Experiment-2

Student Name: Ayush Saini **UID: 22BCS12499**
Branch: BE-CSE **Section/Group: 902-B**
Semester: 6th **Date of Performance: 25/01/2025**
Subject Name: PBLJ **Subject Code: 22CSP-359**

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

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 Video Store to model the actual store.

3. **Implementation/Code:**

Code:

Video.java

```
class Video {
    private String title;
    private boolean checkedOut;
    private double rating;

    public Video(String title) {
        this.title = title;
        this.checkedOut = false;
        this.rating = 0.0;
    }

    public String getTitle() {
        return title;
    }

    public boolean isCheckedOut() {
        return checkedOut;
    }

    public void checkOut() {
        this.checkedOut = true;
    }
```

```
    public void returnVideo() {
        this.checkedOut = false;
    }

    public void setRating(double rating) {
        this.rating = rating;
    }

    public double getRating() {
        return rating;
    }

    @Override
    public String toString() {
        return "Title: " + title + ", Checked Out: " + (checkedOut ? "Yes" : "No") + ", Rating:
" + rating;
    }
}
```

VideoStore.java

```
import java.util.ArrayList;
import java.util.List;

class VideoStore {
    private List<Video> inventory;

    public VideoStore() {
        inventory = new ArrayList<>();
    }

    public void addVideo(String title) {
        Video video = new Video(title);
        inventory.add(video);
        System.out.println("Video \"" + title + "\" added successfully.");
    }
    public void checkOutVideo(String title) {
Video video = findVideoByTitle(title);
```

```
        if (video != null) {
            if (!video.isCheckedOut()) {
                video.checkOut();
                System.out.println("Video \"" + title + "\" checked out.");
            } else {
                System.out.println("Video \"" + title + "\" is already checked out.");
            }
        } else {
            System.out.println("Video \"" + title + "\" not found.");
        }
    }

    public void returnVideo(String title) {
        Video video = findVideoByTitle(title);
        if (video != null) {
            if (video.isCheckedOut()) {
                video.returnVideo();
                System.out.println("Video \"" + title + "\" returned.");
            } else {
                System.out.println("Video \"" + title + "\" was not checked out.");
            }
        } else {
            System.out.println("Video \"" + title + "\" not found.");
        }
    }

    public void receiveRating(String title, double rating) {
        Video video = findVideoByTitle(title);
        if (video != null) {
            video.setRating(rating);
            System.out.println("Rating added to video \"" + title + "\".");
        } else {
            System.out.println("Video \"" + title + "\" not found.");
        }}
    public void listInventory() {
```

```
        System.out.println("Inventory:");
            for (Video video : inventory) {
                System.out.println(video);
            }
        }

        private Video findVideoByTitle(String title) {
            for (Video video : inventory) {
                if (video.getTitle().equalsIgnoreCase(title)) {
                    return video;
                }
            }
            return null;
        }
}
```

VideoLauncher.java

```
    import java.util.Scanner;

    public class VideoLauncher {
        public static void main(String[] args) {
            VideoStore store = new VideoStore();
            Scanner scanner = new Scanner(System.in);
            int choice = 0;

            while (choice != 6) {
                System.out.println("Enter your choice: ");
                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");
```

```
        choice = scanner.nextInt();
                scanner.nextLine();  // Consume the newline

                switch (choice) {
                    case 1:
                        System.out.print("Enter the title of the video to add: ");
                        String titleToAdd = scanner.nextLine();
                        store.addVideo(titleToAdd);
                        break;
                    case 2:
                        System.out.print("Enter the title of the video to check out: ");
                        String titleToCheckOut = scanner.nextLine();
                        store.checkOutVideo(titleToCheckOut);
                        break;
                    case 3:
                        System.out.print("Enter the title of the video to return: ");
                        String titleToReturn = scanner.nextLine();
                        store.returnVideo(titleToReturn);
                        break;
                    case 4:
                        System.out.print("Enter the title of the video to rate: ");
                        String titleToRate = scanner.nextLine();
                        System.out.print("Enter the rating (0.0 to 5.0): ");
                        double rating = scanner.nextDouble();
                        store.receiveRating(titleToRate, rating);
                        break;
                    case 5:
                        store.listInventory();
                        break;
                    case 6:
                        System.out.println("Exiting the application.");
                        break;
                    default:
                        System.out.println("Invalid choice, please try again.");
                }}
```

```
            scanner.close();
        }
}
```

4. OUTPUT:

```
Enter your choice:
1. Add Videos
2. Check Out Videos
3. Return Videos
4. Receive Rating
5. List Inventory
6. Exit
1
Enter the title of the video to add: Project
Video "Project" added successfully.
```

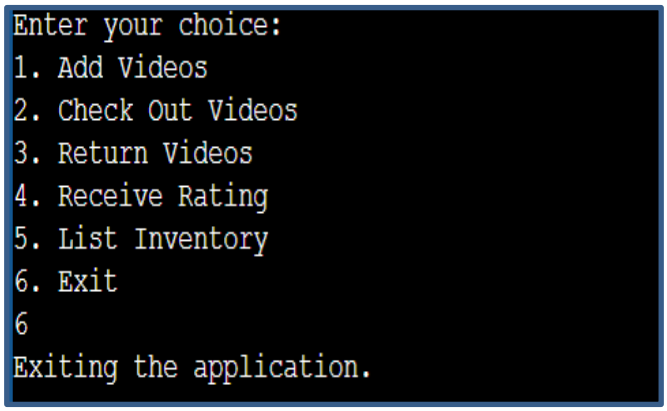
```
Enter your choice:
1. Add Videos
2. Check Out Videos
3. Return Videos
4. Receive Rating
5. List Inventory
6. Exit
1
Enter the title of the video to add: Base Java
Video "Base Java" added successfully.
```

```
Enter your choice:
1. Add Videos
2. Check Out Videos
3. Return Videos
4. Receive Rating
5. List Inventory
6. Exit
2
Enter the title of the video to check out: Project
Video "Project" checked out.
```

```
Enter your choice:
1. Add Videos
2. Check Out Videos
3. Return Videos
4. Receive Rating
5. List Inventory
6. Exit
3
Enter the title of the video to return: Base Java
Video "Base Java" was not checked out.
```

```
Enter your choice:
1. Add Videos
2. Check Out Videos
3. Return Videos
4. Receive Rating
5. List Inventory
6. Exit
4
Enter the title of the video to rate: Project
Enter the rating (0.0 to 5.0): 4.5
Rating added to video "Project".
```

```
Enter your choice:
1. Add Videos
2. Check Out Videos
3. Return Videos
4. Receive Rating
5. List Inventory
6. Exit
5
Inventory:
Title: Project, Checked Out: Yes, Rating: 4.5
Title: Base Java, Checked Out: No, Rating: 0.0
```



5. Learning Outcome

- Understand how to design classes to represent real-world entities, encapsulating properties and behavior.
- Learn how to extend a class (inheritance) and define specific behaviors in the derived class.
- Implement user-driven program flow using a switch-case structure for various menu options.
- Practice core OOP concepts such as abstraction, encapsulation, inheritance, and polymorphism.
- Learn how to take input from users, process the input, and perform operations on objects accordingly.