

Name - Mithunijaya chauhan (7) year 10

Röntgenbild (201420024) mitwürfelförmig

Colleges = B.Tech CSE + Data Science. + MCA +

? (honesti + : honest) 10

Java Assignment -4

Library City Digital Management System

```
import java.util.*; import java.util.
```

import java.util.*; io.*;

public class LibrarySystem{

static class Book implements Comparable<Book>

int BookId;

String title;

String author;

String category; non-enumerable

boolean isIssued;

3 (library_main) Book (int bookId, string title, string author, string category,

(boolean issued) { . . . }

miss-bookId = bookId;

this.title; this . title;

this author = author;

```
this.category = category;
```

for more to follow this. issued issued:

the following + y: the next + month

```

void display() {
    System.out.println("Book ID: " + bookId + " | "
        + title + " | " + author + " | " + category +
        " | Issued: " + issued);
}

```

3

@override

public int compareTo(Book b) {

return this.title.compareToIgnoreCase(b.title);

}

// Book class has been defined earlier

<static class Member> for book issue details

int memberId;

String name;

String email;

List<Integer> issuedBooks = new ArrayList<>();

// Member class has been defined earlier

Member(int memberId, String name, String email) {

this.memberId = memberId;

this.name = name;

this.email = email;

}

void display() {

System.out.print("Member ID: " + memberId + " | Name: " +

name + " | Email: " + email);

3 3

```

static Map<Integer, Book> books = new HashMap<>();
Map<Integer, Member> members = new HashMap<>();
((Optional) memberList)
    static file bookFile = new File("books.txt");
    static file memberFile = new File("members.txt");
    ((n, i) file -> i)
    ((Object) loadData, b) {
        ((o, l, b) static void loadFromFile() {
            try {
                if (bookFile.exists()) {
                    ((FileInputStream) bookFile).new BufferedReader()
                        .new FileReader(bookfile));
                    ((FileInputStream) bookfile).readLine();
                    while ((line = br.readLine()) != null) {
                        String d[] = line.split(",");
                        int id = Integer.parseInt(d[0]);
                        books.put(id, new Book(id, d[1], d[2], d[3],
                            Boolean.parseBoolean(d[4])));
                    }
                    br.close();
                }
            } catch (IOException e) {
                System.out.println("Error: " + e);
            }
        }
    }
}

```

```

if (mainBookfile.exists()) {
    BufferedReader br = new BufferedReader(
        new FileReader(mainBookfile));
    String line;
    while ((line = br.readLine()) != null) {
        String d[] = line.split(",");
        int id = Integer.parseInt(d[0]);
        Member member = Member(id, d[1], d[2]);
        memberList.add(member);
    }
}

for (int i = 3; i < d.length; i++) {
    Member member = new Member();
    member.setId(id);
    member.setName(d[i]);
    memberList.add(member);
}

if (id == 1) {
    System.out.println("Member added");
} else {
    System.out.println("Member updated");
}

try {
    BufferedWriter bw = new BufferedWriter(
        new FileWriter(bookfile));
    for (Book b : books.values()) {
        bw.write(b.bookId + "," + b.title + "," + b.author +
                "," + b.category + "," + b.issued);
        bw.newLine();
    }
}

```

```
continues with bw2.close(); won't fail if bw2 is null
```

```
((C) catch BufferedWriter bw2 = new BufferedWriter(new FileWriter("manzentele.txt"));
```

```
((C) try (bw2.write("Number of registered values: "));
```

```
bw2.write(nrOfUserIDs + ", " + m_name);
```

```
? ((36 sommose) m_email), bw2.newLine();
```

```
for ((int id : maxUsedBooks)) {
```

```
((C) try (bw2.write(id + ","));
```

```
bw2.newLine(); two?
```

```
((C) try (bw2.newLine());
```

```
((C) catch (Exception e) {
```

```
System.out.println("Exception found");
```

```
static void addBook(Scanner sc) {
```

