

ASSIGNMENT - 4

Md. Anas Khan
2401420035

```
import java.io.*;
import java.util.*;
public class CityLibrary {
    public static void main(String [] args) {
        LibraryManager lm = new LibraryManager();
        lm.load();
        lm.menu();
        lm.save();
    }
}
```

```
class Book implements Comparable<Book> {
    int id; String title, author, category;
    boolean issued;
    Book(int id, String t, String a, String c, boolean i) {
        this.id = id; title = t; author = a; category = c; issued = i;
    }
    public int compareTo(Book o) {
        return title.compareToIgnoreCase(o.title);
    }
}
```

```
void print() {
    System.out.println(id + " " + title + " " + author + " " + category +
        " " + (issued ? "Issued" : "Available"));
}
```

```
class Member {
    int id; String name, email;
    List<Integer> books = new ArrayList<>();
```

```
Member(int id, String n, String e) {
    this.id = id; name = n; email = e;
}
```

```
}
```

```
void print() {
    System.out.println(id + " | " + name + " | " + email + " | " + books);
}
```

```
}
```

```
class LibraryManager {
    Map<Integer, Book> books = new HashMap<>();
    Map<Integer, Member> members = new HashMap<>();
    Set<String> categories = new HashSet<>();
    Scanner sc = new Scanner(System.in);
}
```

```
void menu() {
    while(true) {
        System.out.println("1. Add Book 2. Add Member
3. Issue 4. Return 5. Search 6. Sort 7. Exit");
        int ch = Integer.parseInt(sc.nextLine());
        if(ch == 1) addBook();
        else if(ch == 2) addMember();
        else if(ch == 3) issue();
        else if(ch == 4) ret();
        else if(ch == 5) search();
        else if(ch == 6) sort();
        else if(ch == 7) break;
    }
}
```

```
void addBook() {
    System.out.print("Title: ");
    String t = sc.nextLine();
    System.out.print("Author: ");
    String a = sc.nextLine();
    System.out.print("Category: ");
    String c = sc.nextLine();
    int id = books.size() + 101;
}
```

```

books.put(id, new Book(id, t, a, c, false));
categories.add(c);
System.out.println("Book ID: "+id);
save();
}

```

```
void addMember() {
```

```

System.out.print("Name: "); String n = sc.nextLine();
System.out.print("Email: "); String e = sc.nextLine();
int id = members.size() + 1;
members.put(id, new Member(id, n, e));
System.out.println("Member ID: " + id);
save();
}

```

```
}
```

```
void issue() {
```

```

System.out.print("Book ID: "); int b = Integer.parseInt(sc.nextLine());
System.out.print("Member ID: "); int m = Integer.parseInt(sc.nextLine());
if (!books.containsKey(b) || !members.containsKey(m)) {
    System.out.println("Invalid"); return;
}
if (books.get(b).issued) {
    System.out.println("Already issued"); return;
}
books.get(b).issued = true;
members.get(m).books.add(b);
System.out.println("Issued");
save();
}

```

```
}
```

```
void ret() {
```

```

System.out.print("Book ID: ");
int b = Integer.parseInt(sc.nextLine());
System.out.print("Member ID: ");
int m = Integer.parseInt(sc.nextLine());
}

```

```

if (!books.containsKey(b) || !members.containsKey(m)) {
    System.out.println("Invalid"); return;
}

books.get(b).issued = false;
members.get(m).books.remove(Integer.valueOf(b));
System.out.println("Returned");
Save();
}

```

3 void search() {

```

System.out.print("Search: ");
String k = sc.nextLine().toLowerCase();
for (Book b : books.values()) {
    if (b.title.toLowerCase().contains(k) ||
        b.author.toLowerCase().contains(k) ||
        b.category.toLowerCase().contains(k))
        b.print();
}

```

3 void sort() {

```

List<Book> list = new ArrayList<>(books.values());
System.out.println("1. Title 2. Author");
int c = Integer.parseInt(sc.nextLine());
if (c == 1) Collections.sort(list);
else list.sort((x, y) -> x.author.compareToIgnoreCase(y.author));
for (Book b : list) b.print();
}

```

3 void load() {

try {

```

BufferedReader br = new BufferedReader(new FileReader("books.txt"));
String ln;

```

Date _____

while ((ln = br.readLine()) != null) {
 String [] p = ln.split(":");
 Book b = new Book(Integer.parseInt(p[0]), p[1],
 p[2], p[3]; Boolean.parseBoolean(p[4]);
 books.put(b.id, b);
 categories.add(b.category);
}

br.close();
BufferedReader br2 = new BufferedReader(new FileReader
("members.txt"));
while ((ln = br2.readLine()) != null) {
 String [] p = ln.split(":");
 Member m = new Member(Integer.parseInt(
 p[0]), p[1], p[2]);
 if (p.length > 3 && !p[3].isEmpty())
 for (String s : p[3].split(";")) m.books.add(
 Integer.parseInt(s));
 members.put(m.id, m);
}

br2.close();

} catch (Exception e) {}

void save() {

try {

BufferedWriter bw = new BufferedWriter(new FileWriter
 ("books.txt"));
 for (Book b : books.values())
 bw.write(b.id + ", " + b.title + ", " + b.author + ", " + b.category +
 ", " + b.issued + "\n");
 bw.close();

```
    BufferedWriter bw2 = new BufferedWriter(new  
FileWriter("members.txt"));  
    for(Member m : members.values()) {  
        StringBuilder sb = new StringBuilder();  
        for(int x : m.books) sb.append(x).append(" ");  
        bw2.write(m.id + "," + m.name + "," + m.email + ", " + sb + "\n");  
    }  
    bw2.close();  
} catch(Exception e) {}
```