LAB INDEX

NAME: Sahil Kaundal
UID: 21BCS8197
SECTION: 20BCS_WM-616/A

SUBJECT NAME: PBLJ (Lab)
SUBJECT CODE: 20CSP-321

Sr. No	Program	Date	Evaluation				Sign
			LW (12)	VV (10)	FW (8)	Total (30)	
1.	Create an application to save the employee information using arrays.	16/08/2022					
2.	Design and implement a simple inventory control system for a small video rental store.	20/08/2022					

CHANDIGARH UNIVERSITY

UNIVERSITY INSTITUTE OF ENGINEERING

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Submitted By: Sahil Kaundal	**Submitted To:** Neeru Sharma
Subject Name	Programming Based Learning Java (Lab)
Subject Code	20CSP-321
Branch	Computer Science Engineering
Semester	5th

Experiment 2

Student Name: Sahil Kaundal **UID:** 21BCS8197
Branch: BE CSE (Lateral Entry) **Section/Group:** 616/A
Semester: 5th **Date of Performance:** 20/08/2022
Subject Name: PBLJ Lab **Subject Code:** 20CSP-321

1. Aim/Overview of the practical:

Design and implement a simple inventory control system for a small video rental store.

2. Apparatus / Simulator Used:

1. Eclipse IDE - (Java)
2. NetBeans.

The goal of this project is to design and implement a simple inventory control system for a small video rental store.
Define least two classes: a class Video to model a video and a class VideoStore to model the actual store.

Assume that an object of class Video has the following attributes:
1. A title;
2. A flag to say whether it is checked out or not; and
3. An average user rating.

Add instance variables for each of these attributes to the Video class.

In addition, you will need to add methods corresponding to the following:
1. being checked out;
2. being returned; and
3. receiving a rating.

The VideoStore class will contain at least an instance variable that references an array of videos (say of length 10).

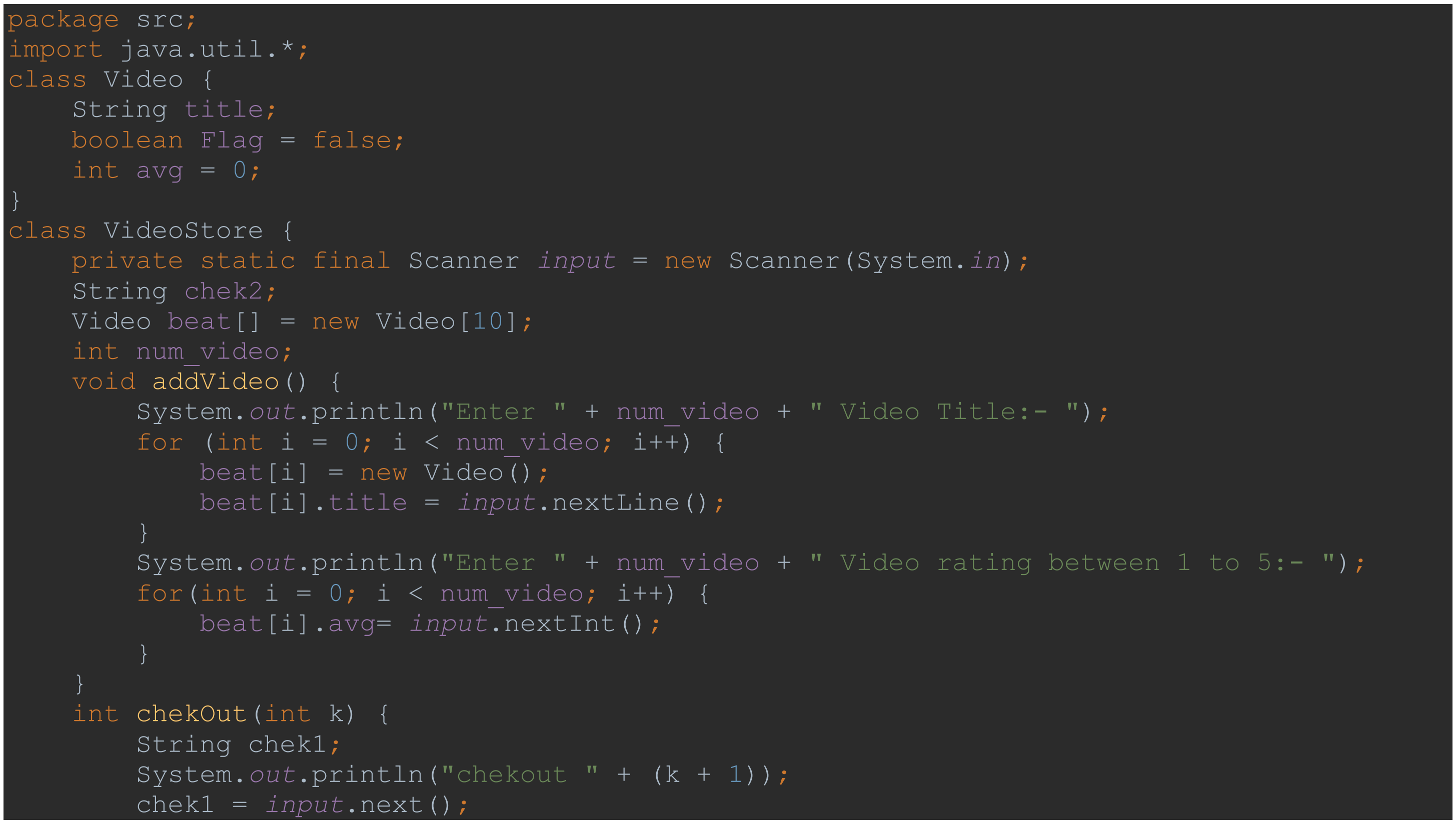
The VideoStore will contain the following methods:

