Experiment 2

Student Name: Anjali Singh **UID:20BCS9239**
Branch: CSE **Section/Group:607/A**
Semester: 5th **Date of Performance:06/09/2022**
Subject Name: PBLJ Lab **Subject Code: 20CSP-321**

1. **Aim/Overview of the practical:** Implement Exploratory Data Analysis on any data set.

2. **Task to be done/ Which logistics used:**
   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. 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; and
   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; and
   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.

Finally, 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. Algorithm/Flowchart (For programming based labs):

1.START
2.Create different methods for different activity like add video, rent video, return video, rating…etc
3.Create main method for taking input from users.
4.Create a switch case in main method for calling different methods out of main method one by one according to user choices.
5.Set default of switch case with printing exiting….
6.END

4. Steps for experiment/practical/Code:

```
package com.company.CWH;

import java.util.Objects;
import java.util.Scanner;

class Video {
    String videoName;
    boolean checkout;
    int rating;

    public Video(String name) {
        videoName = name;
    }

    public String getName() {
```

```
        return videoName;
    }

    public boolean getCheckout() {
        return checkout;
    }

    public void doReturn() {
        checkout = true;
        System.out.println("Video: \"" + getName() + "\" returned successfully.");
    }

    public void receiveRating(int r) {
        rating = r;
    }

    public int getRating() {
        return rating;
    }
}

class VideoStore {
    Video[] videoL;
    static int i = 0;

    public VideoStore() {
        videoL = new Video[10];
    }

    public void addVideo(String name) {
        if (i == 10) {
            System.out.println("Sorry, we can't add more videos.");
        } else {
            videoL[i] = new Video(name);
            i++;
            System.out.println("Video: \"" + name + "\" is added successfully.");
        }
    }

    public void doCheckout(String name) {
        boolean out = false;
        for (int j = 0; j < i; j++) {
            if (Objects.equals(videoL[j].getName(), name)) {
                videoL[j].checkout = true;
                out = true;
            }
        }
        if (!out) {
            System.out.println("Video unavailable.");
        } else {
            System.out.println("Thank you for renting " + name);
        }
    }

    public void doReturn(String name) {

        if (i == 0) {
            System.out.println("No videos available.");
        } else {
```

```
            for (int j = 0; j < i; j++) {
                if (Objects.equals(videoL[j].getName(), name)) {
                    videoL[j].doReturn();
                    break;
                }
            }
        }
    }

    public void receiveRating(String name, int r) {
        if (i == 0) {
            System.out.println("No videos available.");
        } else {
            for (int j = 0; j < i; j++) {
                if (Objects.equals(videoL[j].getName(), name)) {
                    videoL[j].receiveRating(r);
                    System.out.println("Video: \"" + videoL[j].getName() + "\" has mapped with the
rating of " +
                            videoL[j].getRating() + ".");
                    break;
                }
            }
        }
    }

    public void listInventory() {
        if (i == 0) {
            System.out.println("No videos available.");
        } else {
            System.out.format("%-15s | %-15s | %-10s", "Video Name", "Checkout Status", "Rating");
            System.out.println();
            for (int j = 0; j < i; j++) {
                System.out.format("%-15s | %-15b | %-10d", videoL[j].getName(),
videoL[j].getCheckout(),
                        videoL[j].getRating());
                System.out.println();
            }
        }
    }
}

public class main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int c;
        System.out.println("Anjali Singh 20BCS9239");
        VideoStore vs = new VideoStore();

        do {
            System.out.println("**********");
            System.out.println("1. Add video ");
            System.out.println("2. Checkout a video ");
            System.out.println("3. return video to store ");
            System.out.println("4. Take rating for a video ");
            System.out.println("5. List Inventory ");
            System.out.println("6. Exit");
            System.out.print("Enter choice: ");
            c = sc.nextInt();
            switch (c) {
```

```
                case 1: {
                    System.out.print("Enter the name of Video to add: ");
                    vs.addVideo(sc.next());
                }
                break;
                case 2: {
                    System.out.print("Enter the name of Video to checkout: ");
                    vs.doCheckout(sc.next());
                }
                break;
                case
                        3: {
                    System.out.print("Enter the name of Video to return: ");
                    vs.doReturn(sc.next());
                }
                break;
                case
                        4: {
                    System.out.print("Enter the name of Video to rate: ");
                    vs.receiveRating(sc.next(), sc.nextInt());
                }
                break;
                case 5:
                    vs.listInventory();
                    break;
                case
                        6: {
                    System.out.println("Exiting...");
                    System.exit(1);
                }
                break;
            }
        } while (c != 6);
        sc.close();
    }
}
```

