
(3) Sort books
(4) Issue a book
(5) Return a book
(6) Exit
(7) To print all books in library


3
Sort by (select 1 attribute)
(1) By author name(s)
(2) By book title
(3) By year published
(4) By ISBN number
(5) By publisher
(6) By LLC number
1
Books after sorting by author name(s):

Author(s):
Title: Multi-parametric programming. Volume 1,Theory, algorithms, and programming
Year: 2007
ISBN: 9783527316915
Publisher: Wiley-VCH
LLC: T57.7 .M85 2007
InStock: 20
Press Enter to continue...

Author(s):
Title: Multi-parametric model-based control. Volume 2,Theory and applications
```

Figure 2.3.1 Search a book by an author's name *Figure 2.3.2 Sort a book by an author's name*

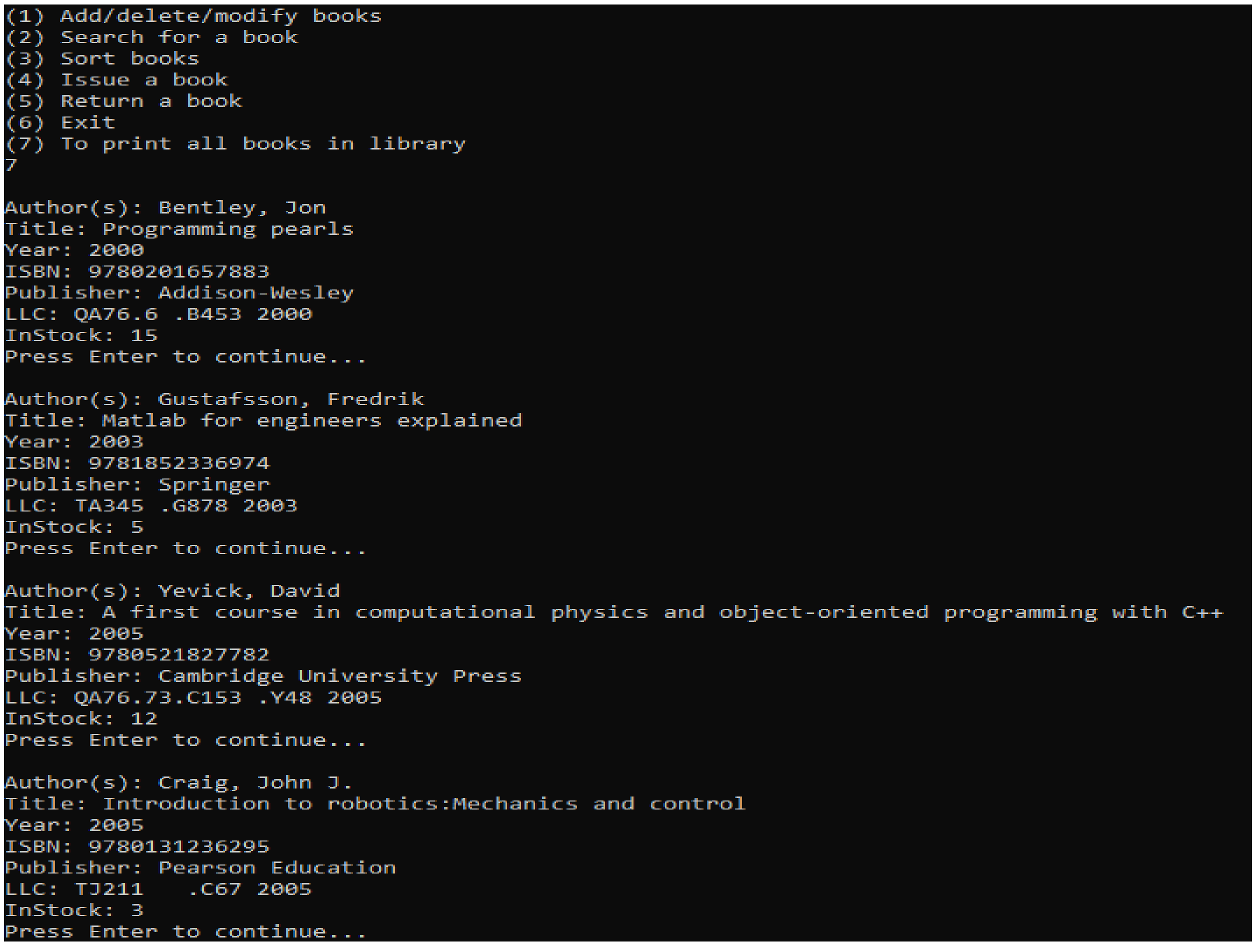


```
Choose an option:
(1) Add/delete/modify books
(2) Search for a book
(3) Sort books
(4) Issue a book
(5) Return a book
(6) Exit
(7) To print all books in library
4
Enter ISBN of the book to issue: 9783527316915
Found:
Author(s):
Title: Multi-parametric programming. Volume 1,Theory, algorithms, and programming
Year: 2007
ISBN: 9783527316915
Publisher: Wiley-VCH
LLC: T57.7 .M85 2007
InStock: 20
Do you want to issue this book? (Y|N)y
Book issued successfully!
Press Enter to continue...


