**CHRIST (Deemed to be University)**

**Department of Computer Science**

**MSc – Artificial Intelligence and Machine Learning**

**Name:** Joel Joseph Motha **Reg No.:** 2448521

**Course:** Java Programming **Component:** Lab Practical CIA 2

**Description:**

* This is the code for a library management system which does the basic functions of adding, removing, displaying, borrowing and returning a book.
* It contains 3 classes - book, library and main.
* The book classes is used to store the details of book id, title, availability etc. and the library class is used to perform the said functionalities.
* The main class is used to enter the book details, and it uses a menu driven approach wherein the user would find it more systematic to borrow and return a book, and the librarian would find it easier to add and remove a book.

**Program:**

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

class Book {

private int bookId;

private String title;

private String author;

private boolean isAvailable;

public Book(int bookId, String title, String author) {

if (bookId <= 0) throw new IllegalArgumentException("Book ID must be positive.");

this.bookId = bookId;

this.title = title;

this.author = author;

this.isAvailable = true;

}

public int getBookId() {

return bookId;

}

public String getTitle() {

return title;

}

public boolean isAvailable() {

return isAvailable;

}

public void borrow() {

isAvailable = false;

}

public void returnBook() {

isAvailable = true;

}

public void displayInfo() {

System.out.println("ID: " + bookId + ", Title: " + title + ", Author: " + author + ", Available: " + isAvailable);

}

}

class Library {

private List<Book> books = new ArrayList<>();

public void addBook(Book book) {

books.add(book);

System.out.println("Added: " + book.getTitle());

}

public void removeBook(int bookId) {

books.removeIf(book -> book.getBookId() == bookId);

System.out.println("Removed book with ID: " + bookId);

}

public Book findBook(int bookId) {

for (Book book : books) {

if (book.getBookId() == bookId) {

return book;

}

}

return null;

}

public void displayLibraryInfo() {

System.out.println("Library Books:");

for (Book book : books) {

book.displayInfo();

}

}

}

public class LibraryManagementSystem {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Library library = new Library();

String userInput;

while (true) {

System.out.println("\n--- Library Management System ---");

System.out.println("1. Add Book");

System.out.println("2. Remove Book");

System.out.println("3. Display Books");

System.out.println("4. Borrow Book");

System.out.println("5. Return Book");

System.out.println("6. Exit");

System.out.print("Choose an option: ");

userInput = scanner.nextLine();

switch (userInput) {

case "1":

System.out.print("Enter Book ID: ");

int addId = Integer.parseInt(scanner.nextLine());

System.out.print("Enter Title: ");

String title = scanner.nextLine();

System.out.print("Enter Author: ");

String author = scanner.nextLine();

library.addBook(new Book(addId, title, author));

break;

case "2":

System.out.print("Enter Book ID to remove: ");

int removeId = Integer.parseInt(scanner.nextLine());

library.removeBook(removeId);

break;

case "3":

library.displayLibraryInfo();

break;

case "4":

System.out.print("Enter Book ID to borrow: ");

int borrowId = Integer.parseInt(scanner.nextLine());

Book borrowBook = library.findBook(borrowId);

if (borrowBook != null && borrowBook.isAvailable()) {

borrowBook.borrow();

System.out.println("You borrowed: " + borrowBook.getTitle());

} else {

System.out.println("Book not available or does not exist.");

}

break;

case "5":

System.out.print("Enter Book ID to return: ");

int returnId = Integer.parseInt(scanner.nextLine());

Book returnBook = library.findBook(returnId);

if (returnBook != null) {

returnBook.returnBook();

System.out.println("You returned: " + returnBook.getTitle());

} else {

System.out.println("Book does not exist.");

}

break;

case "6":

System.out.println("Exiting the system.");

scanner.close();

return;

default:

System.out.println("Invalid option. Please try again.");

}

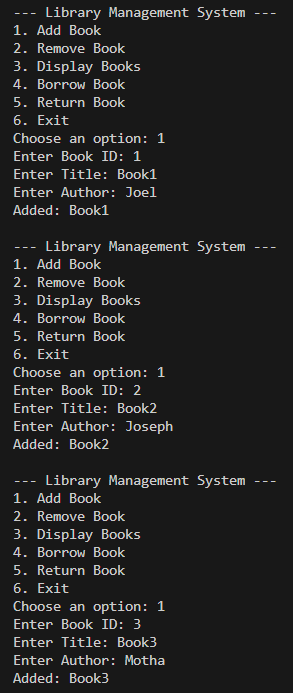
}

}

}

**Output:**

//Adding a book

****

//Displaying the books available for the user to borrow

A screen shot of a computer

Description automatically generated

//Removing a book

A computer screen shot of a program

Description automatically generated

//Borrowing a book

A screenshot of a computer program

Description automatically generated

//Returning the book

A computer screen shot of a program

Description automatically generated

//Exit

A screen shot of a computer

Description automatically generated