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

```
? ((String id = sc.nextLine()); id != null;)
```

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

```
String title = sc.nextLine();
```

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

```
String author = sc.nextLine();
```

```
? ((String category = sc.nextLine()); !category.equals("quit")) {
```

```
System.out.println(category);
```

```
books.put(id, new Book(id, title, author,
                        edition, price, category, false));
((Book) books).saveToFile();
System.out.println("Book added");
+ " Name: " + name + " Email: " + email);
static void addMember(Scanner sc) {
    Sout("Member ID: ");
    int id = sc.nextInt(); sc.nextLine();
    Sout("Name: ");
    String name = sc.nextLine();
    System.out.print("Email: ");
    String email = sc.nextLine();
    ("book added") true;
members.put(id, new Member(id, name, email));
saveToFile();
Souts("Member added");
(" : at Node") true;
static void issueBook(Scanner sc) {
    Sout("Book ID: ");
    int bid = sc.nextInt();
    Sout("Member ID: ");
    int mid = sc.nextInt();
    if (!books.containsKey(bid) || !members.
        containsKey(mid)) {
        Sout("Invalid ID!");
    }
    return;
}
```

$\{ \text{if } (\text{Book} \cdot \text{id}) = \text{books} \cdot \text{get}(\text{bid}) \text{ in static}$

Member m = members · get(mid);

$\{ \text{if } (\text{Book} \cdot \text{id}) \text{ exists in two books}$

$\{ \text{System.out.println("Book issued");}$

System.out.println("Already issued!");

$\{ \text{return book;}\}$

$\{ \text{if } (\text{Book} \cdot \text{id}) \text{ exists;}\}$

$\{ \text{Member m = members · get(id);}\}$

$\{ \text{if } (\text{Book} \cdot \text{id}) \text{ exists;}\}$

m · issuedBooks · add(id);

saveToFile();

System.out.println("Book issued!");

3

static void returnBook(Scanner sc) {

System.out.println("Book ID: ");

int bid = sc.nextInt();

System.out.println("Member ID: ");

int mid = sc.nextInt();

book b = books · get(bid);

Member m = members · get(mid);

b · isIssued = false;

m · issuedBooks · remove(Integer.valueOf(bid));

saveToFile();

System.out.println("Book returned!");

3

```
static void searchBooks (Scanner sc) {  
    sc.nextLine();  
    System.out.println ("Book keyword : ");  
    String key = sc.nextLine ();toLowerCase (key);  
    for (Book b : books.values ()) {  
        if ((b.title.toLowerCase ()).contains (key))  
            b.author.toLowerCase ().contains (key))  
                b.category.toLowerCase ().contains (key)) {  
                    b.display ();  
    }  
}
```

```
static void sortBooks {  
    list < Books > list = new ArrayList < > (books.values());  
    Collections.sort (list);  
    for (Book b : list) {  
        b.display ();  
    }  
}
```

```
3  
;(b[0].title.equals ("J. K. Rowling"))  
;(b[1].title.equals ("J. R. R. Tolkien"))  
;(b[2].title.equals ("C. S. Lewis"))  
;(b[3].title.equals ("Stephen King"))  
;
```

"Character book" true

```
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    addFromFile();
    while (true) {
        System.out.println("1. City Library digital
                           system");
        System.out.println("1. Add Book");
        System.out.println("2. Add Member");
        System.out.println("3. Issue Book");
        System.out.println("4. Return Book");
        System.out.println("5. Search Book");
        System.out.println("6. Sort Books");
        System.out.println("7. Exit");
        System.out.println("Choice:");
        int ch = sc.nextInt();
        switch (ch) {
            case 1: addBook(sc); break;
            case 2: addMember(sc); break;
            case 3: issueBook(sc); break;
            case 4: returnBook(sc); break;
            case 5: searchBook(sc); break;
            case 6: sortBooks(sc); break;
            case 7: saveToFile();
                    System.out.println("Goodbye!");
                    return;
        }
    }
}
```