Choose an option:
(1) Add/delete/modify books
(2) Search for a book
(3) Sort books
(4) Issue a book
(5) Return a book
(6) Exit
(7) To print all books in library
5
Enter ISBN of the book to return: 9783527316915
Found:
Author(s):
Title: Multi-parametric programming. Volume 1,Theory, algorithms, and programming
Year: 2007
ISBN: 9783527316915
Publisher: Wiley-VCH
LLC: T57.7 .M85 2007
InStock: 19
Do you want to return this book? (Y|N)y
Book returned successfully!
Press Enter to continue...
```

Figure 2.3.3 Issue and return a book by an author's name

2.4 Print all books in the library

Last option to choose for users is to print all books in the library. To print all books user should press “7” in the main menu and press enter to see each book an in figure 2.4. Pressing enter is the only method to see all books in the library



```
(1) Add/delete/modify books
(2) Search for a book
(3) Sort books
(4) Issue a book
(5) Return a book
(6) Exit
(7) To print all books in library
7

Author(s): Bentley, Jon
Title: Programming pearls
Year: 2000
ISBN: 9780201657883
Publisher: Addison-Wesley
LLC: QA76.6 .B453 2000
InStock: 15
Press Enter to continue...

Author(s): Gustafsson, Fredrik
Title: Matlab for engineers explained
Year: 2003
ISBN: 9781852336974
Publisher: Springer
LLC: TA345 .G878 2003
InStock: 5
Press Enter to continue...

Author(s): Yevick, David
Title: A first course in computational physics and object-oriented programming with C++
Year: 2005
ISBN: 9780521827782
Publisher: Cambridge University Press
LLC: QA76.73.C153 .Y48 2005
InStock: 12
Press Enter to continue...

Author(s): Craig, John J.
Title: Introduction to robotics:Mechanics and control
Year: 2005
ISBN: 9780131236295
Publisher: Pearson Education
LLC: TJ211   .C67 2005
InStock: 3
Press Enter to continue...
```

Figure 2.4 Print all books in the library

3. Discussion

3.1 Challenges

We faced two main challenges. One of them is that we do not know how to use lmsdb.xls file. We cannot link the .xls Excel file to the code and get data from it. The solution to this challenge was to use another format. We converted .xls Excel file to a new format, which is called .csv format (comma-separated values). The main advantage of this format is that the values in this file are separated by a comma, which is very useful since we were able to know the length of each value by checking whether there is a comma or not. If there is a comma, it means that value is finished, and we can go to the next. If there is not a comma, it means that value is not finished.

The second challenge is to write the code for searching. "Searching for a book" is the second option in the main menu. In the beginning, a programme indicates books that satisfy certain criteria randomly. It was not so convenient for user interaction. The program has to indicate the best-fitted books at first for a better user interaction. The solution to this challenge was to create a new variable, which is called “similar”. “Similar” is a variable which increases every time when a book is more fitted with certain criteria. It means that the more value of “similar”, the more fitted book. Then we made such a program, which allows sorting “similar” values in descending order. In the result, the program indicates the best-fitted books at first.

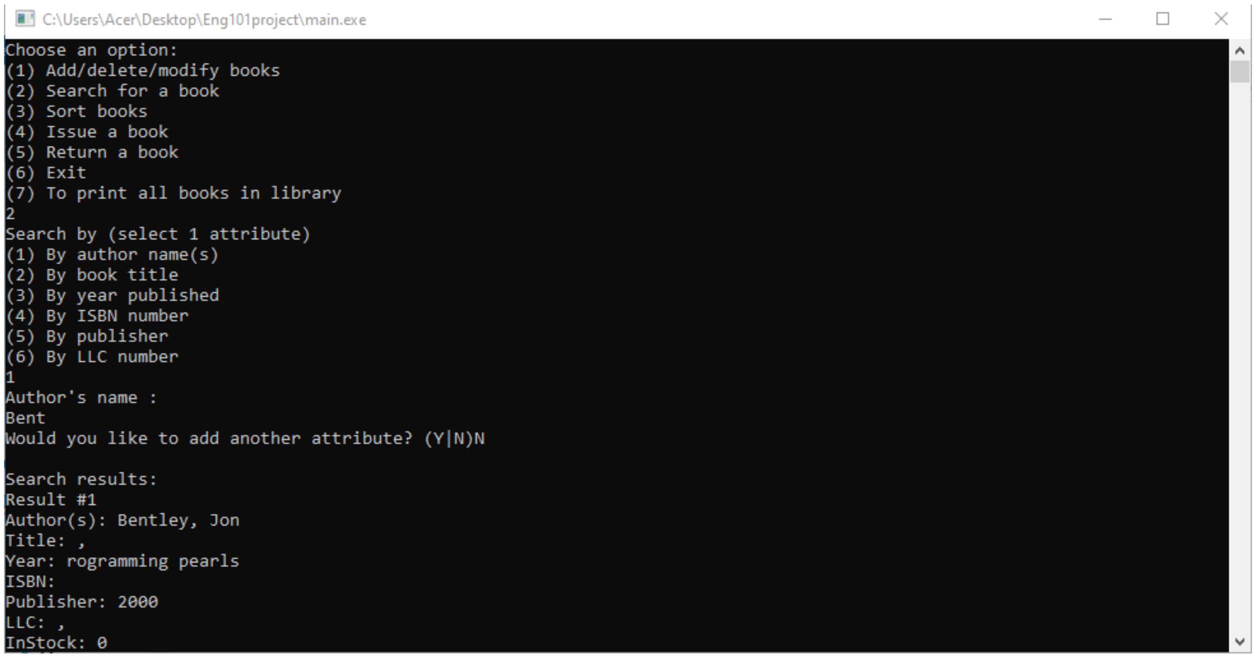
3.2 Evaluation

The evaluation of the project was done by considering its strengths and weaknesses:

3.2.1 Strengths

The first strength is the class library. The advantage of a class library is that there is an opportunity to create another base of the library. Thus, two or even more bases can exist in one single code and they can work separately from each other.

The second strength is the convenience of searching. It is convenient to search because it is possible to search by a “key words”. It is enough to write only some letters.



