EXPERIMENT 1.2

Student Name: Ashutosh Yadav	**UID:** 22BCS10541
Branch: BE[CSE]	**Section/Group:** DL-902(B)
Semester: 6	**Date of Performance: 14.01.2025**
Subject Name: Project Based Learning in Java	**Subject Code:** 22CSH-359

o **Aim**:

Design and implement a simple inventory control system for a small video rental store. o

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.

Assume that an object of class Video has the following attributes: A title, a flag to say whether it is checked out or not, 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: checkedout, returned, 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: addVideo(String): add a new video (by title) to the inventory. checkOut(String): check out a video (by title) returnVideo(String): return a video to the store receiveRating(String, int) : take a user's rating for a video 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:

Add 3 videos: "The Matrix", "Godfather II", "Star Wars Episode IV: A New Hope".

Give several ratings to each video.

List the inventory after "Godfather II" has been rented out.

o **Algorithm:**

- Start
- Class Video
  - Define attributes: Title, Flag, rating.
  - Define methods: checkedout, returned, setRating.
- Class VideoStore
  - Create object array of size 10.
  - Create methods: Add video, Checkout, return, receive rating and list inventory.
- Main Class
  - Create an object of VideoStore class.
  - Now you can add videos, give ratings, checkout and return the videos.
  - By calling listinventory( ), you can print it.
- End

o **Source Code:**

```
class Video { String
    Title; boolean
    flag; double
    rating;

    Video(String name) {
        Title = name;  flag
        = false; rating = 0;
    }
    public void checkedout() { flag
        = true;
    }
    public void returned() { flag
        = false;
```

```java
    }
    public void setRating(double rate) { rating
        = rate;
    }
    public String getTitle() { return
        Title;



    }
    @Override
    public String toString() { return "Title: " + Title + ", Checked Out: " + flag
        + ", Rating: " + rating;
    }
}

class VideoStore {
    Video[] inventory = new Video[10];
    int count = 0;

    public void addVideo(String n) {
        inventory[count++] = new Video(n);
    }
    public void checkOut(String name) {  for (int i = 0; i <
        count;              i++)                {              if
        (inventory[i].getTitle().equalsIgnoreCase(name))      {
        inventory[i].checkedout();
            return;
        }
    }
    System.out.println("Video not found.");
    }
    public void returnVideo(String name) { for (int i = 0; i <
        count;              i++)                {              if
        (inventory[i].getTitle().equalsIgnoreCase(name))      {
        inventory[i].returned(); return;
        }
    }
```

```java
            System.out.println("Video not found.");
        }
    public void recieveRating(String name, double r) { for (int
        i    =    0;    i    <    count;    i++)    {    if
        (inventory[i].getTitle().equalsIgnoreCase(name))    {
        inventory[i].setRating(r); return;
            }

        }
        System.out.println("Video not found.");
    }
    public void listInventory() {
        System.out.println("Inventory List:");
        for (int i = 0; i < count; i++) { if
        (inventory[i] != null) {
            System.out.println(inventory[i]);
          }
        }
      }
    }
}

class Exp2 {
    public static void main(String[] args) {
        VideoStore store = new VideoStore();
        store.addVideo("The        Matrix");
        store.addVideo("Godfather II");
        store.addVideo("Star Wars Episode IV: A New Hope");
        store.recieveRating("The matrix", 4); store.recieveRating("Godfather
        II", 4.5);
        store.recieveRating("Star    Wars    Episode    IV:    A    New    Hope",    3.5);
        store.checkOut("Godfather II");
        store.listInventory();
    }
}
```

○ **Screenshot of Outputs:**

```
Inventory List:
Title: The Matrix, Checked Out: false, Rating: 4.0
Title: Godfather II, Checked Out: true, Rating: 4.5
Title: Star Wars Episode IV: A New Hope, Checked Out: false, Rating: 3.5
PS E:\CU Syllabus\Semester 6\Project based learning in JAVA\6th Sem JAVA Lab>
```

○ **Learning Outcome:**

Understanding how to use classes (Video and VideoStore) to encapsulate data and behavior.

Recognize the limitations of fixed-size arrays and explore alternatives like ArrayList for dynamic storage.

Develop skills to manage arrays containing custom objects and track their state.

Gain experience in designing intuitive methods for common operations like adding, updating, and listing items.

Use string comparison methods (e.g., equalsIgnoreCase) for case-insensitive operations.

pply Java concepts to a real-world inventory management scenario, providing a foundation for more complex systems.