Experiment 2

Student Name: Shikha Jumta **UID:** 22BCS12480
Branch: BE-CSE **Section/Group:** IoT_631(B)
Semester: 6th **Date of Performance:** 10/1/25
Subject Name: Project based learning in Java **Subject Code:** 22CSH-359

1. **Aim:** The goal of this project is to design and implement a simple inventory control system for a small video rental store.

2. **Objective:** Define at least two classes: a class Video to model a video and a class Video Store to model the actual store.

3. **Implementation:**

```
class Video {
    private String title;
    private boolean checkedOut;
    private double averageRating;
    private int ratingCount;
    private int ratingSum;

    // Constructor
    public Video(String title) {
        this.title = title;
        this.checkedOut = false;
        this.averageRating = 0.0;
        this.ratingCount = 0;
        this.ratingSum = 0;
    }

    // Check out the video
    public void checkOut() {
        if (!checkedOut) {
            checkedOut = true;
            System.out.println("Video '" + title + "' has been checked out.");
        } else {
            System.out.println("Video '" + title + "' is already checked out.");
        }
    }
```

```
    // Return the video
    public void returnVideo() {
        if (checkedOut) {
            checkedOut = false;
            System.out.println("Video '" + title + "' has been returned.");
        } else {
            System.out.println("Video '" + title + "' was not checked out.");
        }
    }

    // Receive a rating for the video
    public void receiveRating(int rating) {
        if (rating >= 1 && rating <= 5) {
            ratingCount++;
            ratingSum += rating;
            averageRating = (double) ratingSum / ratingCount;
            System.out.println("Rating of " + rating + " received for video '" + title + "'.");
        } else {
            System.out.println("Invalid rating. Please provide a rating between 1 and 5.");
        }
    }

    // Get the title
    public String getTitle() {
        return title;
    }

    // ToString method for inventory display
    @Override
    public String toString() {
        return "Title: " + title + ", Checked Out: " + checkedOut + ", Average Rating: " +
      String.format("%.2f", averageRating);
    }
}

class VideoStore {
    private Video[] inventory;
    private int count;
```

```
    // Constructor with preloaded videos
    public VideoStore() {
        inventory = new Video[]{
            new Video("The Matrix"),
            new Video("Godfather II"),
            new Video("Star Wars Episode IV: A New Hope")
        };
        count = inventory.length;
    }

    // Check out a video by title
    public void checkOut(String title) {
        Video video = findVideo(title);
        if (video != null) {
            video.checkOut();
        } else {
            System.out.println("Video '" + title + "' not found in inventory.");
        }
    }

    // Return a video by title
    public void returnVideo(String title) {
        Video video = findVideo(title);
        if (video != null) {
            video.returnVideo();
        } else {
            System.out.println("Video '" + title + "' not found in inventory.");
        }
    }

    // Receive a rating for a video
    public void receiveRating(String title, int rating) {
        Video video = findVideo(title);
        if (video != null) {
            video.receiveRating(rating);
        } else {
            System.out.println("Video '" + title + "' not found in inventory.");
        }
    }
```

```
    // List the inventory
    public void listInventory() {
        System.out.println("\nVideo Inventory:");
        for (Video video : inventory) {
            System.out.println(video);
        }
    }

    // Find a video by title
    private Video findVideo(String title) {
        for (Video video : inventory) {
            if (video.getTitle().equalsIgnoreCase(title)) {
                return video;
            }
        }
        return null;
    }
}

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

        // Give ratings to videos
        store.receiveRating("The Matrix", 5);
        store.receiveRating("Godfather II", 4);
        store.receiveRating("Star Wars Episode IV: A New Hope", 3);

        // Rent and return videos
        store.checkOut("The Matrix");
        store.returnVideo("The Matrix");
        store.checkOut("Godfather II");

        // List the inventory
        store.listInventory();
    }
}
```

4. OUTPUT:

```
Rating of 5 received for video 'The Matrix'.
Rating of 4 received for video 'Godfather II'.
Rating of 3 received for video 'Star Wars Episode IV: A New Hope'.
Video 'The Matrix' has been checked out.
Video 'The Matrix' has been returned.
Video 'Godfather II' has been checked out.

Video Inventory:
Title: The Matrix, Checked Out: false, Average Rating: 5.00
Title: Godfather II, Checked Out: true, Average Rating: 4.00
Title: Star Wars Episode IV: A New Hope, Checked Out: false, Average Rating: 3.00


...Program finished with exit code 0
Press ENTER to exit console.
```

5. Learning Outcomes

Understand object-oriented concepts like classes, objects, and encapsulation.
Learn to manage data using arrays and iterate through them.
Implement real-world logic like renting and returning videos.
Practice abstraction and relationships between classes.