1. addVideo(String): add a new video (by title) to the inventory;
2. checkOut(String): check out a video (by title);
3. returnVideo(String): return a video to the store;
4. receiveRating(String, int) : take a user's rating for a video;
5. listInventory(): list the whole inventory of videos in the store.

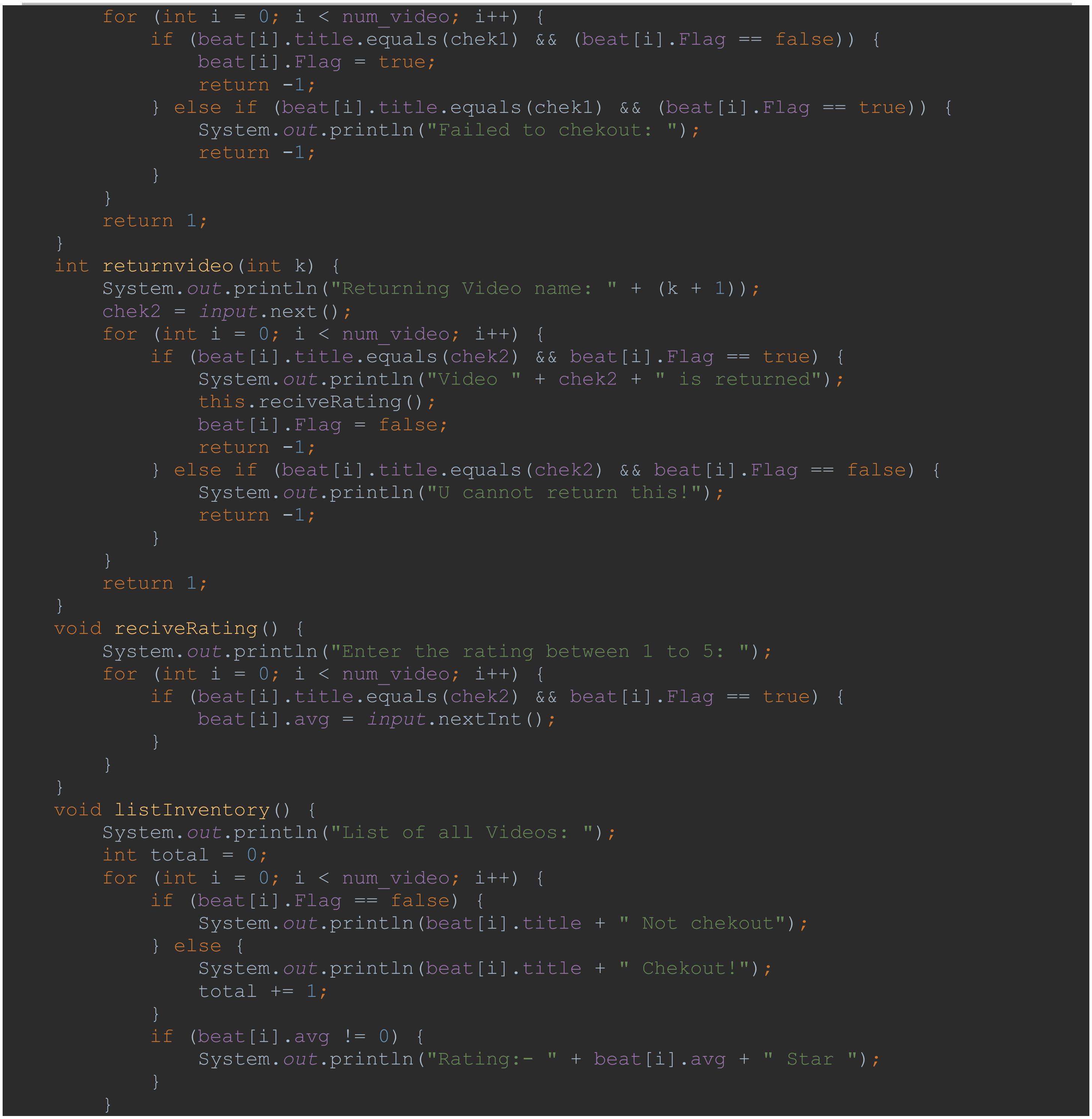
Finally, create a VideoStoreLauncher class with a main() method which will test the functionality of your other two classes.

