

library borrowing system

Description

The program in this assignment is a **library borrowing system** that can be used by any university member to borrow books. The library staff in a university will use the system to:

- (1) Add book details in the system,
- (2) Add member details in the system,
- (3) Allow members to borrow books based on the request.

To use the library booking system, library staff has to set **initial parameters** one time. These parameters are the **total number of books** the system has to manage, the **total number of members** the system has to create records for, and the **number of borrowing books allowed** the system has to make [see sample output file].

The Initial Procedure of the Program

To set the **initial parameters**, your program will **first** read the following parameters from [input.txt]:

1. **Total Number of Books**
2. **Total Number of Members**
3. **Number of Borrowing Books allowed**

These parameters will be in the first line of the file [input.txt]. To read from the file, the system should first check if the file [input.txt] exist or not and will display an error message if the file does not exist.

- The first number (4) in the first line determines the **Total Number of Books**
[this means the system will have the maximum of THREE Books Categories]
- The second number (3) in the first line determines the **Total Number of Members**
[this means the system will have the maximum of 3 Members]
- The third number (3) in the first line determines the **Number of Borrowing Books allowed**
[this means the system will allow for three books to be borrowed by the members and the maximum number of Borrowing will be $\text{Number of Members} * \text{Number of Borrowing Books allowed}$]

The system (your program) then will read the following command to perform one of the programs methods:

Add_Book

Using this command, the system will add the **book details** based on the **total number of books** available in the system [see sample output file].

Example

Add_Book CS202 JAVA Liang 5

In this line **Add_Book** is the command and **CS202 JAJA Liang 5** are the details of the book where **CS202** is the book no, **JAVA** is the book title, **Liang** is author name, and **5** is the number of book's copies [see Table1].

Note that the system will not allowed more than **Total Number of Books** to be registered. If this happen, a message that says "*Book <book no> Was not Added. You exceed the maximum number of Books* " will be displayed.

Add_Member

Using this command, the system will add the member details taking into account the **total number of members** allowed in the system [see sample output file].

Example

Add_Member Turki_Ahmed 12345 Faculty

In this line **Add_Member** is the command and **Turki_Ahmed 12345 Faculty** are the details of the member where **Turki_Ahmed** is the name, **12345** is the member id, and **Faculty** is the member type (**Note: There are two types of members: Faculty and other**).

Note that the system will not allowed more than **Total Number of Memebrs** to be registered. If this happen, a message that says "Member **Member_name** Was not Added, You exceed the maximum number of Members" will be displayed.

Borrow

Through this command, the system will create a new borrowing for a member based on the **number of borrowing requests** allowed in the system [see sample output file].

Example

Borrow 23156 2 CS204 CS211

[**Borrow** member_Id number_of_borrowing_books book1_No book2_No....]

In the above command the system will borrow a 2 books for the member 23156.

Notes that when the system **borrow a new books**:

- The system will first check the **member id** that the member wants to borrow. If the member id is not in the system, the system should display an error message that says, “**Member 19678 was not registered in the System**”.
 - The system will check the **books’ no** to be borrowed in current borrowing request [see sample output file]. If any one of the books is not in the system, the system will display an error message that says, “**No Record found with the Book Code book_no**”.
 - If the number of requested copies exceed the number of remaining books for the required books, the system will display an error message that says, “**books requested exceed the number of remaining books**”.
 - If the number of requested copies plus to the already borrowed exceed the number of maximum borrowing allowed, the system will display an error message that says, “**---The required plus the previous borrowed books is exceeded the maximum of Allowed**”.
4. The system will not allow more than **Number of Borrowing Books allowed** to be registered. If this happen, an error message that says, “**You exceeded the number of borrowing requests**” will be displayed.

Finally, the method will generate and return a unique borrowing record (BRN) for every borrowing, BR must start from BR1000 onwards, i.e., First borrowing BR1000, next borrowing BR1001, Next BR1002, etc.

Search_Borrowing

This command will **search and print a borrowing** information given a BRN code [see sample output file]. This option will show complete details of the borrowing, which includes the books details, the member name, date of borrowing, and number of borrowing books [see sample output file]. Note that this command will return an error message if the BRN is not found. For example, “**No Record found with the BR BB1000**”

Example

Search_Borrowing BB1000

This command will show complete details of the borrowing code BR1000

Book_Status

This command will print / display complete book status and details which includes title, author, etc. Note that this command will return an error message if the book number is not found. For example, “**Book number book_no is not found in the system**” [see sample output file].

Example

Book_Status CS202

This command will display the complete information of the book CS202.

Class Description:

You have to create the following base classes in this program.

- **Book** class
- **Member** class
- **Borrowing** class

1. The Data fields in the **Book** class are as follows.

fbookNo	CS202	title	JAVA	author	Laing
numCoies	5	remCopies	2		

2. The Data fields in the **Member** class are as follows.

Name	Abdullah Ali	id	12876	Type	Faculty
-------------	--------------	-----------	-------	-------------	---------

3. The Data fields in the **i** class are as follows.

BRNNumber	BR1000	borrowingDate	2020-01-26
member	Object from Member class	Books []	Array of Books as there might be more than one books borrowing by the same member.
numBookBorrowing	2		

You program also need to include a class called **AAR134343P2BookBorrowing.java** class to test the Borrowing System. This is your main class in the application. Note that **[AAR134343P2]** **will be different from student to student as it is sectionStudentIDP2.**

See the UML diagram to know complete data members and methods required in each class.

