Experiment-2

Student Name: Manpreet Singh **UID: 22BCS50009**
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:**

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

  public Video(String title)
    { this.title = title;
    this.checkedOut = false;
    this.rating = 0.0; this.ratingCount
    = 0;  } public String

  getTitle() { return title;

  }

  public boolean isCheckedOut() {
    return checkedOut;
  }
```

```
    public double getRating() {
        return ratingCount > 0 ? rating / ratingCount : 0.0;
    }

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

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

    public void receiveRating(int rating) {
        this.rating += rating; this.ratingCount++;
    }
}


class VideoStore { private
    Video[] videos; private
    int videoCount;

    public VideoStore()
        { this.videos = new
        Video[10]; this.videoCount =  0;
        }

    public void addVideo(String title) { if
        (videoCount < videos.length) {
        videos[videoCount]    =         new
        Video(title);  videoCount++;
        } else {
            System.out.println("Inventory is full! Cannot add more videos."); }  }
    public void checkOut(String title) { Video video =
        findVideo(title); if (video != null
        && !video.isCheckedOut())
            { video.checkOut();
            System.out.println("Video checked out: " + title);
        } else if (video == null)
            { System.out.println("Video not found: " + title);
```

```
    } else {
        System.out.println("Video is already checked out: " + title); }
}

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

public void receiveRating(String title, int rating)
    { Video video = findVideo(title);  if
    (video != null)
        { video.receiveRating(rating)
        ;
        System.out.println("Rating " + rating + " received for video: " + title);
    } else {
        System.out.println("Video not found: " + title); }
}

public void listInventory() { for (int i =
    0; i < videoCount; i++) {
        Video video = videos[i];
        System.out.println("Title: " + video.getTitle() +
                           ", Checked Out: " + video.isCheckedOut() +
                           ", Rating: " + video.getRating());  }
}
private Video findVideo(String title)
    { for (int i = 0; i < videoCount; i++)
    { if (videos[i].getTitle().equals(title))
            { return videos[i];
        }
    }         return
    null;
}
```

```
}


public class VideoStoreLauncher
   { 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.receiveRating("The Matrix", 5);
      store.receiveRating("The Matrix", 4);
      store.receiveRating("Godfather II", 5);
      store.receiveRating("Godfather II", 3);
      store.receiveRating("Star Wars Episode IV: A New Hope", 4);

      store.receiveRating("Star Wars Episode IV: A New Hope", 5);

      store.checkOut("The Matrix");

      store.returnVideo("The Matrix"); store.checkOut("Godfather
      II");

      System.out.println("\nInventory after 'Godfather II' has been rented out:");
      store.listInventory();
      System.out.println("UID-22BCS15843,VAIBHAV");
   }
}
```

4. Output

```
Output                                                    Clear
Rating 5 received for video: The Matrix
Rating 4 received for video: The Matrix
Rating 5 received for video: Godfather II
Rating 3 received for video: Godfather II
Rating 4 received for video: Star Wars Episode IV: A New Hope
Rating 5 received for video: Star Wars Episode IV: A New Hope
Video checked out: The Matrix
Video returned: The Matrix
Video checked out: Godfather II

Inventory after 'Godfather II' has been rented out:
Title: The Matrix, Checked Out: false, Rating: 4.5
Title: Godfather II, Checked Out: true, Rating: 4.0
Title: Star Wars Episode IV: A New Hope, Checked Out: false, Rating: 4.5
UID-22BCS16843, Karan

=== Code Execution Successful ===
```

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.