Experiment - 1.2

Name: Debdulal Das **UID: 21BCS9011**
Branch: BE- CSE **Section/Group: CC_647-A**
Semester: 6th **Date of Performance:25-01-2024**
Subject Code: 21CSH-319
Subject Name: Project based learning in Java with Lab

1. **Aim:** Design and implement a simple inventory control system for a small video rental
2. **Objective:** 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.
3. **Algorithm:**

- Start
- Create a data structure to store movie information, such as title, genre, availability status, and rental details.
- Display a menu with options for users, such as:
  Display available movies.
  Rent a movie.
  Return a movie.
  Display rental history.
  Exit the system.
- Prompt the user to enter the title of the movie they want to rent.
  Check if the movie is available.
  If available, mark the movie as rented, record rental details (user and date), and update the inventory.
  If not available, notify the user.
- Display Rental History: Display a list of rented movies along with rental details (user and date).
- Exit the System: Terminate the program.
- Repeat: Loop back to the main menu after completing an operation until the user chooses to exit.
- End.

4. Code:

```
package program2;
import java.util.Arrays;
class Video {
    private String title;
    private boolean checkedOut;
    private double averageRating;
    public Video(String title) {
        this.title = title;
        this.checkedOut = false;
        this.averageRating = 0.0;
    }
    public String getTitle() {
        return title;
    }
    public boolean isCheckedOut() {
        return checkedOut;
    }
    public void checkOut() {
        checkedOut = true;
    }
    public void returnVideo() {
        checkedOut = false;
    }
    public void receiveRating(int rating) {
        averageRating = (averageRating + rating) / 2.0;
    }
    @Override
    public String toString() {return "Video{" + "title='" + title + '\'' + ", checkedOut=" + checkedOut +
 ", averageRating=" +
averageRating +
 '}';
 }
}
class VideoStore {
    private Video[] inventory;
    public VideoStore() {
        this.inventory = new Video[10];
    }
    public void addVideo(String title) {
        for (int i = 0; i < inventory.length; i++) {
            if (inventory[i] == null) {
                inventory[i] = new Video(title);
                break;
            }
        }
```

```
}
 public void checkOut(String title) {
    for (Video video : inventory) {
       if (video != null && video.getTitle().equals(title) && !video.isCheckedOut()) {
          video.checkOut();
          break;
       }
    }
 }
 public void returnVideo(String title) {
    for (Video video : inventory) {
       if (video != null && video.getTitle().equals(title) && video.isCheckedOut()) {
          video.returnVideo();
          break;
       }
    }
 }
 public void receiveRating(String title, int rating) {
    for (Video video : inventory) {
       if (video != null && video.getTitle().equals(title)) {
          video.receiveRating(rating);
          break;
       }
    }
 }
 public void listInventory() {
    System.out.println("Inventory:");
    for (Video video : inventory) {
       if (video != null) {
          System.out.println(video);
        }
      }
   }
}
public class VideoStoreLauncher {
   public static void main(String[] args) {
      VideoStore videoStore = new VideoStore();
      videoStore.addVideo("The Matrix");
      videoStore.addVideo("Godfather II");
      videoStore.addVideo("Star Wars Episode IV: A New Hope");
      videoStore.receiveRating("The Matrix", 5);
      videoStore.receiveRating("The Matrix", 4);
      videoStore.receiveRating("Godfather II", 5);
      videoStore.receiveRating("Godfather II", 4);
      videoStore.receiveRating("Star Wars Episode IV: A New Hope", 5);
      videoStore.receiveRating("Star Wars Episode IV: A New Hope", 4);
```

```
videoStore.checkOut("The Matrix");
        videoStore.checkOut("Godfather II");
        videoStore.checkOut("Star Wars Episode IV: A New Hope");
        videoStore.returnVideo("Godfather II");
        videoStore.listInventory();
    }
}
```

5. Output:

6. Learning Outcome:

- Learn how to code in java environment.
- Learnt about classes ofjava.
- Learnt OOPs concepts.
- Learnt about eclipse.