1. Project Overview:

Group Members: Calvin White, Pratham Snehi

Group Number: 12

The main goal of the project is to operate a library, where basic management operations (done by the library staff) like adding and removing books can be done, as well as member operations like checking out and returning a book can also be done.

Key functionalities:  
Adding and removing books as administrative functions (not available to the member), the member being able to borrow the book by searching through the titles of all available books, managing the isAvailable attribute of each book to ensure no unavailable book is checked out

OOP Concepts Used:  
splitting up all the functionalities required to run a library into different classes (like Book, Library, Member). Use of appropriate methods that can be used from other classes in order to make the library functionality work (like toString() method of is useful when we are trying to displayBooks(), which loops through the booksArray ArrayList, and converts them to strings). We also have used to objects concept of OOP, where we create a new library and member objects that represent different members and different libraries.

2. Class Descriptions

**Book Class**

Attributes:

title, author, isAvailable

Methods:

borrowBook() : Checks if Book is available, Then marks Book as unavailable if it was available.

returnBook() : Checks if Book is available, Then marks Book as available if it was unavailable. If available, then prints “book has already been returned”.

**Library Class**

Attributes:

ArrayList booksArray

Methods:

addBook(Book book) : Adds book to ArrayList, prints Successful

removeBook(String title) : Loops through the ArrayList, removes index of book, prints Successful

displayBooks() : For loop through the ArrayList, Prints books using toString() from Book

searchBook(String title) : Loops through ArrayList, if found return Book, else return null

**Member Class**

Methods:

borrowBook(Library library, String title) : Calls searchBook(title) from Library, then calls borrowBook() from Book Class.

returnBook(Library library, String title) : Calls searchBook(title) from Library, then calls returnBook() from Book Class.

**Main Class**

Details in 3. Implementation Details

3. Implementation Details:

(In order of execution of the main function):

1. We first create a library object that contains the basic functionalities like adding books, searching books, and listing all the books available
2. We then initialize a member, who can check out different books. When a member checks out a book, it marks the book as not available to check out anymore
3. We then create a couple of book objects that we then add to the library using the library.addBook(book object) method, which then adds the books to the arraylist present inside the library object
4. We then carry out member operations like checking out a book (which makes the book unavailable) and returning the book (which makes the book available).
5. We have implemented methods that helped us to test if books were added correctly or not, where we used the displayBooks() method in the library class, where it loops through all the book objects in the arraylist, and calls the toString() method on each book to print out their details

4. Test Cases  
The following test cases were tested, and gave successfully gave these outputs:  
  
Action | Expected Output  
Add a book | "The book was successfully added"  
Search a book | "Book was found"  
Borrow a book | title + " is now being borrowed"  
Return a book | title + " is now returned"  
Remove a book | "The book was removed successfully"

5. Challenges and Solutions:

Some of the challenges that we encountered were as follows:

1. We were having difficulties in the borrow and return book methods because we weren’t able to get hold of the book object whose isAvailable attribute we had to change. We overcame this issue by changing our searchBook() method in the library class, where it returned the book object if it managed to find the book. When we got the reference of the book object, it was simple to change the isAvailable attribute afterwards
2. We were unable to carry out the searchBook() method because in order to find the book by title, we had to make the title attribute of a book to be visible by the other class (which we fixed by using the default visibility), making it be accessible within the same package non subclass
3. While borrowing and returning books, we were simply setting the isAvailable to true or false without checking if the book was already borrowed or not. We fixed this challenge by implementing conditionals, which first checked if the book was available or not, and if it was only then it would make it checkout-able.

**6. Conclusion & Future Enhancements (Optional)**• What did you learn from this project?

From this project we learnt what it was like it be working on a multi class project with a partner, where we had to send each other files to test different methods and make sure everything was running smoothly.

• What improvements could be made? (E.g., "Add a GUI, implement a  
user login system.")

We could improve the program assigning the books borrowed to the members because as of now anyone can return a book and its not connected to who borrowed it. Also, it has no way of differentiating books with the same name, so if two authors had the same book its would be whichever was added first that gets borrowed. Also, adding a GUI could also be an improvement so that normal users who do not know how java works can still check out and return books.

7. References

1. [W3Schools](https://www.w3schools.com/java/)

2. [GeeksForGeeks Private, protected, default access modifiers](https://www.geeksforgeeks.org/protected-vs-private-access-modifiers-in-java/)