Library Management System –

Book Borrowing and Returning

# Aim

To design and implement an object-oriented program that allows a library member to borrow books and store the borrowed book titles in a list using constructors and appropriate member methods.

# Algorithm

* 1. Start the program.
  2. Create a Member class with attributes:
     + name (String)
     + borrowedBooks (List of Strings).
  3. Define methods:
     + borrowBook(String bookTitle) → adds a book to the borrowed list.
     + showBorrowedBooks() → displays all borrowed books.
  4. In the main method:
     + Create a Member object.
     + Borrow multiple books.
     + Display borrowed books.
  5. **Enhancement Made:**
     + Add a returnBook(String bookTitle) method.
     + This method checks if the book exists in the borrowed list.
     + If yes → remove it and confirm the return.
     + If no → display a message that the book wasn’t borrowed.
     + After returning, display the updated borrowed books list.
  6. End the program

# Java Code

**package** Library;

**import** java.util.ArrayList;

**import** java.util.List;

**class** Member {

**private** String name;

**private** List<String> borrowedBooks;

**public** Member(String name) {

**this**.name = name;

**this**.borrowedBooks = **new** ArrayList<>();

}

**public void** borrowBook(String bookTitle) { borrowedBooks.add(bookTitle); System.***out***.println(name + " borrowed: " + bookTitle);

}

**public void** returnBook(String bookTitle) {

**if** (borrowedBooks.contains(bookTitle)) { borrowedBooks.remove(bookTitle);

System.***out***.println(name + " returned: " + bookTitle);

} **else** {

System.***out***.println(name + " has not borrowed " + bookTitle);

}

}

**public void** showBorrowedBooks() { System.***out***.println("\nBooks borrowed by " + name + ":");

**if** (borrowedBooks.isEmpty()) {

System.***out***.println("No books borrowed yet.");

} **else** {

**for** (String book : borrowedBooks) { System.***out***.println("- " + book);

}

}

}

}

**public class** LibraryManagement {

**public static void** main(String[] args) { Member m1 = **new** Member("Rohitha"); m1.borrowBook("Harry Potter"); m1.borrowBook("Pride and Prejudice"); m1.borrowBook("Data Structures in Java"); m1.showBorrowedBooks(); m1.returnBook("Harry Potter");

m1.showBorrowedBooks();

}

}

1. **Output**

Rohitha borrowed: Harry Potter

Rohitha borrowed: Pride and Prejudice Rohitha borrowed: Data Structures in Java

Books borrowed by Rohitha:

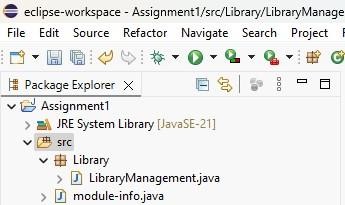
* Harry Potter
* Pride and Prejudice
* Data Structures in Java
* Rohitha returned: Harry Potter

Books borrowed by Rohitha:

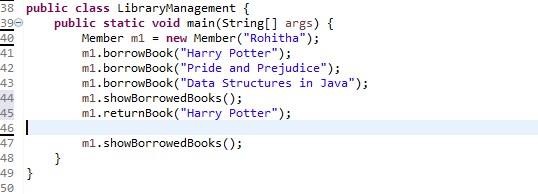
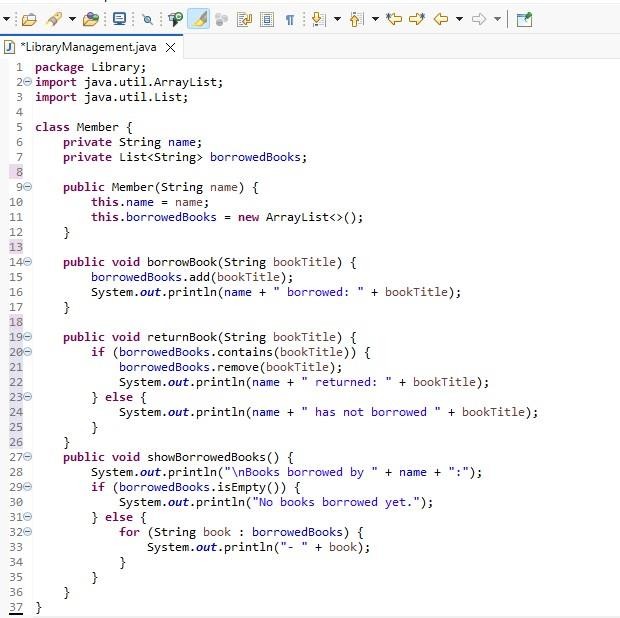
* Pride and Prejudice
* Data Structures in Java

