Library Management System

Overview:

- This system provides real-time information about books available in the library and user details.
- It reduces manual work and is designed for librarians and students. Each student has a unique ID used
- for book issues, returns, and fines. The system allows book and user management and keeps records of all transactions in the library.

Flowchart Overview:

1. Book Issue

-> Input Student ID & Book ID

-> Check Book Availability

-> Issue Book & Update Records

2. Book Return

-> Input Student ID & Book ID

-> Update Return Date

-> Calculate Fine (if any)

3. User Management

-> Add/View/Update Student Records

4. Book Management

-> Add/View/Search/Update Book Records

Core Java Concepts Used and Why:

1. OOP Principles (Encapsulation, Abstraction):

- Used to design modular and maintainable code.

- The Book, Student, and Library classes encapsulate data and related behaviors.

2. Classes and Objects:

- Classes define the blueprint (Book, Student, Library) and objects are instances used at runtime.

- Makes the system scalable and easy to manage.

3. Collections - ArrayList:

- Chosen for storing dynamic lists of books and students.
- Allows easy addition, iteration, and search operations.

4. Scanner Class:

- Used to accept user input from the console interactively.

5. Control Statements (if, switch):

- Used for decision making and menu navigation.
- Keeps the program logic clear and structured.

6. Exception Handling (try-catch - not shown but recommended):

- Should be added to handle invalid input or runtime issues gracefully.

Classes and Methods:

1. Book (Class)

- Fields: bookId, title, author, availability
- Methods: displayBookDetails()

2. Student (Class)

- Fields: studentId, name, issuedBooks
- Methods: displayStudentDetails()

3. Library (Class)

- Fields: List<Book>, List<Student>
- Methods: issueBook(), returnBook(), addBook(), addStudent(), viewRecords()

4. LibraryManagementSystem (Main Class)

- main(String[] args): Console interface to manage system operations

```
import java.util.*;

class Book {
    private String bookId;
    private String title;
    private String author;
    private boolean isAvailable;

    public Book(String bookId, String title, String author) {
        this.bookId = bookId;
        this.title = title;
        this.author = author;
        this.isAvailable = true;
    }

    public String getBookId() { return bookId; }
    public String getTitle() { return title; }
    public String getAuthor() { return author; }
    public boolean isAvailable() { return isAvailable; }
    public void setAvailable(boolean status) { this.isAvailable = status; }

    public void displayBookDetails() {
        System.out.println("Book ID: " + bookId);
        System.out.println("Title: " + title);
        System.out.println("Author: " + author);
        System.out.println("Available: " + isAvailable);
        System.out.println("-------------------------");
    }
```

```
}

class Student {
    private String studentId;
    private String name;
    private List<String> issuedBooks;

    public Student(String studentId, String name) {
        this.studentId = studentId;
        this.name = name;
        this.issuedBooks = new ArrayList<>();
    }

    public String getStudentId() { return studentId; }
    public String getName() { return name; }
    public List<String> getIssuedBooks() { return issuedBooks; }

    public void issueBook(String bookId) {
        issuedBooks.add(bookId);
    }

    public void returnBook(String bookId) {
        issuedBooks.remove(bookId);
    }

    public void displayStudentDetails() {
        System.out.println("Student ID: " + studentId);
        System.out.println("Name: " + name);
        System.out.println("Issued Books: " + issuedBooks);
        System.out.println("-------------------------");
    }
```

```
}

class Library {
    private List<Book> books = new ArrayList<>();
    private List<Student> students = new ArrayList<>();

    public void addBook(Book book) {
        books.add(book);
        System.out.println("Book added: " + book.getTitle());
    }

    public void addStudent(Student student) {
        students.add(student);
        System.out.println("Student added: " + student.getName());
    }

    public void issueBook(String studentId, String bookId) {
        Student student = findStudentById(studentId);
        Book book = findBookById(bookId);

        if (student != null && book != null && book.isAvailable()) {
            student.issueBook(bookId);
            book.setAvailable(false);
            System.out.println("Book issued successfully.");
        } else {
            System.out.println("Cannot issue book. Check availability and student ID.");
        }
    }

    public void returnBook(String studentId, String bookId) {
        Student student = findStudentById(studentId);
```

```
    Book book = findBookById(bookId);

    if (student != null && book != null && !book.isAvailable()) {
        student.returnBook(bookId);
        book.setAvailable(true);
        System.out.println("Book returned successfully.");
    } else {
        System.out.println("Cannot return book. Check records.");
    }
}

public void viewRecords() {
    System.out.println("--- Book Records ---");
    for (Book b : books) {
        b.displayBookDetails();
    }
    System.out.println("--- Student Records ---");
    for (Student s : students) {
        s.displayStudentDetails();
    }
}

private Book findBookById(String bookId) {
    for (Book b : books) {
        if (b.getBookId().equalsIgnoreCase(bookId)) {
            return b;
        }
    }
    return null;
}
```

```
    private Student findStudentById(String studentId) {
        for (Student s : students) {
            if (s.getStudentId().equalsIgnoreCase(studentId)) {
                return s;
            }
        }
        return null;
    }
}

public class LibraryManagementSystem {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        Library library = new Library();

        while (true) {
            System.out.println("\n--- Library Management System ---");
            System.out.println("1. Add Book");
            System.out.println("2. Add Student");
            System.out.println("3. Issue Book");
            System.out.println("4. Return Book");
            System.out.println("5. View Records");
            System.out.println("6. Exit");
            System.out.print("Enter your choice: ");
            int choice = sc.nextInt();
            sc.nextLine();

            switch (choice) {
                case 1:
                    System.out.print("Enter Book ID: ");
                    String bookId = sc.nextLine();
```

```
            System.out.print("Enter Title: ");
            String title = sc.nextLine();
            System.out.print("Enter Author: ");
            String author = sc.nextLine();
            library.addBook(new Book(bookId, title, author));
            break;
        case 2:
            System.out.print("Enter Student ID: ");
            String studentId = sc.nextLine();
            System.out.print("Enter Name: ");
            String name = sc.nextLine();
            library.addStudent(new Student(studentId, name));
            break;
        case 3:
            System.out.print("Enter Student ID: ");
            String sidIssue = sc.nextLine();
            System.out.print("Enter Book ID: ");
            String bidIssue = sc.nextLine();
            library.issueBook(sidIssue, bidIssue);
            break;
        case 4:
            System.out.print("Enter Student ID: ");
            String sidReturn = sc.nextLine();
            System.out.print("Enter Book ID: ");
            String bidReturn = sc.nextLine();
            library.returnBook(sidReturn, bidReturn);
            break;
        case 5:
            library.viewRecords();
            break;
        case 6:
```

```
                System.out.println("Exiting system. Goodbye!");
                sc.close();
                System.exit(0);
            default:
                System.out.println("Invalid choice. Please try again.");
        }
      }
   }
}
```