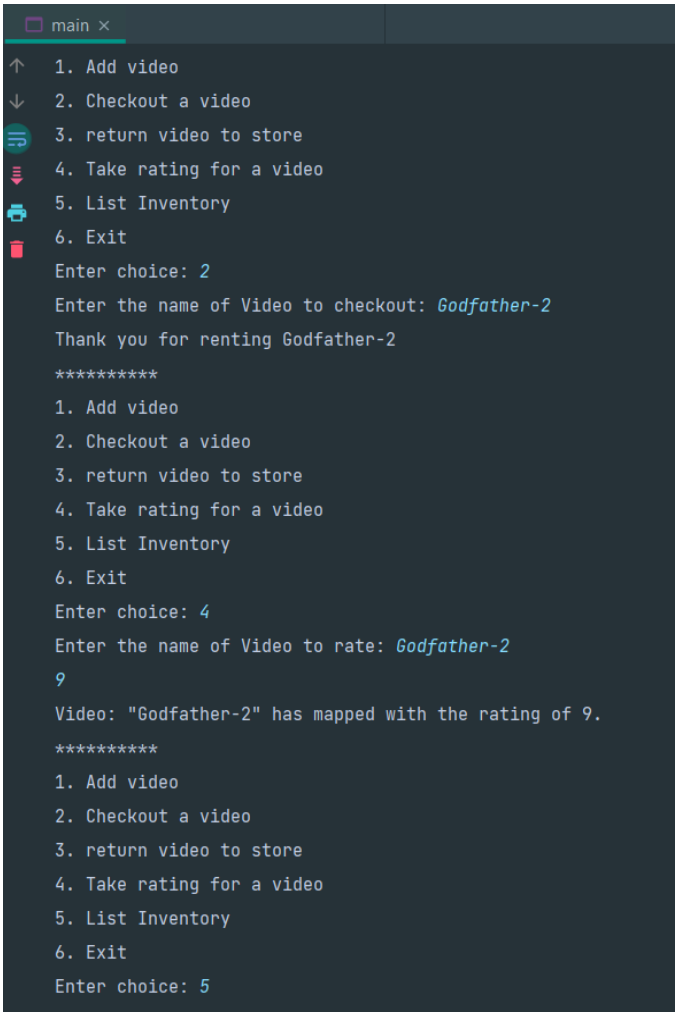
4. Observations/Discussions/ Complexity Analysis:

Here we are creating movie inventory system which store movie its rating its information related to checkout and return. We are doing all this by using methods of java by creating separate method for each work. We are also using switch case for providing user a number of choices of functions to do with our movie inventory system.

5. Result/Output/Writing Summary:

```
main ×
"C:\Program Files\Java\jdk-16.0.2\bin\java.exe" "-javaagent:C:
 Community Edition 2021.2\bin" -Dfile.encoding=UTF-8 -classpat
Anjali Singh 20BCS9239
**********
1. Add video
2. Checkout a video
3. return video to store
4. Take rating for a video
5. List Inventory
6. Exit
Enter choice: 1
Enter the name of Video to add: The_Matrix
Video: "The_Matrix" is added successfully.
**********
1. Add video
2. Checkout a video
3. return video to store
4. Take rating for a video
5. List Inventory
6. Exit
Enter choice: 4
Enter the name of Video to rate: The_Matrix
7
Video: "The_Matrix" has mapped with the rating of 7.
**********
1. Add video
2. Checkout a video
3. return video to store
```

```
main ×
3. return video to store
4. Take rating for a video
5. List Inventory
6. Exit
Enter choice: 1
Enter the name of Video to add: Godfather-2
Video: "Godfather-2" is added successfully.
**********
1. Add video
2. Checkout a video
3. return video to store
4. Take rating for a video
5. List Inventory
6. Exit
Enter choice: 1
Enter the name of Video to add: Star-wars-5
Video: "Star-wars-5" is added successfully.
**********
1. Add video
2. Checkout a video
3. return video to store
4. Take rating for a video
5. List Inventory
6. Exit
Enter choice: 4
Enter the name of Video to rate: Star-wars-5
8
Video: "Star-wars-5" has mapped with the rating of 8.
**********
```



Learning outcomes (What I have learnt):

1. I learn how to use intellij idea for executing java programs.
2. I learn basics related to java program implementation.
3. I learn to store information using array in java.
4. I learn working of switch case.
5. I learn how to use methods and how to call methods in java.