Experiment 2

Student Name: Deepanshu **UID: 22BCS15133**
Branch: BE-CSE **Section/Group:DL-902/B**
Semester: Sixth **Date of Performance: 15/01/2025**
Subject Name: Project Based Learning in Java with Lab
Subject Code: 22CSH-359

1. **Aim:** To design and implement a simple inventory management system for a video rental store, enabling functionalities like adding videos, checking out videos, returning videos, receiving user ratings, and listing the inventory. The system will use object-oriented principles to model videos and the store effectively.

2. **Objective:**
   - To model a video rental store's operations by designing a Video class to represent individual videos and a VideoStore class to manage the store's inventory.
   - To enable video management, including adding new videos, checking out videos, and returning videos.
   - To collect user feedback by allowing users to rate videos and calculate an average rating for each video.
   - To maintain and display inventory details using a method that lists all videos, their availability status, and their ratings.
   - To validate the functionality of the system by testing the operations with a sample dataset in the VideoStoreLauncher class.

3. **Code:**

```
// Class to model a video
class Video {
```

```
private String title;
private boolean isCheckedOut;
private double averageRating;
private int ratingCount;

public Video(String title) {
    this.title = title;
    this.isCheckedOut = false;
    this.averageRating = 0.0;
    this.ratingCount = 0;
}

public String getTitle() {
    return title;
}

public boolean isCheckedOut() {
    return isCheckedOut;
}

public void checkOut() {
    if (!isCheckedOut) {
        isCheckedOut = true;
    }
}
```

```
    public void returnVideo() {
        if (isCheckedOut) {
            isCheckedOut = false;
        }
    }

    public void receiveRating(int rating) {
        if (rating >= 1 && rating <= 5) {
            averageRating = ((averageRating * ratingCount) + rating) / (ratingCount + 1);
            ratingCount++;
        }
    }

    @Override
    public String toString() {
        return "Title: " + title +
               ", Checked Out: " + isCheckedOut +
               ", Average Rating: " + String.format("%.2f", averageRating);
    }
}

// Class to model a video store
class VideoStore {
    private Video[] inventory;
```

```
private int count;

public VideoStore(int size) {
    inventory = new Video[size];
    count = 0;
}

public void addVideo(String title) {
    if (count < inventory.length) {
        inventory[count] = new Video(title);
        count++;
    }
}

public void checkOut(String title) {
    Video video = findVideo(title);
    if (video != null && !video.isCheckedOut()) {
        video.checkOut();
    }
}

public void returnVideo(String title) {
    Video video = findVideo(title);
    if (video != null && video.isCheckedOut()) {
        video.returnVideo();
```

```
        }
    }

    public void receiveRating(String title, int rating) {
        Video video = findVideo(title);
        if (video != null) {
            video.receiveRating(rating);
        }
    }

    public void listInventory() {
        for (int i = 0; i < count; i++) {
            System.out.println(inventory[i]);
        }
    }

    private Video findVideo(String title) {
        for (int i = 0; i < count; i++) {
            if (inventory[i].getTitle().equalsIgnoreCase(title)) {
                return inventory[i];
            }
        }
        return null;
    }
}
```

```
// Main class to test the functionality
public class VideoStoreLauncher {
    public static void main(String[] args) {
        VideoStore store = new VideoStore(10);

        // Adding videos
        store.addVideo("The Matrix");
        store.addVideo("Godfather II");
        store.addVideo("Star Wars Episode IV: A New Hope");

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

        // Checking out and returning videos
        store.checkOut("Godfather II");
        store.returnVideo("Godfather II");

        // Listing inventory
        System.out.println("Inventory:");
        store.listInventory();
    }
}
```

4. Output

```
input
The Matrix added to inventory.
Godfather II added to inventory.
Star Wars Episode IV: A New Hope added to inventory.
Video Inventory:
Title: The Matrix, Checked Out: false, Rating: 4.5
Title: Godfather II, Checked Out: true, Rating: 5.0
Title: Star Wars Episode IV: A New Hope, Checked Out: false, Rating: 3.5


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

5. Learning Outcomes:

- **Understanding Object-Oriented Concepts**: Learn to model real-world entities using classes, objects, encapsulation, and methods.
- **Implementing Inventory Management:** Develop a system to manage videos, including adding, renting, returning, and rating operations.
- **Working with Data Structures:** Gain experience in managing collections of data using arrays and performing operations on them.
- **Validations and Error Handling:** Practice handling edge cases and ensuring system reliability through proper checks and validations.
- **Testing and Debugging Skills:** Enhance problem-solving by testing functionality and debugging issues in a complete, interactive system**.**