It should allow the following.
1. Add 3 videos: "The Matrix", "Godfather II", "Star Wars Episode IV: A New Hope".
2. Give several ratings to each video.
3. Rent each video out once and return it.

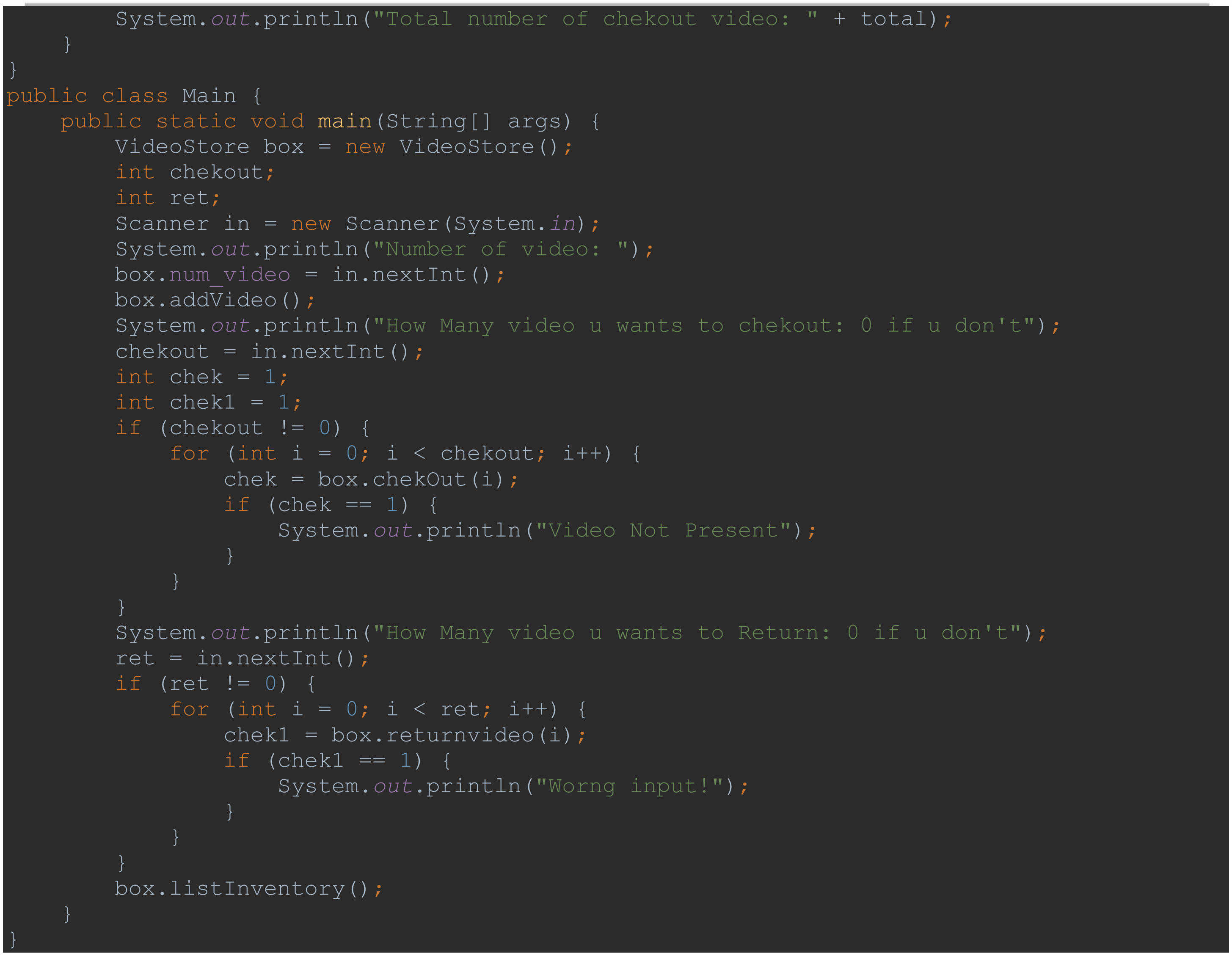
3. Programs:



```
package src;
import java.util.*;
class Video {
    String title;
    boolean Flag = false;
    int avg = 0;
}
class VideoStore {
    private static final Scanner input = new Scanner(System.in);
    String chek2;
    Video beat[] = new Video[10];
    int num_video;
    void addVideo() {
        System.out.println("Enter " + num_video + " Video Title:- ");
        for (int i = 0; i < num_video; i++) {
            beat[i] = new Video();
            beat[i].title = input.nextLine();
        }
        System.out.println("Enter " + num_video + " Video rating between 1 to 5:- ");
        for(int i = 0; i < num_video; i++) {
            beat[i].avg= input.nextInt();
        }
    }
    int chekOut(int k) {
        String chek1;
        System.out.println("chekout " + (k + 1));
        chek1 = input.next();
```

