Experiment 2

Student Name: Manav Raj **UID:** 22BCS12121

Branch: BE-CSE **Section/Group:** KRG_IOT_2_B

Semester: 6th **Date of Performance:** 11-01-25

Subject Name: PBLJ Lab **Subject Code:** 22CSH-359

1. **Aim:** Design and implement a simple inventory control system for a small video rental store.

2. **Objective:**
   - To learn about concept of inheritance, objects and classes.
   - To learn about arrays in Java.
   - To learn about access specifiers in java.

3. **Problem Statement:** 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 class attributes: A title, a flag to say whether it is checked out or not, An average user rating.

   Video class methods: being checked out , being returned, receiving a rating.

   VideoStore class methods: addVideo(String), checkOut(String), returnVideo(String), receiveRating(String, int), listInventory().

   VideoStoreLauncher class methods: main().

4. **Algorithm:**
   a) Create a class Video.java with the mentioned attributes and methods.
   b) Create a class VideoStore.java with the mentioned attributes and methods.
   c) Now inherit the Video.java in VideoStore.java.
   d) Create an array of type Video in VideoStore.java to maintain inventory.
   e) Create a class VideoStoreLauncher.java with the main methods, and inherit the VideoStore.java.

f) Keep in mind the different access specifiers and implement the methods respectively.

5. Implementation/Code:

Video.java

```
package Practical1;

public class Video {
    String title;
    boolean isRented;
    int ratingsCount = 0;
    double avgRating;

    static int rentedIndex = 0, availableIndex = 0;
    static String[] rentedVideos = new String[10];
    static String[] availableVideos = new String[10];

    public Video() {
        // call to default
    }

    public Video(String title, boolean isRented, double avgRating) {
        this.title = title;
        this.isRented = isRented;
        this.avgRating = avgRating;
    }

    public static void beingReturned(String title) {
        int i = 0;
        boolean found = false;

        while (i < rentedIndex) {
            if (rentedVideos[i].equals(title)) {
                found = true;
                break;
            }
            i++;
        }
        if (found) {
            availableVideos[availableIndex++] = rentedVideos[i];
            System.arraycopy(rentedVideos, i + 1, rentedVideos, i, rentedIndex - i - 1);
            rentedVideos[--rentedIndex] = null;
        } else {
            System.out.println("Video not among checked-out ones");
        }
    }

    public static void beingCheckedOut(String title) {
        int i = 0;
        boolean found = false;

        while (i < availableIndex) {
            if (availableVideos[i].equals(title)) {
                found = true;
                break;
            }
            i++;
        }

        if (found) {
            rentedVideos[rentedIndex++] = availableVideos[i];
            System.arraycopy(availableVideos, i + 1, availableVideos, i, availableIndex - i - 1);
            availableVideos[--availableIndex] = null;
        } else {
            System.out.println("Video already rented or not available");
        }
    }
}
```

