Experiment 02

Student Name: Deepak Kumar **UID: 22BCS10494**

Branch: BE-CSE **Section/Group: TPP-DL-902/B**

Semester: 6th **Date of Performance: 16/01/25**

Subject Name: Java Programming **Subject Code: 22CSH-359**

1. **Aim:** Design and implement a simple inventory control system for a small video rental store.

2. **Objectives of the Experiment:**

   - To learn about simple inventory control system and how to implement.

3. **Alogrithm:**

   a) **Initialize Inventory**:

   - Create an Inventory object.
   - Initialize an empty list to store Video objects.

   b) **Add Videos to Inventory**:

   - For each video to be added:
   - Create a Video object with title, genre, and year.
   - Add the Video object to the inventory list.

   c) **List Videos**:

   - Iterate through the videos list.
   - For each video, print its title, year, genre, and rental status (Rented or Available).

d) **Rent a Video**:

- Use findVideo(title) to search for the video by title.
- If the video is found and not rented, mark it as rented and print a confirmation message.
- If the video is already rented, print a message indicating it's unavailable.

e) **Return a Video**:

- Use findVideo(title) to search for the video by title.
- If the video is found and currently rented, mark it as available and print a confirmation message.
- If the video was not rented, print a message indicating so.

f) **Remove a Video**:

- Search for the video by title in the videos list.
- If the video is found, remove it from the list and print a confirmation message.
- If the video is not found, print a message indicating it's not in the inventory.

4. Implementation/Code:

```
import java.util.List;
import java.util.ArrayList;

class Video {
    private String title;
    private String genre;
    private int year;
    private boolean isRented;

    public Video(String title, String genre, int year) {
        this.title = title;
        this.genre = genre;
        this.year = year;
        this.isRented = false;
    }
```

```
    public void rentVideo() {
        if (!isRented) {
            isRented = true;
            System.out.println(title + " has been rented.");
        } else {
            System.out.println(title + " is already rented.");
        }
    }

    public void returnVideo() {
        if (isRented) {
            isRented = false;
            System.out.println(title + " has been returned.");
        } else {
            System.out.println(title + " was not rented.");
        }
    }

    public String getTitle() {
        return title;
    }

    public String getGenre() {
        return genre;
    }

    public int getYear() {
        return year;
    }

    public boolean isRented() {
        return isRented;
    }
}
class Inventory {
    private List<Video> videos;

    public Inventory() {
        videos = new ArrayList<>();
    }

    public void addVideo(Video video) {
```

```
        videos.add(video);
        System.out.println(video.getTitle() + " has been added to the inventory.");
    }

    public void removeVideo(String title) {
        for (Video video : videos) {
            if (video.getTitle().equals(title)) {
                videos.remove(video);
                System.out.println(title + " has been removed from the inventory.");
                return;
            }
        }
        System.out.println(title + " not found in the inventory.");
    }
    public void listVideos() {
        if (videos.isEmpty()) {
            System.out.println("No videos in the inventory.");
        } else {
            for (Video video : videos) {
                String status = video.isRented() ? "Rented" : "Available";
                System.out.println(video.getTitle() + " (" + video.getYear() + ", " +
video.getGenre() + ") - " + status);
            }
        }
    }
    public Video findVideo(String title) {
        for (Video video : videos) {
            if (video.getTitle().equals(title)) {
                return video;
            }
        }
        return null;
    }
    public static void main(String[] args) {
        Inventory inventory = new Inventory();

        inventory.addVideo(new Video("Tere Naam", "Emotional", 2009));
        inventory.addVideo(new Video("Inception", "Sci-Fi", 2010));
        inventory.addVideo(new Video("Devara", "Action", 2024));
        inventory.listVideos();
        Video video = inventory.findVideo("Inception");
        if (video != null) {
```

```
            video.rentVideo();
        }
        inventory.listVideos();
        if (video != null) {
            video.returnVideo();
        }
        inventory.listVideos();
        inventory.removeVideo("The Matrix");
        inventory.listVideos();
    }
}
```

5. Output:

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\dipak\OneDrive\Desktop\Java Programming> java Inventory
>>
Tere Naam has been added to the inventory.
Inception has been added to the inventory.
Devara has been added to the inventory.
Tere Naam (2009, Emotional) - Available
Inception (2010, Sci-Fi) - Available
Devara (2024, Action) - Available
Inception has been rented.
Tere Naam (2009, Emotional) - Available
Inception (2010, Sci-Fi) - Rented
Devara (2024, Action) - Available
Inception has been returned.
Tere Naam (2009, Emotional) - Available
Inception (2010, Sci-Fi) - Available
Devara (2024, Action) - Available
The Matrix not found in the inventory.
Tere Naam (2009, Emotional) - Available
Inception (2010, Sci-Fi) - Available
Devara (2024, Action) - Available
PS C:\Users\dipak\OneDrive\Desktop\Java Programming>
```

6. Learning Outcomes:

a) Understanding how to define and use classes (Video and Inventory).

b) Learning how to use methods with different parameters (method overloading).

c) Using the List interface and ArrayList class to manage a collection of Video objects.

d) Applying logical reasoning to solve problems like managing inventory and ensuring the correct state of video rentals.

e) Recognizing and fixing common compilation and runtime errors.