The UML Diagram

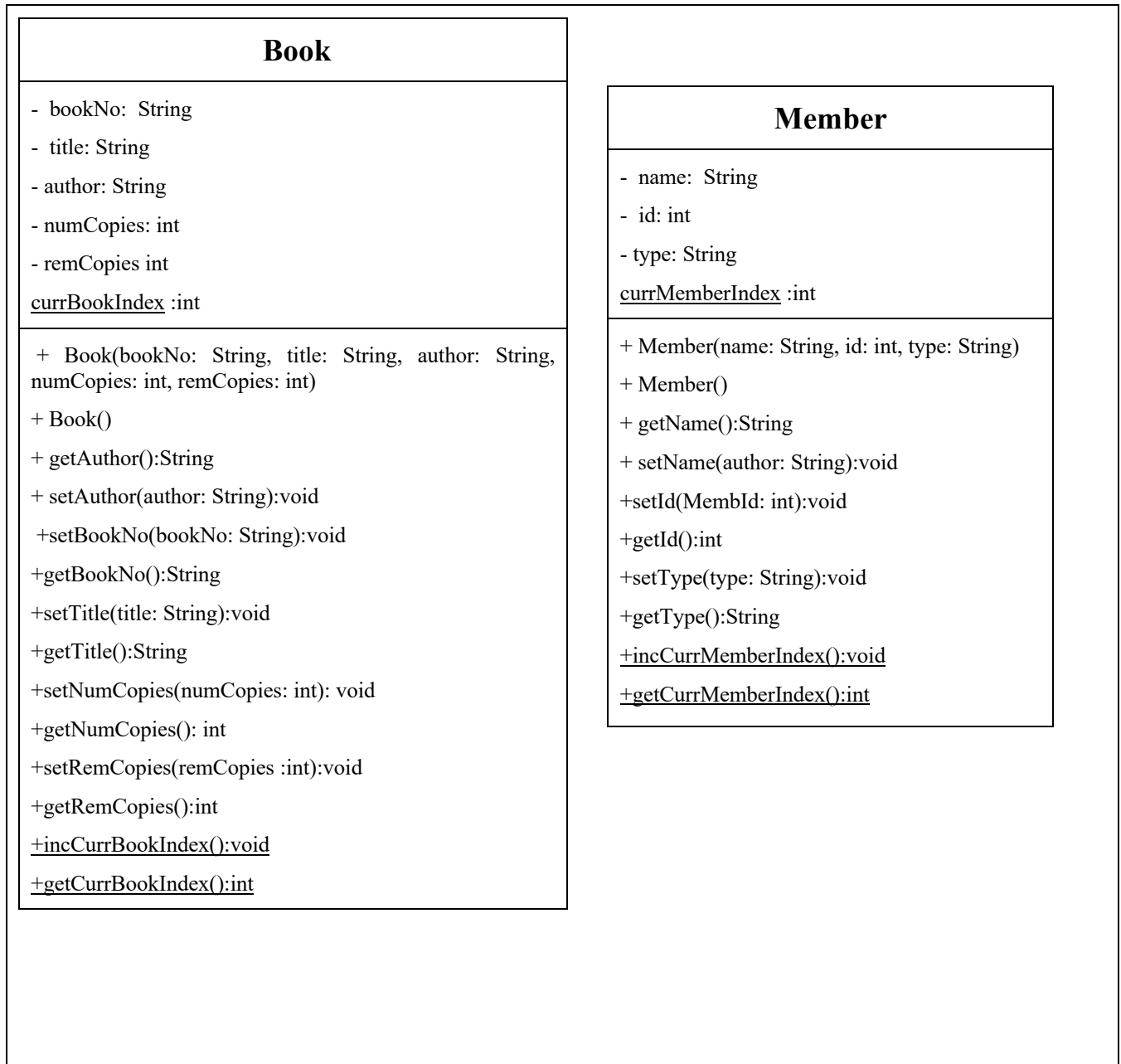




Figure 1: UML Diagram Library Borrowing System

Arrays and their description:

You will have to create the following arrays:

- ✓ **Book [] books**, which is an array of **Book** objects.
- ✓ **Member [] members**, which is an array of **Member** objects.
- ✓ **Borrowing [] borrowing**, which is an array of **Borrowing** objects.

Methods and their description:

a) **inputBook**

This method will be used to read the book details from the input file and add the book details to the system. System only allows limited book entry [see sample output file].

b) **inputMember**

This method will be used to read the member details from the file and enter member details to the system. System only allows limited member entry [see sample output file].

c) **BorrowingBook**

This method will allow the members to borrow based on the details from the file. System only allows limited number of borrowing requests entry [see sample output file].

d) **BookStatus**

This method will search and print the details of a book given its book no. [see sample output file].

e) **BorrowingDetails**

This method will search and display the details of a single borrowing given its BRN code.

The functionality of the classes is shown in UML diagram, see Figure 1, mainly there are mutators(setter) and accessors (getter) in every class see Figure 1.