VideoStore.java

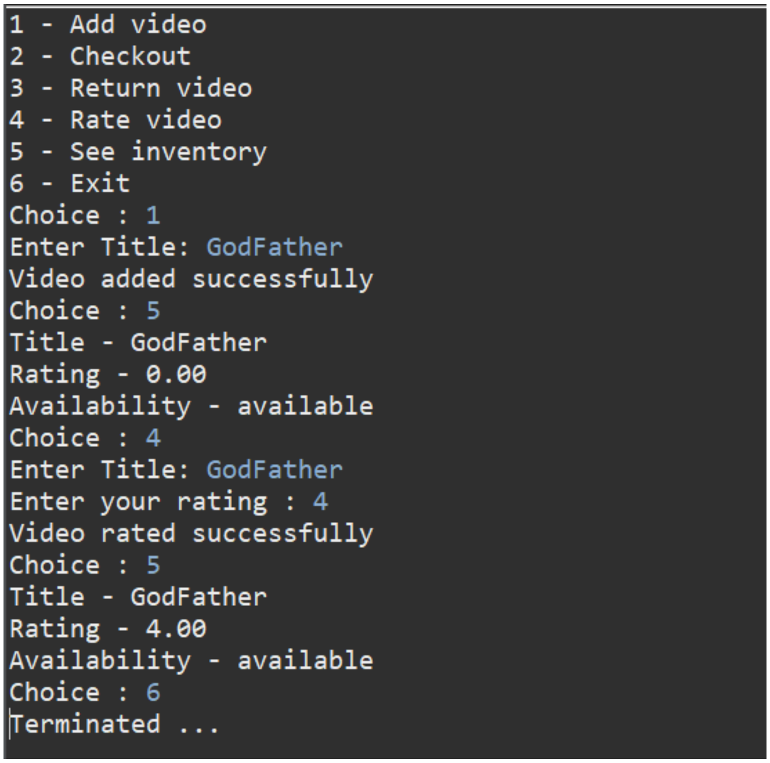
```
package Practical1;
public class VideoStore extends Video {
    Video[] inventory = new Video[10];
    static int inventoryIndex = 0;

    public VideoStore() {
        super();
```

```
}

public VideoStore(String title, boolean isRented, double avgRating) {
    super(title, isRented, avgRating);
}

void checkInventory() {
    int i = -1;
    while (++i < inventoryIndex){
        System.out.println("Title - " + inventory[i].title);
        System.out.println("Rating - " + String.format("%.2f", inventory[i].avgRating));
        System.out.println("Availability - "+(inventory[i].isRented?"rented":"available"));
    }
    if(i == 0 )
              System.out.println("Inventory Empty");
}

void addVideo(String title) {
    if (inventoryIndex < 10) {
        inventory[inventoryIndex++] = new Video(title, false, 0.0);
        availableVideos[availableIndex++] = title;
        System.out.println("Video added successfully");
    } else
        System.out.println("Inventory full");
    }

void checkOut(String title) {
    int i = 0;
    boolean found = false;
    while (i < inventoryIndex) {
        if (inventory[i].title.equalsIgnoreCase(title) && !inventory[i].isRented) {
            inventory[i].isRented = true;
            Video.beingCheckedOut(title);
            System.out.println("Video checked out successfully");
            found = true;
            break;
        }
        i++;
        }
        if (!found)
            System.out.println("Video not found or already checked out");
    }
    void returnVideo(String title) {
        int i = 0;
        boolean found = false;
        while (i < inventoryIndex) {
            if (inventory[i].title.equalsIgnoreCase(title) && inventory[i].isRented) {
                inventory[i].isRented = false;
                Video.beingReturned(title);
                System.out.println("Video returned successfully");
                found = true;
                break;
            }i++;
        }
        if (!found)
            System.out.println("Video not found or already returned");
    }

    void receiveRating(String title, int rating) {
        int i = 0;
        boolean found = false;
        while (i < inventoryIndex) {
            if (inventory[i].title.equalsIgnoreCase(title)) {
                inventory[i].avgRating =
                ((inventory[i].avgRating * inventory[i].ratingsCount)+rating)/
                (++inventory[i].ratingsCount);
                System.out.println("Video rated successfully");
                found = true;
                break;
            }i++;  }
        if (!found) {
            System.out.println("Video not found");
        }
    }
}
```

VideoStoreLauncher.java

```
package Practical1;
import java.util.Scanner;
public class VideoStoreLauncher {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        VideoStore videoStore = new VideoStore();
        System.out.println("1 - Add video\n2 - Checkout\n3 - Return video\n4 - Rate video"
                          + "\n5 - See inventory\n6 - Exit");
        while (true) {
            System.out.print("Choice : ");
            String input = scanner.next();
            scanner.nextLine(); // Consume the newline character left by next()
            switch (input) {
                case "1":
                    System.out.print("Enter Title: ");
                    String title1 = scanner.nextLine();
                    videoStore.addVideo(title1);
                    break;
                case "2":
                    System.out.print("Enter Title: ");
                    String title2 = scanner.nextLine();
                    videoStore.checkOut(title2);
                    break;
                case "3":
                    System.out.print("Enter Title: ");
                    String title3 = scanner.nextLine();
                    videoStore.returnVideo(title3);
                    break;
                case "4":
                    System.out.print("Enter Title: ");
                    String title4 = scanner.nextLine();
                    System.out.print("Enter your rating : ");
                    int rating = scanner.nextInt();
                    scanner.nextLine();
                    videoStore.receiveRating(title4, rating);
                    break;
                case "5":
                    videoStore.checkInventory();
                    break;
                case "6":
                    System.out.println("Terminated ... ");
                    scanner.close();
                    System.exit(0);
                default:
                    System.out.println("Invalid option, please try again.");
            }
        }
    }
}
```

6. Output:



```
1 - Add video
2 - Checkout
3 - Return video
4 - Rate video
5 - See inventory
6 - Exit
Choice : 1
Enter Title: GodFather
Video added successfully
Choice : 5
Title - GodFather
Rating - 0.00
Availability - available
Choice : 4
Enter Title: GodFather
Enter your rating : 4
Video rated successfully
Choice : 5
Title - GodFather
Rating - 4.00
Availability - available
Choice : 6
Terminated ...
```

7. Learning Outcomes:

a) Learned about inheritance in Java.

b) Learned about access specifiers in Java.

c) Learned about using custom data types as data types of array in Java.