```
Choose an option:
(1) Add/delete/modify books
(2) Search for a book
(3) Sort books
(4) Issue a book
(5) Return a book
(6) Exit
(7) To print all books in library
2
Search by (select 1 attribute)
(1) By author name(s)
(2) By book title
(3) By year published
(4) By ISBN number
(5) By publisher
(6) By LLC number
1
Author's name :
Bent
Would you like to add another attribute? (Y|N)N

Search results:
Result #1
Author(s): Bentley, Jon
Title: ,
Year: rogramming pearls
ISBN:
Publisher: 2000
LLC: ,
InStock: 0
```

Figure 3.2.1 Key words

The third strength is the convenience of user interaction. “To print all books in library” is the seventh option in a code. In order to see the next book, you have to press a button of “Enter” which leads to the continuation, that is, displaying each following book. It is convenient for a user to follow each book and its corresponding information (author's name, title of a book, published year, ISBN, publisher's name, LLC and number of books available in stock). So after each book, there is written: "Press Enter to continue…".

3.2.2 Weaknesses

The first weakness is design. There are not any changes in the design as a decent GUI (graphical user design) design. Thus, it is not user-friendly and easy to use.

The second weakness is a code length. It is too long for this kind of library. It can be written in a shorter code. There is a possibility to get confused by such length and such number of characters.

The third weakness is related to OOP concepts. As it is object-oriented programming, it would be an exemplar to use all four concepts. However, there are only two of them such as encapsulation and polymorphism, in addition classes and object, obviously. Nevertheless, it is not such a big deal since there is no need for inheritance and abstraction and the code does not require them.

3.3 Special features

1) Vector function

We literally built our code on the **vector** functions**.** We used vector as a container where all books are stored. We preferred the **vector** class than the standard array as we change the number of books by adding or modifying their features, so we need to allocate the memory dynamically. The big advantage is that the **vector** class has all the necessary functions for our code such that it is enough to use them in order to complete the library management system.

So there are many functions that we used in the code related to the vector class. They are:

Push_back - used when we want to add the book or one of its features to the vector.

Erase - we used it to get rid of some useless and hindering characters like a comma before a word or a special symbol before a year.

Swap - swap functions was used when we wanted to sort the books using the bubble sort.

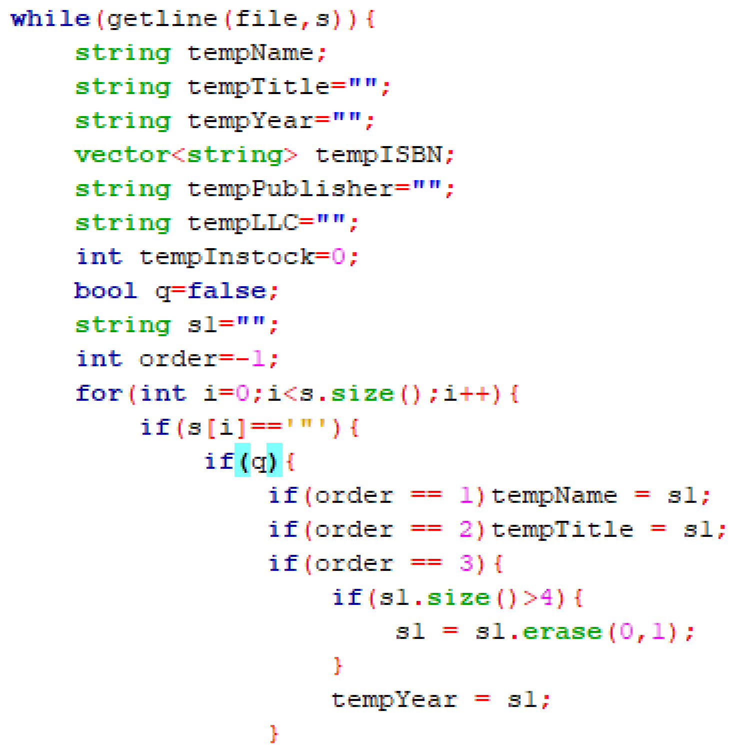
Find and npos - where find gave the position(line) of the book which we are searching for and npos returns us -1 when the position hasn't been found.

Begin - used when we wanted to return an iterator to the beginning and delete a book.

Size - to return the size of the array (number of books).

2) Parsing the document

First, it needs to be said the format of the file containing all the books was changed from .xlsx into .csv (comma separated value) - the format which prints out the information separated by commas and which can be recorded by CodeBlocks and other IDEs. We used the method **fstream** to open our .csv file and read out the data from it. We also used the command **getline** which allows us to read the document line by line until the end of each line.