# Screenshots

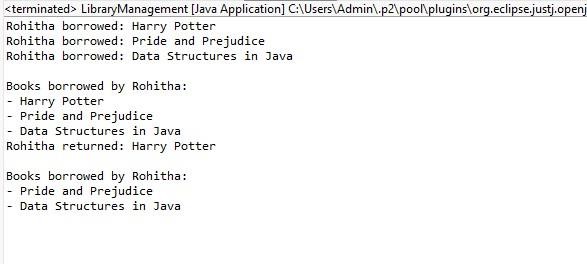
* **Project structure in Eclipse**

****

* **Code files**



* **Program output in console**

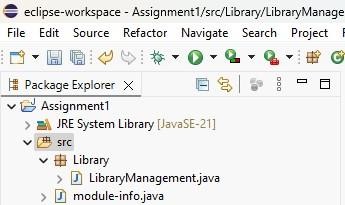
****

# GitHub Repository Link

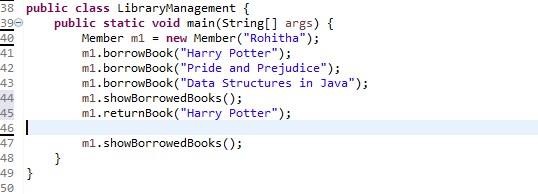
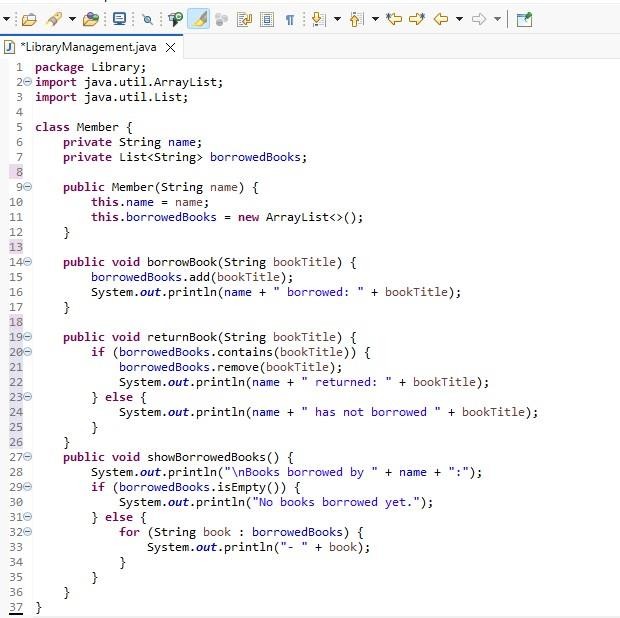
🔗 https://github.com/tha-rohi/assignment1

# Screenshots

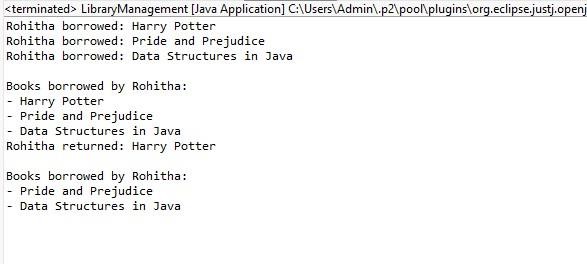
* **Project structure in Eclipse**

****

* **Code files**



* **Program output in console**

****

# GitHub Repository Link

🔗 https://githu