Experiment No: 1.2

Student Name: Kamal Ale Magar **UID: 21BCS10155**
Branch: CSE **Section/Group: 616/B**
Semester: 6th **Date of Performance: 01/20/24**
Subject: Project Based Learning in Java with Lab **Subject Code: 21CSH-319**

1. Aim:

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

Task:

A Video Rental Inventory System The goal of this project is 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.Video has the following attributes:
A title, a flag to say whether it is checked out or not and an average
user rating. .In addition, you will need to add methods corresponding
to the following:
1. being checked out; being returned; and receiving a rating.
The VideoStore class will contain at least an instance variable that references an array of videos (say of length10). 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.

Finally, create a VideoStoreLauncher class with a main() method which will test the functionality of yourother 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.

2. Objective:

The main objective is to simulate a basic video store management system where users can add videos, check out and return videos, provide ratings, and view the current inventory. The program runs in a loop until the user chooses to exit.

3. Input/Apparatus Used:

VS Code

4. Procedure/Algorithm/Pseudocode:

Step 1: Define a class Video with attributes title, isCheckedOut, and rating. Provide a constructor to initialize these attributes.

Step 2: Define a class VideoStore that extends the Video class. Add an array videos to store Video objects.

Step 3: Implement a method addVideo in the VideoStore class to add a new video to the inventory.

Step 4: Implement a method checkOutVideo in the VideoStore class to check out a video.

Step 5: Implement a method returnVideo in the VideoStore class to return a video.

Step 6 Implement a method receiveRating in the VideoStore class to receive and set a rating for a video.

Step 7 Implement a method listInventory in the VideoStore class to display information about all videos in the inventory.

Step 8: Create a main method in the VideoLauncher class for program execution. Initialize a VideoStore object with a Scanner object for user input.

Step 9: Use a while loop for the main menu. Use a switch statement to handle different choices.

Step 10: For each choice, interact with the VideoStore object to perform the corresponding action.

.

5. Code:

```
import java.util.Scanner;

class Video {
    String title;
    boolean isCheckedOut;
    int rating;

    public Video(String title) {
        this.title = title;
        isCheckedOut = false;
        rating = 0;
    }
}

class VideoStore extends Video {
    Video[] videos;

    public VideoStore() {
        super("");
        videos = new Video[10];
    }

    public void addVideo(String title) {
        Video video = new Video(title);
        for (int i = 0; i < videos.length; i++) {
            if (videos[i] == null) {
                videos[i] = video;
                System.out.println("Video added successfully: " + title);
                return;
            }
        }
        System.out.println("Video store is full. Cannot add more videos.");
    }

    public void checkOutVideo(String title) {
```

```
    for (Video video : videos) {
        if (video != null && video.title.equalsIgnoreCase(title)) {
            if (!video.isCheckedOut) {
                video.isCheckedOut = true;
                System.out.println("Video checked out successfully: " + title);
            } else {
                System.out.println("Video is already checked out: " + title);
            }
            return;
        }
    }
    System.out.println("Video not found: " + title);
}

public void returnVideo(String title) {
    for (Video video : videos) {
        if (video != null && video.title.equalsIgnoreCase(title)) {
            if (video.isCheckedOut) {
                video.isCheckedOut = false;
                System.out.println("Video returned successfully: " + title);
            } else {
                System.out.println("Video is not checked out: " + title);
            }
            return;
        }
    }
    System.out.println("Video not found: " + title);
}

public void receiveRating(String title, int rating) {
   for (Video video : videos) {
        if (video != null && video.title.equalsIgnoreCase(title)) {
            if (rating >= 1 && rating <= 5) {
                video.rating = rating;
                System.out.println("Rating added successfully for " + title + ": " + rating);
            } else {
                System.out.println("Invalid rating. Please enter a rating between 1 and 5.");
            }
            return;
        }
    }
    System.out.println("Video not found: " + title);
}
```

```
    public void listInventory() {
        System.out.println("Video Inventory:");
        for (Video video : videos) {
            if (video != null) {
                System.out.println("Title: " + video.title +
                        ", Checked Out: " + (video.isCheckedOut ? "Yes" : "No") +
                        ", Rating: " + video.rating);
            }
        }
    }
}

public class VideoLauncher {
    public static void main(String[] args) {
        VideoStore videoStore = new VideoStore();
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.println("\nChoose an option:");
            System.out.println("1. Add Videos");
            System.out.println("2. Check Out Videos");
            System.out.println("3. Return Videos");
            System.out.println("4. Receive Rating");
            System.out.println("5. List Inventory");
            System.out.println("6. Exit");

            int choice = scanner.nextInt();

            switch (choice) {
                case 1:
                    System.out.println("Enter video title to add:");
                    scanner.nextLine();
                    String addTitle = scanner.nextLine();
                    videoStore.addVideo(addTitle);
                    break;

                case 2:
                    System.out.println("Enter video title to check out:");
                    scanner.nextLine();
                    String checkOutTitle = scanner.nextLine();
                    videoStore.checkOutVideo(checkOutTitle);
                    break;

                case 3:
```

```
                System.out.println("Enter video title to return:");
                scanner.nextLine();
                String returnTitle = scanner.nextLine();
                videoStore.returnVideo(returnTitle);
                break;

            case 4:
                System.out.println("Enter video title to receive rating:");
                scanner.nextLine();
                String ratingTitle = scanner.nextLine();
                System.out.println("Enter rating (1-5):");
                int rating = scanner.nextInt();
                videoStore.receiveRating(ratingTitle, rating);
                break;

            case 5:
                videoStore.listInventory();
                break;

            case 6:
                System.out.println("Exiting the program. Goodbye!");
                System.exit(0);

            default:
                System.out.println("Invalid choice. Please enter a valid option.");
        }
    }
}
```

6. Result/Output:

```
Choose an option:
1. Add Videos
2. Check Out Videos
3. Return Videos
4. Receive Rating
5. List Inventory
6. Exit
1
Enter video title to add:
Kamal
Video added successfully: Kamal

Choose an option:
1. Add Videos
2. Check Out Videos
3. Return Videos
4. Receive Rating
5. List Inventory
6. Exit
2
Enter video title to check out:
Kamal
Video checked out successfully: Kamal

Choose an option:
1. Add Videos
2. Check Out Videos
3. Return Videos
4. Receive Rating
5. List Inventory
6. Exit
3
Enter video title to return:
Kamal
```

```
Video returned successfully: Kamal

Choose an option:
1. Add Videos
2. Check Out Videos
3. Return Videos
4. Receive Rating
5. List Inventory
6. Exit
4
Enter video title to receive rating:
hello
Enter rating (1-5):
4
Video not found: hello

Choose an option:
1. Add Videos
2. Check Out Videos
3. Return Videos
4. Receive Rating
5. List Inventory
6. Exit
5
Video Inventory:
Title: Kamal, Checked Out: No, Rating: 0

Choose an option:
1. Add Videos
2. Check Out Videos
3. Return Videos
4. Receive Rating
5. List Inventory
6. Exit
```

7. Learning Outcomes:

- Understood the principles of encapsulation, as demonstrated by the Video class with private attributesand methods.
- Learnt how to use the ArrayList class to manage a dynamic list of Video objects.
- Gain experience in handling user input using the Scanner class.
- Understood the use of loops, specifically the do-while loop, for presenting a menu and processing userchoices until they choose to exit.
- Gain experience in error handling and validation for user input, such as consuming the newline characterafter reading an integer.