

1

Name - Kashish Sharma

Roll No - 2401420039

Course - BTech CSE - DS

# JAVA PROGRAMMING

## ASSIGNMENT - 4

Code :

```
import java.util.*;  
import java.io.*;  
public class LibrarySystem {  
  
    static class Book implements Comparable<Book> {  
        int bookId;  
        String title;  
        String author;  
        String category;  
        boolean isIssued;  
  
        Book(int bookId, String title, String author, String category,  
             boolean isIssued) {  
            this.bookId = bookId;  
            this.title = title;  
            this.author = author;  
            this.category = category;  
            this.isIssued = isIssued;  
        }  
    }  
}
```

3

2

PAGE NO.:  
DATE: / /

```
void display() {
    System.out.println("ID" + bookId + " | " + title + " | "
        + author + " | " + category + " | Issued" + isIssued);
```

}  
public int compareTo(Book b) {
 return this.title.compareToIgnoreCase(b.title);

}  
Static class Member {
 int memberId;
 String name;
 String email;
 List<Integer> issuedBooks = new ArrayList<>();

```
Member(int memberId, String name, String email) {
    this.memberId = memberId;
    this.name = name;
    this.email = email;
}
```

void display() {
 System.out.println("MemberId: " + memberId +
 "Name: " + name + "Email: " + email);

}  
Static Map<Integer, Book> books = new HashMap<>();
Static Map<Integer, Member> members = new HashMap<>();
Static File bookfile = new File("books.txt");
Static File memberfile = new File("members.txt");

```
3  
ON 30/11/2018  
  
Static void LoadFromFile() {  
    try {  
        if (bookfile.exists()) {  
            BufferedReader br = new BufferedReader  
(new FileReader(bookfile));  
            String line;  
            while ((line = br.readLine()) != null) {  
                String d[] = line.split(",");  
                int id = Integer.parseInt(d[0]);  
                book.put(id, new Book(id, d[1], d[2], d[3],  
                    Boolean.parseBoolean(d[4])));  
            }  
            br.close();  
        }  
        if (memberfile.exists()) {  
            BufferedReader br = new BufferedReader  
(new FileReader(memberfile));  
            String line;  
            while ((line = br.readLine()) != null) {  
                String d[] = line.split(",");  
                int id = Integer.parseInt(d[0]);  
                Member m = new Member(id, d[1], d[2]);  
                for (int i = 3; i < d.length; i++) {  
                    m.issuedBooks.add(Integer.parseInt(d[i]));  
                }  
                member.put(id, m);  
            }  
            br.close();  
        }  
    } catch (Exception e) {}  
}
```

```
1c void saveToFile() {
```

```
try {
```

```
    BufferedWriter bw = new BufferedWriter(new
```

```
        FileWriter(bookfile));
```

```
    for (Book b : books.values()) {
```

```
        bw.write(" " + b.bookId + ", " + b.title + ", " +
```

```
        b.author + ", " + b.category + ", " + b.isIssued);
```

```
        bw.newLine();
```

```
}
```

```
    bw.close();
```

```
    BufferedWriter bw2 = new BufferedWriter(
```

```
        new FileWriter(memberfile));
```

```
    for (Member m : members.value()) {
```

```
        bw2.write(m.memberId + ", " + m.name
```

```
        + ", " + m.email);
```

```
    for (int id : m.issuedBooks) bw2.write(", " + id);
```

```
    bw2.newLine();
```

```
}
```

```
    bw2.close();
```

```
} catch (Exception e) {}
```

```
}
```

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

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

```
    int id = sc.nextInt(); sc.nextLine();
```

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

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

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

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

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

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

```

books.put(id, newBook(id, title, author, category,
    false));
saveToFile();
System.out.print("Book Added!");
}

static void addMember(Scanner sc) {
    System.out.println("Member Id :");
    int id = sc.nextInt(); sc.nextLine();
    System.out.print("Name :");
    String name = sc.nextLine();
    System.out.print("Email :");
    String email = sc.nextLine();
    members.put(id, new Member(id, name,
        email));
    saveToFile();
    System.out.println("Member added!");
}

```

```

static void issueBook(Scanner sc) {
    System.out.print("Book Id");
    int bid = sc.nextInt();
    System.out.print("Member Id");
    int mid = sc.nextInt();
    if (!book.containsKey(bid) || !member.containsKey(
        mid)) {
        System.out.println("Invalid Ids!");
        return;
    }

```

```

    Book b = books.get(bid);
    Member m = members.get(mid);

```

```

if (b.issued) {
    System.out.println("Already issued!");
    return;
}

```

```

b.issued = true;
m.issuedBooks.add(bid);
saveToFile();
System.out.println("Book issue!");
}

```

```

static void returnBook(Scanner sc) {
    System.out.print("Book Id ");
    int bid = sc.nextInt();
    System.out.print("Member ID: ");
    int mid = sc.nextInt();
    b.issued = false;
    m.issuedBook.remove(Integer.valueOf(bid));
    saveToFile();
    System.out.println("Book returned!");
}

```

```

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

```

```

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

```

```

public static void main (String [ ] args) {
    Scanner sc = new Scanner (System.in);
    loadFromFile();
    while (true) {

```

~~System.out~~ ("City Library Digital System");

Sout ("1. Add Book");

Sout ("2. Add Member");

Sout ("3. Issue Book");

Sout ("4. Return Book");

Sout ("5. Search Books");

Sout ("6. Sort Book");

Sout ("7. Exit");

Sout ("Choice: ");

int ch = sc.nextInt();

switch (ch) {

case 1: addBook (sc); break;

case 2: addMembers (sc); break;

case 3: issueBook (sc); break;

case 4: returnBook (sc); break;

case 5: searchBooks (sc); break;

case 6: sortBooks (sc); break;

case 7:

saveToFile ()

System.out.println ("Goodbye");

return;

default;

Sout ("Invalid");