

## Assignment - 4

Page No. \_\_\_\_\_  
Date \_\_\_\_\_

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

class Book implements Comparable <Book> {

```
private int bookId;  
private String title;  
private String author;  
private boolean issued;
```

```
public Book (int bookId, String title, String author,  
String category, boolean issued) {  
this.bookId = bookId;  
this.title = title;  
this.author = author;  
this.category = category;  
this.issued = issued;
```

```
public int getBookId() { return bookId; }  
public String getTitle() { return title; }  
public String getAuthor() { return author; }  
public String getCategory() { return category; }
```

```
public boolean isIssued() { return issued; }  
public void markAsIssued () { issued = true; }  
public void markAsReturned () { issued = false; }
```

```
public void displayBookDetails () {
```

```
System.out.println ("Book ID: " + bookId);  
System.out.println ("Title: " + title);
```

System.out.println ("author" + author);  
 System.out.println ("category:" + category);  
 System.out.println ("Issued:" + issued?  
 "yes"; "No"));

y  
public int compareTo (Book other) {  
return

this.title.compareTo (other.title);

y

public String toFileString() {  
return bookId + ";" + title + ", " + author + ", "  
+ category + ", " + issued;

y

public static Book fromFileString (String s) {

String [] p = s.split (" ");

return new Book (

Integer.parseInt (p[0]),

p[1],

p[2],

p[3],

Boolean.parseBoolean (p[4]))

};

y

y

Class Member?

private int memberId;  
 private String name;  
 private String email;

private int <integer> issued Books = new Arraylist >();

```
public Member (int memberid, String name, String email) {
    this.memberid = memberid;
    this.name = name;
    this.email = email;
```

```
} public int get Memberid () { return memberid; }
```

```
public void add Issued Book (int bookid) {
    issued Books . add (Bookid);
```

```
} public void return Issued Book (int bookid) {
    issued Books . remove (Integer . ValueOf (Bookid));
```

```
public String to File String () {
    return memberid + "," + name + "," + email + ","
        + issued Books . to String();
```

```
} public String to Member From File String (String s) {
    String [] p = s . split (" , ");
```

```
Member m = new
```

```
Member (Integer . parseInt (p [0]), p [1], p [2]);
```

```
if (p . length == 4) & & p [3] . length () > 2) {
```

```
String book = p [3] . Substring (1, p [3] . length () - 1);
```

```
if (! book . Empty ()) {
```

for (String x : Books split (" "))) {

m.addIssuedBook (Integer.parseInt(x));

y  
y  
y

return n;

y  
y

public class Main {

private Map< Integer, Book > books = new HashMap();

private Map< Integer, Member > members =

new HashMap<>();

private Set< String > categories = new HashSet<>();

private final String Book\_file = "Books.txt";

private final String member\_file = "members.txt";

Scanner sc = new Scanner (System.in);

public Main () {

load Books();

load Members();

y

void load Books () {

try {  
 BufferedReader br = new BufferedReader(new FileReader("book-file.txt"));

}

String line;

while (line = br.readLine() != null) {

book b = Book.fromString(line);

books.put(b.getBookId(), b);

categories.add(b.getCategory());

}

} catch (Exception e) {}

y

Void load Members () {

try {  
 BufferedReader br = new BufferedReader(new FileReader("MEMBER-FILE.txt"));

String line;

while (line = br.readLine() != null) {

book b = Book.fromString(line);

books.put(b.getBookId(), b);

categories.add(b.getCategory());

}

} catch (Exception e) {}

3

Void save Books () {

```

try (BufferedWriter bw = new BufferedWriter writer
     new Buffered Writer (new File Writer (Book - file))
for (Book b : book Values ()) {
    bw.write (b.to file strip ());
    bw.newLine ();
}

```

3  
} catch (Exception e) {  
}

3

void save Members () {

```

try (BufferedWriter bw = new BufferedWriter writer
     for (Member m : members values ()) {
        bw.newLine ();
}

```

3

} catch (Exception e) {  
}

3

public void add Book () {

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

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

bw.newLine ();

String title = sc.nextLine ();

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

String author = sc.nextLine ();

book b = new Book (id, title, author, category, false);

books.put (id, b);

Categories.add (category);

SaveBooks();

System.out.print ("New Book");

3

```
public void addMember() {
    System.out.println("Enter Member ID : ");
    int id = sc.nextInt(); sc.nextLine();
    System.out.println("Enter Email : ");
    String email = sc.nextLine();
    System.out.println("Book added.");
}
```

```
public void issueBook() {
    System.out.println("Enter Book ID : ");
    int bookId = sc.nextInt();
    System.out.println("Enter Member ID : ");
    int memberId = sc.nextInt();
    Book b = books.get(bookId);
    Member m = members.get(memberId);
    if (b == null || m == null) {
        System.out.println("Invalid book");
        return;
    }
}
```

```
if (b.isIssued()) {
    System.out.println("Book already issued");
    return;
}
```

```
b.markAssumed();
m.addIssuedBook(bookId);
saveBook();
saveMember();
System.out.println("Book issued successfully");
```

public void search books () {

sc. next line();

System.out.println ("search by title / author / category : ");

String key = sc.next line(). tolower case();

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

if

(b.get title(). tolower case() . contains (key)) ||

(b.get Author(). tolower case() . contains (key)) ||

(b.get Category(). tolower case() . contains (key)) ||

b.display Book Details();

System.out.println ("---");

y

y

y

public void sort Books () {

List < Book > list = new

ArrayList <> (books . values());

System.out.println ("1. Sort by Title");

System.out.println ("2. Sort by Author");

int choice = sc.next Int();

if (choice == 1)

list.sort

for (Book b : list) {

b.display Book Details();

System.out.println ("---");

y

y

```
public void menu () {
```

```
    int choice;
```

```
    do {
```

```
        System.out.println ("1 == city library Digital  
        management system = . = ")
```

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

```
        System.out.println ("2: Add Member");
```

```
        System.out.println ("3 Issue Book");
```

```
        System.out.println ("4 Search Books");
```

```
        System.out.println ("5 Return Books");
```

```
        System.out.println ("6 Sort Books");
```

```
        System.out.println ("7 Exit");
```

```
    switch (choice) {
```

```
        case 1: addBook(); break;
```

```
        case 2: addMember(); break;
```

```
        case 3: issueBook(); break;
```

```
        case 4: returnBook(); break;
```

```
        case 5: searchBooks(); break;
```

```
        case 6: sortBooks(); break;
```

```
        case 7:
```

```
            saveBooks();
```

```
            saveMembers();
```

```
            System.out.println ("Goodbye!");
```

```
            break;
```

```
        default:
```

```
            System.out.println ("Invalid choice");
```

Y while (choice != 7);

Y

Public static void main (String [ ] args) {

Main.Dm = new Main();

Dm.menu();

Y

Y