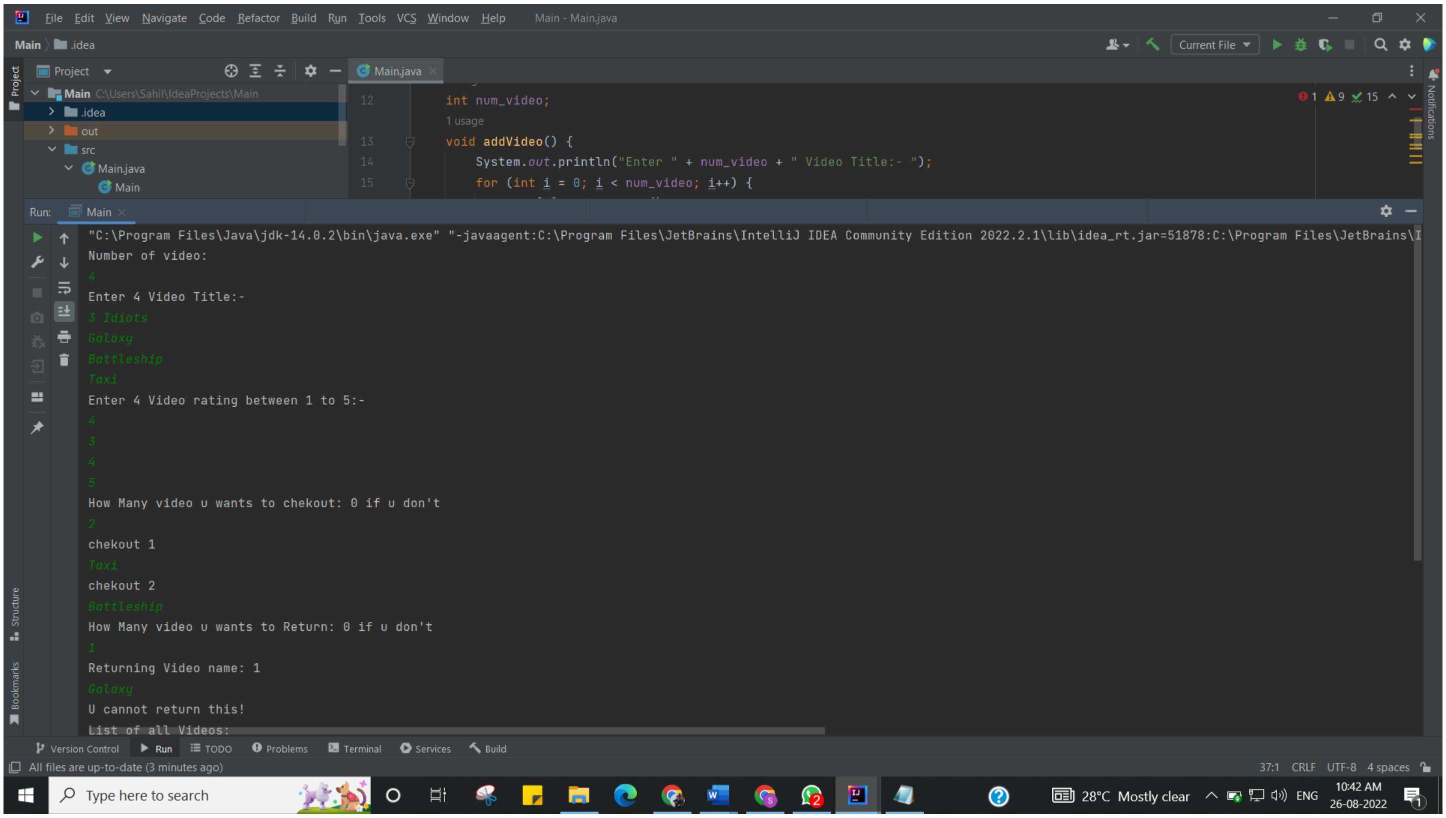


```
        for (int i = 0; i < num_video; i++) {
            if (beat[i].title.equals(chek1) && (beat[i].Flag == false)) {
                beat[i].Flag = true;
                return -1;
            } else if (beat[i].title.equals(chek1) && (beat[i].Flag == true)) {
                System.out.println("Failed to chekout: ");
                return -1;
            }
        }
        return 1;
    }
    int returnvideo(int k) {
        System.out.println("Returning Video name: " + (k + 1));
        chek2 = input.next();
        for (int i = 0; i < num_video; i++) {
            if (beat[i].title.equals(chek2) && beat[i].Flag == true) {
                System.out.println("Video " + chek2 + " is returned");
                this.reciveRating();
                beat[i].Flag = false;
                return -1;
            } else if (beat[i].title.equals(chek2) && beat[i].Flag == false) {
                System.out.println("U cannot return this!");
                return -1;
            }
        }
        return 1;
    }
    void reciveRating() {
        System.out.println("Enter the rating between 1 to 5: ");
        for (int i = 0; i < num_video; i++) {
            if (beat[i].title.equals(chek2) && beat[i].Flag == true) {
                beat[i].avg = input.nextInt();
            }
        }
    }
    void listInventory() {
        System.out.println("List of all Videos: ");
        int total = 0;
        for (int i = 0; i < num_video; i++) {
            if (beat[i].Flag == false) {