```
while(getline(file,s)){
    string tempName;
    string tempTitle="";
    string tempYear="";
    vector<string> tempISBN;
    string tempPublisher="";
    string tempLLC="";
    int tempInstock=0;
    bool q=false;
    string s1="";
    int order=-1;
    for(int i=0;i<s.size();i++){
        if(s[i]=='"'){
            if(q){
                if(order == 1)tempName = s1;
                if(order == 2)tempTitle = s1;
                if(order == 3){
                    if(s1.size()>4){
                        s1 = s1.erase(0,1);
                    }
                    tempYear = s1;
                }
```

That is how it looks inside the code. We defined each feature of the book with the int variable order, where order==1 is the name of the book, order==2 is the title and so on. Using if-else statement we made a condition that if the program sees “ symbol, it will start assigning the temp variables of features to the line that will be then included into the program. Else it will increment the order from -1 to 0 until it finds the “ symbol.

3) Using flag

We thought that the flag usage can be a special feature as the code can be written without it, but we find it convenient and quite understandable technique. We used the flag when in the case when a user wants to delete a particular book. It can be done by entering the ISBN number, so our flag will show the position of the book with at least one same ISBN number (it can be more than for some books) and via the if-else statements call the vector function erase to delete the book.

```
void remove(string isbn){
    int pos = -1;
    for(int i=0;i<books.size();i++){
        Book temp = books[i];
        for(int j=0;j<temp.ISBN.size();j++){
            if(temp.ISBN[j]==isbn){
                temp.print();
                pos = i;
            }
        }
    }
    if(pos == -1){
        cout<<"No results!"<<endl;
    }
    else {
        cout<<"Do you want to delete this book? (Y|N):";
        char ans;
        cin>>ans;
        if(ans == 'Y' || ans == 'y'){
            books.erase(books.begin()+pos);
            cout<<"Book with ISBN ("<<isbn<<") deleted successfully!\n";
        }
```

4) Other special techniques

A)For the Search action, we used a non-trivial way to initialize an int variable called **similar**. We created a vector pair of the int similar and the book we want to find. Then our line of words or numbers or the keywords are assumed to be put into each of the lines excerpted from the .csv file, it starts finding the common words or numbers. If they exist, the program increments the variable int similar by one each time it finds new similarities. Then using the code below, it sorts the found books by the number of similarities: the first one is the most suitable option.

```
if(foundBooks.size()==0)return result;
for(int i=0;i<foundBooks.size()-1;i++){
    for(int j=i+1;j<foundBooks.size();j++){
        if(foundBooks[i].first < foundBooks[j].first){
            swap(foundBooks[i], foundBooks[j]);
        }
    }
```

B) We have a separate function **tolowerString** which converts the capital letters into lowercase letters and vice versa.

C) We also used a command **cin.ignore()** in order to clear the previous variables and be able to enter a new one.

4. Major contribution

Name of the member	**Part in the code**
Daniyar Zhakyp	The flow of the program+code for reading the document
Kaisar Tuguzbayev	Sort + Return/Issue
Aisar Abdrakhmanov	Add/Delete/Modify
Alisher Turganbay	Search