Experiment 2

Student Name: Jayanth
UID: 22BCS11651
Branch: BE-CSE
Section/Group: 902/B
Semester: 6th
Date of Performance: 16/01/2025
Subject Name: Project Based Learning in Java with Lab
Subject Code: 22CSH-359

1. **Aim:** A Video Rental Inventory System.

2. **Objective:**
   The objective of this experiment is to design and implement a Video Rental Inventory System using Object-Oriented Programming (OOP) in Java. The system will manage a collection of videos, allowing users to:
   - Add new videos to the inventory.
   - Rent (check out) videos when a customer borrows them.
   - Return videos after use.
   - Assign ratings to videos based on user feedback.
   - Display the complete inventory with details like title, availability status, and average rating.

3. **Implementation/Code:**

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

    public Video(String title) {
```

```
    this.title = title;
    this.checkedOut = false;
    this.rating = 0.0;
    this.ratingCount = 0;
}

public void checkOut() {
    if (!checkedOut) {
        checkedOut = true;
        System.out.println(title + " has been checked out.");
    } else {
        System.out.println(title + " is already checked out.");
    }
}

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

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

```
            System.out.println("Rating received for " + title + ": " + rating);
        } else {
            System.out.println("Invalid rating. Please enter a rating between 1 and 5.");
        }
    }

    public String getTitle() {
        return title;
    }

    public boolean isCheckedOut() {
        return checkedOut;
    }

    public double getRating() {
        return rating;
    }
}

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

    public VideoStore() {
        inventory = new Video[10];
        count = 0;
    }
```

```
public void addVideo(String title) {
    if (count < inventory.length) {
        inventory[count++] = new Video(title);
        System.out.println(title + " added to inventory.");
    } else {
        System.out.println("Inventory full! Cannot add more videos.");
    }
}

public void checkOut(String title) {
    for (int i = 0; i < count; i++) {
        if (inventory[i].getTitle().equals(title)) {
            inventory[i].checkOut();
            return;
        }
    }
    System.out.println("Video not found.");
}

public void returnVideo(String title) {
    for (int i = 0; i < count; i++) {
        if (inventory[i].getTitle().equals(title)) {
            inventory[i].returnVideo();
            return;
        }
    }
```

```
        System.out.println("Video not found.");
    }

    public void receiveRating(String title, int rating) {
        for (int i = 0; i < count; i++) {
            if (inventory[i].getTitle().equals(title)) {
                inventory[i].receiveRating(rating);
                return;
            }
        }
        System.out.println("Video not found.");
    }

    public void listInventory() {
        System.out.println("\nCurrent Inventory:");
        for (int i = 0; i < count; i++) {
            System.out.println("Title: " + inventory[i].getTitle() + ", Checked Out: " +
inventory[i].isCheckedOut() + ", Rating: " + inventory[i].getRating());
        }
    }
}

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

        store.addVideo("The Matrix");
        store.addVideo("Godfather II");
```

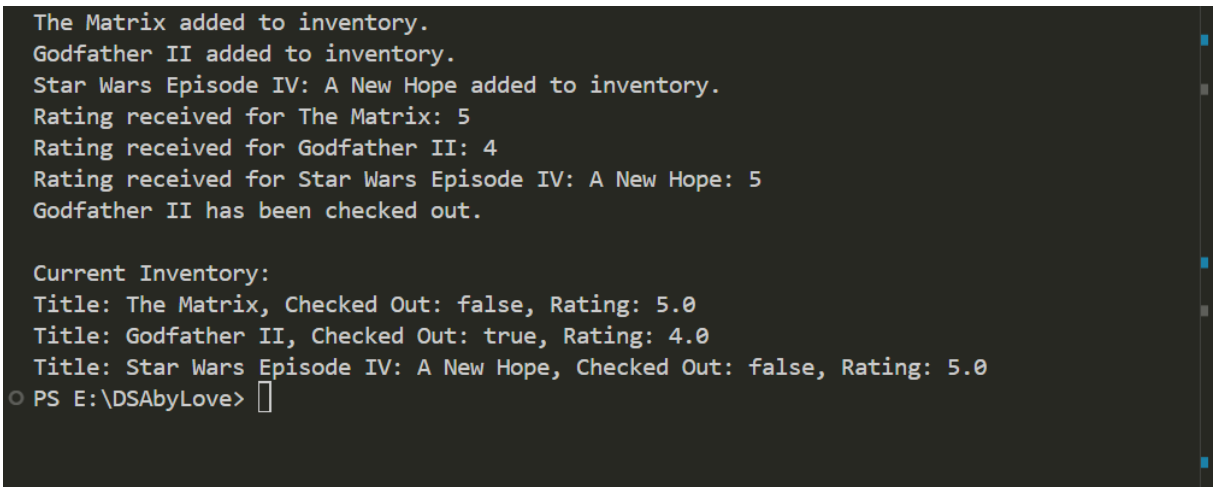
```
        store.addVideo("Star Wars Episode IV: A New Hope");

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

        store.checkOut("Godfather II");
        //store.returnVideo("Godfather II");

        store.listInventory();
    }
}
```

4. Output:



5. Learning Outcome:

- Understand object-oriented programming concepts like classes, objects, encapsulation, and methods.
- Learn to design and implement classes with attributes and methods to model real-world entities.
- Gain experience in managing arrays of objects, including adding, searching, and updating elements.
- Develop skills in using control structures like loops and conditionals for object manipulation.
- Practice testing and debugging by creating a launcher class to verify system functionality.