                System.out.println(beat[i].title + " Not chekout");
            } else {
                System.out.println(beat[i].title + " Chekout!");
                total += 1;
            }
            if (beat[i].avg != 0) {
                System.out.println("Rating:- " + beat[i].avg + " Star ");
            }
        }
```



```
        System.out.println("Total number of chekout video: " + total);
    }
}
public class Main {
    public static void main(String[] args) {
        VideoStore box = new VideoStore();
        int chekout;
        int ret;
        Scanner in = new Scanner(System.in);
        System.out.println("Number of video: ");
        box.num_video = in.nextInt();
        box.addVideo();
        System.out.println("How Many video u wants to chekout: 0 if u don't");
        chekout = in.nextInt();
        int chek = 1;
        int chek1 = 1;
        if (chekout != 0) {
            for (int i = 0; i < chekout; i++) {
                chek = box.chekOut(i);
                if (chek == 1) {
                    System.out.println("Video Not Present");
                }
            }
        }
        System.out.println("How Many video u wants to Return: 0 if u don't");
        ret = in.nextInt();
        if (ret != 0) {
            for (int i = 0; i < ret; i++) {
                chek1 = box.returnvideo(i);
                if (chek1 == 1) {
                    System.out.println("Worng input!");
                }
            }
        }
        box.listInventory();
    }
}
```

