



Mo	Tu	We	Th	Fr	Sa	Su
----	----	----	----	----	----	----

Memo No. _____

Date: / /

```
import java.util.*;  
class Book implements Comparable<Book> {  
    int id;  
    String title, author;
```

```
    Book(int id, String t, String a) {  
        this.id = id;  
        title = t;  
        author = a;  
    }
```

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

```
}
```

```
public class LibraryManager {  
    Scanner sc = new Scanner(System.in);  
    ArrayList<Book> books = new ArrayList<>();  
    HashMap<Integer, Integer> issued = new HashMap<>();  
  
    void addBook() {
```



Mo	Tu	We	Th	Fr	Sa	Su
----	----	----	----	----	----	----

Memo No. _____

Date: / /

try {

```
System.out.print("Book ID:"); int id = sc.nextInt();  
sc.nextLine();
```

```
System.out.print("Title:"); String t = sc.nextLine();
```

```
System.out.println("Author:"); String a = sc.nextLine();
```

```
books.add(new Book(id, t, a));
```

```
System.out.println("Added! \n");
```

```
} catch (Exception e) { sc.nextLine(); System.out.println  
("Invalid!");
```

}

void searchBook() {

try {

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

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

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

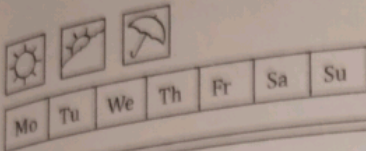
```
if (b.id == id) {
```

```
System.out.println("ID: " + b.id);
```

```
System.out.println("Title: " + b.title);
```

```
System.out.println("Author: " + b.author);
```

```
System.out.println("Status: " + (issues.contains(id)?  
"Issued": "Available"));
```

```
return;
```

```
}
```

```
}
```

```
System.out.println("Not Found");
```

```
} catch (Exception e) { sc.nextLine(); System.out.println("Invalid!");
```

```
}
```

```
}
```

```
void sortBooks() {
```

```
    Collections.sort(books);
```

```
    System.out.println("\nSorted Books:");
```

```
    for (Book b : books)
```

```
        System.out.println(b.id + " - " + b.title + " - " + b.author);
```

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

```
}
```

```
void issueBooks() {
```

```
    try {
```

```
        System.out.print("Book ID:"); int id = sc.nextInt();
```

```
        System.out.print("Member ID:"); int m = sc.nextInt();
```

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




Memo No. _____

Date: / /

Mo	Tu	We	Th	Fr	Sa	Su
----	----	----	----	----	----	----

```
if (bid == id) {
```

```
    if (issued.containsKey(id)) { System.out.println  
        ("Already Issued"); return; }
```

```
    issued.put(id, m);
```

```
    System.out.println("Issued! \n");
```

```
}
```

```
}
```

```
System.out.println("Not Found");
```

```
} catch (Exception e) { sc.nextLine(); System.out.println("Invalid!");
```

```
}
```

```
}
```

```
void returnBook() {
```

```
    try {
```

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

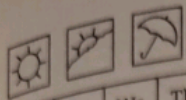
```
        int id = sc.nextInt();
```

```
        System.out.println(issued.remove(id) != null ? "Returned! \n":  
            "Not Issued");
```

```
    } catch (Exception e) { sc.nextLine(); System.out.println("Invalid!");
```

```
    }
```

```
}
```

Mo	Tu	We	Th	Fr	Sa	Su
----	----	----	----	----	----	----

Memo No. _____

Date: / /

```
void menu() {
```

```
    int ch = 0;
```

```
    while (ch != 6) {
```

```
        System.out.println("1. Add Book\n2. Search Book\n3.
```

```
        sort Books\n4. Issue\n5. Return\n6. Exit");
```

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

```
        try { ch = sc.nextInt(); }
```

```
        catch (Exception e) { sc.nextLine(); continue; }
```

```
        switch (ch) {
```

```
            case 1 -> addBook();
```

```
            case 2 -> searchBook();
```

```
            case 3 -> sortBook();
```

```
            case 4 -> issueBook();
```

```
            case 5 -> returnBook();
```

```
            case 6 -> System.out.println("Exiting...");
```

```
            default -> System.out.println("Invalid");
```

```
        }
```

```
    }
```

```
}
```

```
public static void main (String [] args) {
```



Mo	Tu	We	Th	Fr	Sa	Su
----	----	----	----	----	----	----

new Library Manager (U. menu());

y

y