Experiment: 2

Student Name: AKASH DEEP **UID:22BCS10195**
Branch: CSE **Section/Group: DL-902**
Semester: 6th **Date of Performance:18/01/25**
Subject Name: Project Based Learning In Java with Lab **Subject Code: 22CSH-359**

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

2. Objective: To design and implement a Video Rental Inventory System that allows a video store to manage its inventory by performing the following operations:

1. Add videos to the inventory.
2. Rent out (check out) videos to customers.
3. Return videos back to the store.
4. Accept user ratings for videos and calculate their average ratings.
5. Display the current inventory with details such as title, rental status, and average ratings.

3. Steps to Solve:

1. Understand Requirements: Identify operations: Add videos, rent, return, rate, and display inventory.

2. Design Classes:

- Video: Handles individual video details and operations.
- VideoStore: Manages the inventory and video-related actions.
- VideoStoreLauncher: Tests the system.

3. Implement Functionality: Add methods for adding videos, renting, returning, rating, and displaying inventory.

4. Test the System: Add sample videos, perform operations, and validate the output.

5. Handle Edge Cases: Ensure proper error handling for invalid inputs or full inventory.

4. Code:

```
class Video {
    private String title;
    private boolean checkedOut = false;
    private double avgRating = 0.0;
    private int ratingCount = 0;

    Video(String title) {
        this.title = title;
    }
    String getTitle() {
        return title;
    }
    boolean isCheckedOut() {
        return checkedOut;
    }
    double getAvgRating() {
        return avgRating;
    }
    void checkOut() {
        if (!checkedOut) {
            checkedOut = true;
            System.out.println(title + " checked out.");
        } else {
            System.out.println(title + " already checked out.");
```

```
        }
    }
    void returnVideo() {
        if (checkedOut) {
            checkedOut = false;
            System.out.println(title + " returned.");
        } else {
            System.out.println(title + " was not checked out.");
        }
    }
    void rate(int rating) {
        if (rating >= 1 && rating <= 5) {
            avgRating = (avgRating * ratingCount + rating) / (++ratingCount);
            System.out.println("Rating added for " + title + ".");
        } else {
            System.out.println("Invalid rating. Must be between 1 and 5.");
        }
    }
}

class VideoStore {
    private Video[] videos = new Video[10];
    private int count = 0;
    void addVideo(String title) {
        if (count < videos.length) {
```

```
            videos[count++] = new Video(title);
            System.out.println(title + " added.");
        } else {
            System.out.println("Store is full. Can't add more videos.");
        }
    }
    void checkOut(String title) {
        Video video = findVideo(title);
        if (video != null) {
            video.checkOut();
        } else {
            System.out.println(title + " not found.");
        }
    }
    void returnVideo(String title) {
        Video video = findVideo(title);
        if (video != null) {
            video.returnVideo();
        } else {
            System.out.println(title + " not found.");
        }
    }
    void rateVideo(String title, int rating) {
        Video video = findVideo(title);
        if (video != null) {
```

```
            video.rate(rating);
        } else {
            System.out.println(title + " not found.");
        }
    }
    void showInventory() {
        System.out.println("Store Inventory:");
        for (int i = 0; i < count; i++) {
            System.out.printf("%s | Checked out: %b | Avg Rating: %.2f%n",
                    videos[i].getTitle(), videos[i].isCheckedOut(), videos[i].getAvgRating());
        }
    }
    private Video findVideo(String title) {
        for (int i = 0; i < count; i++) {
            if (videos[i].getTitle().equalsIgnoreCase(title)) {
                return videos[i];
            }
        }
        return null;
    }
}

public class VideoStoreLauncher {
    public static void main(String[] args) {
        VideoStore store = new VideoStore();
```

```
        store.addVideo("The Matrix");
        store.addVideo("Godfather II");
        store.addVideo("Star Wars Episode IV");
        store.rateVideo("The Matrix", 5);
        store.rateVideo("Godfather II", 4);
        store.rateVideo("Star Wars Episode IV", 3);
        store.checkOut("Godfather II");
        store.returnVideo("Godfather II");
        store.showInventory();
    }
}
```

5. Output:

```
The Matrix added.
Godfather II added.
Star Wars Episode IV added.
Rating added for The Matrix.
Rating added for Godfather II.
Rating added for Star Wars Episode IV.
Godfather II checked out.
Godfather II returned.
Store Inventory:
The Matrix | Checked out: false | Avg Rating: 5.00
Godfather II | Checked out: false | Avg Rating: 4.00
Star Wars Episode IV | Checked out: false | Avg Rating: 3.00
```

6. Learning Outcome :

1. Designed a functional system to manage video rentals, demonstrating the use of classes and objects in Java.

2. Implemented methods for operations like adding videos, renting out, returning, and recording user ratings.

3. Applied arrays to store and efficiently manage the video inventory within the store.

4. Learned to integrate multiple classes and enable seamless interaction among them in a structured program.

5. Strengthened understanding of object-oriented programming concepts like encapsulation and method abstraction.