4. Result/Output/Writing Summary:

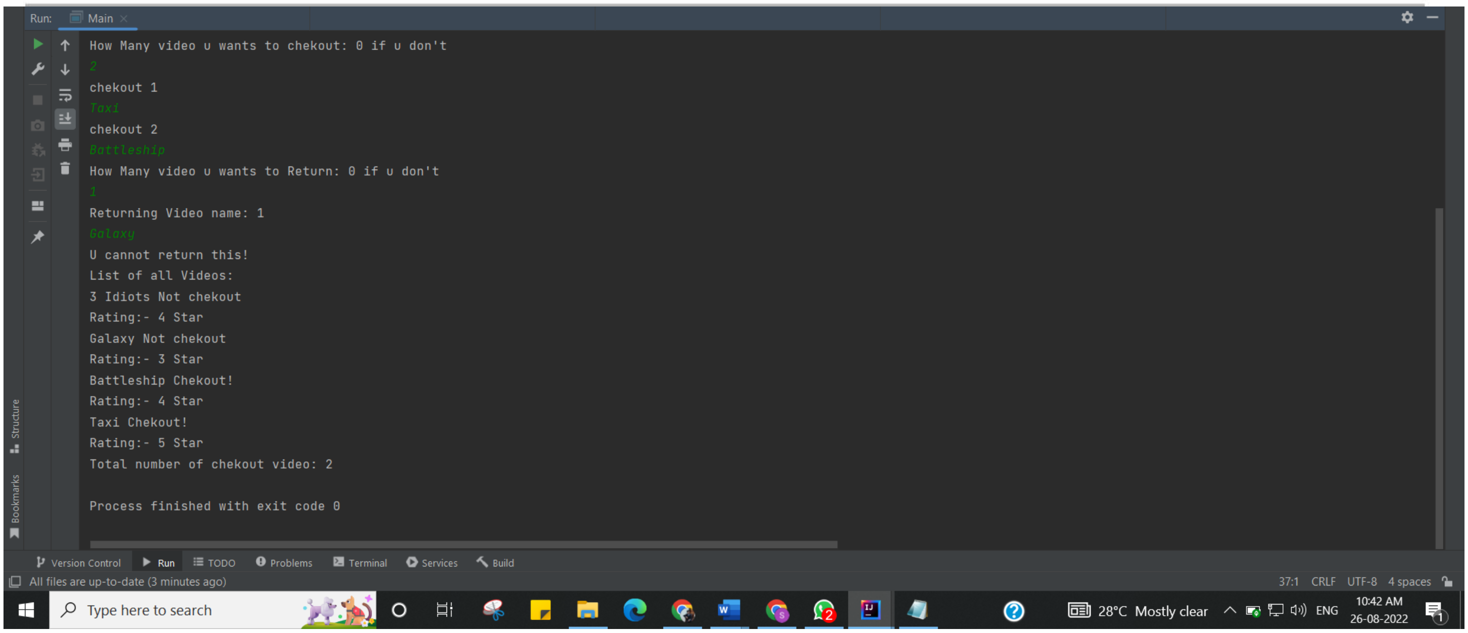


```
int num_video;

void addVideo() {
    System.out.println("Enter " + num_video + " Video Title:- ");
    for (int i = 0; i < num_video; i++) {
```

```
"C:\Program Files\Java\jdk-14.0.2\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.2.1\lib\idea_rt.jar=51878:C:\Program Files\JetBrains\I
Number of video:
4
Enter 4 Video Title:-
3 Idiots
Galaxy
Battleship
Taxi
Enter 4 Video rating between 1 to 5:-
4
3
4
5
How Many video u wants to chekout: 0 if u don't
2
chekout 1
Taxi
chekout 2
Battleship
How Many video u wants to Return: 0 if u don't
1
Returning Video name: 1
Galaxy
U cannot return this!
List of all Videos:
```

File Edit View Navigate Code Refactor Build Run Tools VCS Window Help Main - Main.java
Main .idea
Project
Main C:\Users\Sahil\IdeaProjects\Main
.idea
out
src
Main.java
Main
Main.java
1 usage
Current File
Run: Main
Version Control Run TODO Problems Terminal Services Build
All files are up-to-date (3 minutes ago)
37:1 CRLF UTF-8 4 spaces
Type here to search
28°C Mostly clear
ENG
10:42 AM
26-08-2022



```
How Many video u wants to chekout: 0 if u don't
2
chekout 1
Taxi
chekout 2
Battleship
How Many video u wants to Return: 0 if u don't
1
Returning Video name: 1
Galaxy
U cannot return this!
List of all Videos:
3 Idiots Not chekout
Rating:- 4 Star
Galaxy Not chekout
Rating:- 3 Star
Battleship Chekout!
Rating:- 4 Star
Taxi Chekout!
Rating:- 5 Star
Total number of chekout video: 2

Process finished with exit code 0
```

All files are up-to-date (3 minutes ago)

I have successfully done this program.

Learning Outcomes (What I have learnt):

1. Learn How use the inheritance concept.
2. java classes and all the features.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			