Experiment 2

Student Name: Mohammad Farhan Alam **UID: 22BCS13460**
Branch: BE-CSE **Section/Group: DL-902-A**
Semester:6th **Date of Performance: 17 Jan 2025**
Subject Name: Project-Based Learning in Java with Lab **Subject Code: 22CSH-359**

1. **Aim:** This project aims to design and implement a simple inventory control system for a small video rental store. Define least two classes: a class Video to model a video and a class VideoStore to model the

   actual store.

   Assume that an object of class Video has the following attributes:

   1. A title;
   2. a flag to say whether it is checked out or not;
   3. An average user rating.

   Add instance variables for each of these attributes to the Video class.

   In addition, you will need to add methods corresponding to the following:

   1. being checked out;
   2. being returned; 3. receiving a rating.

   The VideoStore class will contain at least an instance variable that references an array of videos (say of length 10). The VideoStore will contain the following methods:

   1. addVideo(String): add a new video (by title) to the inventory;
   2. checkOut(String): check out a video (by title);
   3. returnVideo(String): return a video to the store;
   4. receiveRating(String, int) : take a user's rating for a video; and 5. listInventory(): list the whole inventory of videos in the store.

2. **Objective:** Create a VideoStoreLauncher class with a main() method which will test the functionality of your other two classes. It should allow the following.

   1. Add 3 videos: "The Matrix", "Godfather II", "Star Wars Episode IV: A New Hope".
   2. Give several ratings to each video.
   3. Rent each video out once and return it.

   List the inventory after "Godfather II" has been rented out.

3. Implementation/Code:

1. **Video Class:-**

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

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

    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.");
        }
    }
```

```
    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.");
        }
    }

    public void receiveRating(int rating) {
if (rating < 1 || rating > 5) {
            System.out.println("Invalid rating. Please rate between 1 and 5.");
return;
        }
        averageRating = (averageRating * ratingCount + rating) /
(++ratingCount);
        System.out.println("Received rating of " + rating + " for video \"" + title +
"\".");
    }

     public String getTitle() {
        return title;
    }

    public boolean isCheckedOut() {
        return checkedOut;
    }

    public double getAverageRating() {
        return averageRating;
    }
}
```

2. VideoStore Class:- class

```
VideoStore { private
Video[] videos;
  private int count;

  public VideoStore(int capacity) {
videos = new Video[capacity];
    count = 0;
  }

  public void addVideo(String title) {
if (count < videos.length) {
      videos[count++] = new Video(title);
System.out.println("Added video: " + title);
    } else {
      System.out.println("Inventory is full. Cannot add more videos.");
    }
  }

  public void checkOut(String title) {
Video video = findVideo(title);
    if (video != null) {
video.checkOut();
    } else {
      System.out.println("Video \"" + title + "\" not found.");
    }
  }

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

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

    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.");
        }
    }

    public void listInventory() {
System.out.println("\nInventory:");         for
(int i = 0; i < count; i++) {
            Video video = videos[i];
            System.out.println("Title: " + video.getTitle() + ", Checked Out: " +
video.isCheckedOut() +
                    ", Average Rating: " + video.getAverageRating());
        }
    }

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

3. **VideoStoreLauncher Class:-** public class VideoStoreLauncher { public static void main(String[] args) { VideoStore store = new VideoStore(10);

```
        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:**

```
Added video: The Matrix
Added video: Godfather II
Added video: Star Wars Episode IV: A New Hope
Received rating of 5 for video "The Matrix".
Received rating of 4 for video "Godfather II".
Received rating of 5 for video "Star Wars Episode IV: A New Hope".
Video "Godfather II" has been checked out.
Video "Godfather II" has been returned.

Inventory:
Title: The Matrix, Checked Out: false, Average Rating: 5.0
Title: Godfather II, Checked Out: false, Average Rating: 4.0
Title: Star Wars Episode IV: A New Hope, Checked Out: false, Average Rating: 5.0


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

5. **Learning Outcomes:**
    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.