Experiment-2

Student Name: Om Mishra **UID: 22BCS166609**
Branch: BE-CSE **Section/Group: DL_901/A**
Semester: 6th **Date of Performance:22/1/25**
Subject Name: Project-Based Learning in Java **Subject Code: 22CSH-359**

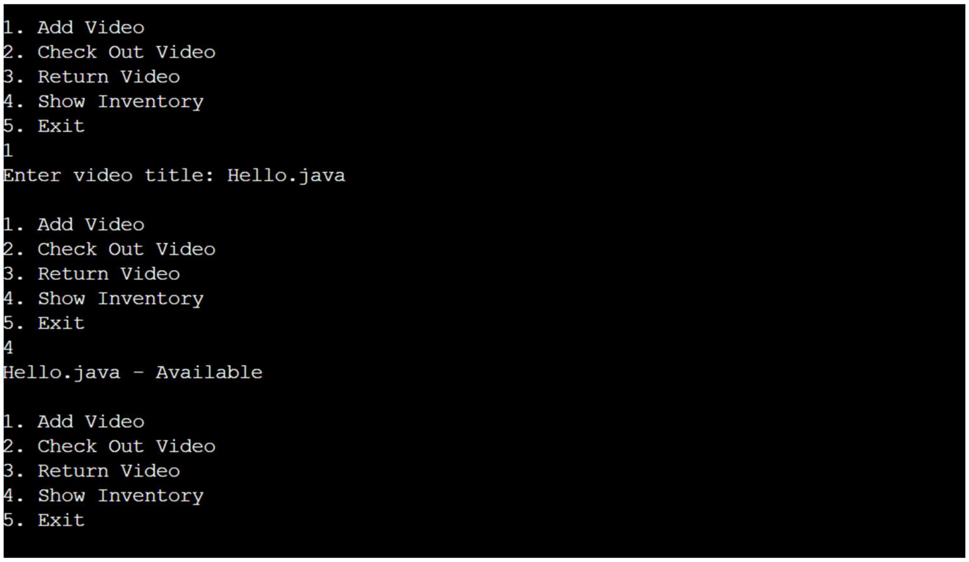
1. **Aim:** Design and implement a simple inventory control system for a small video rental store**.**

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 VideoStore to model the actual store.

3. Implementation/Code:

```
import java.util.*;

class Video {
    private String title;
    private boolean checkedOut;

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

    public String getTitle() {
        return title;
    }

    public boolean isCheckedOut() {
        return checkedOut;
    }

    public void checkOut() {
        checkedOut = true;
    }

    public void returnVideo() {
        checkedOut = false;
    }

    public String getStatus() {
        return checkedOut ? "Checked Out" : "Available";
    }
}

class Inventory {
    private List<Video> videos = new ArrayList<>();

    public void addVideo(String title) {
        videos.add(new Video(title));
    }

    public void checkOutVideo(String title) {
        for (Video v : videos) {
            if (v.getTitle().equalsIgnoreCase(title) && !v.isCheckedOut()) {
                v.checkOut();
                return;
            }
        }
    }

    public void returnVideo(String title) {
        for (Video v : videos) {
            if (v.getTitle().equalsIgnoreCase(title) && v.isCheckedOut()) {
                v.returnVideo();
                return;
            }
        }
    }

    public void listInventory() {
        for (Video v : videos) {
            System.out.println(v.getTitle() + " - " + v.getStatus());
        }
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Inventory inventory = new Inventory();
        boolean running = true;

        while (running) {
            System.out.println("\n1. Add Video\n2. Check Out Video\n3. Return Video\n4. Show Inventory\n5. Exit");
            int choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    System.out.print("Enter video title: ");
                    String title = scanner.nextLine();
                    inventory.addVideo(title);
                    break;
                case 2:
                    System.out.print("Enter video title to check out: ");
                    title = scanner.nextLine();
                    inventory.checkOutVideo(title);
                    break;
                case 3:
                    System.out.print("Enter video title to return: ");
                    title = scanner.nextLine();
                    inventory.returnVideo(title);
                    break;
                case 4:
                    inventory.listInventory();
                    break;
                case 5:
                    running = false;
                    break;
            }
        }

        scanner.close();
    }
}
```

4. Output:



```
1. Add Video
2. Check Out Video
3. Return Video
4. Show Inventory
5. Exit
1
Enter video title: Hello.java

1. Add Video
2. Check Out Video
3. Return Video
4. Show Inventory
5. Exit
4
Hello.java - Available

1. Add Video
2. Check Out Video
3. Return Video
4. Show Inventory
5. Exit
```

5. Learning Outcome

a) **Object-Oriented Programming (OOP) Concepts**: Understanding and applying key OOP principles such as classes, objects, encapsulation, and methods to model real-world entities and their behaviors.

b) **Data Structures and Arrays**: Learning how to use arrays to store and manage collections of objects, such as the video inventory in the VideoStore class.

c) **Method Implementation**: Gaining experience in defining and implementing methods to perform specific actions, such as adding videos, checking out and returning videos, and receiving ratings.

d) **Basic User Interaction**: Designing a simple user interface through the main() method in the VideoStoreLauncher class to interact with the inventory system and perform various operations.