Experiment 2

Student Name: RAKSHIT | UID: 22BCS12726
Branch: BE-CSE | Section/Group: IoT_631/B
Semester: 6th | Date of Performance: 22/01/25
Subject Name: Project based learning in Java with LAB | Subject Code: 22CSH-359

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

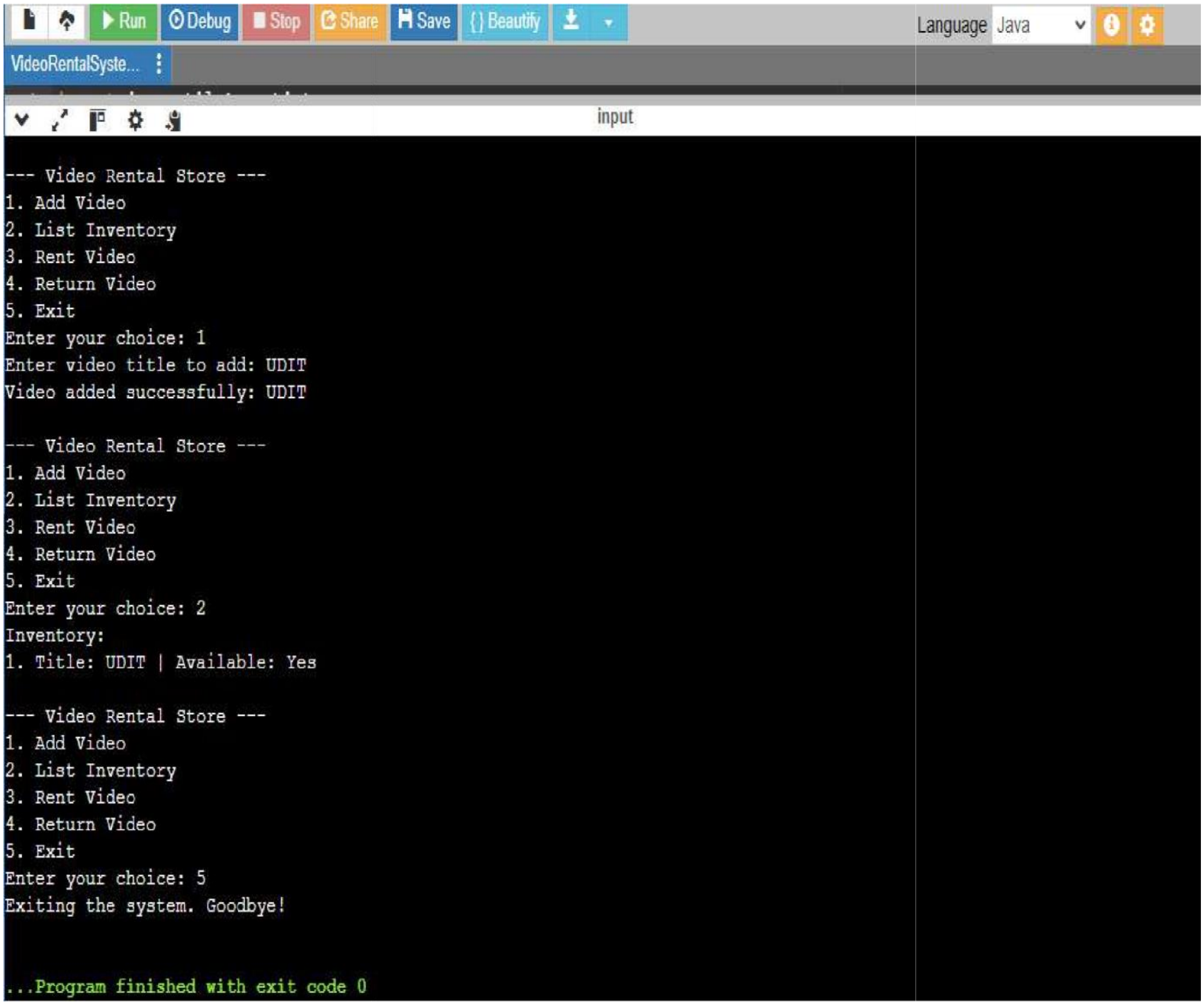
2. Objective: To design and implement a user-friendly inventory control system for a small video rental store, enabling efficient management of video inventory, including functionalities for adding, renting and returning videos.

3. Implementation/Code: import java.util.ArrayList; import java.util.Scanner; class Video { privateStringtitle; privatebooleanisAvailable; public Video(String title) { this.title = title; this.isAvailable=true;

```
} publicStringgetTitle(){
return title; }
publicbooleanisAvailable(){
return isAvailable;
}
publicvoidrent(){ if(isAvailable){
    isAvailable=false;
    }else{
        System.out.println("Error:Videoisalreadyrentedout.");
} } publicvoidreturnVideo(){ if
    (!isAvailable) {
        isAvailable=true;
    }else{
        System.out.println("Error:Videowasnotrented.");
    }
    }
    @Override
    publicStringtoString(){
        return"Title:"+title+"|Available:"+(isAvailable?"Yes":"No");
    }
}

classVideoStore{
    privateArrayList<Video>inventory;

    publicVideoStore(){
        inventory=newArrayList<>();
    }

    publicvoidaddVideo(Stringtitle){ for
        (Video video : inventory) { if
            (video.getTitle().equalsIgnoreCase(title)) {
                System.out.println("Error:Videoalreadyexistsintheinventory.");
                return;
            } }
        inventory.add(newVideo(title))
        ;
        System.out.println("Videoadde
        dsuccessfully:"+title); }

    publicvoidlistInventory(){
        if(inventory.isEmpty()){
            System.out.println("Novideosininventory.");
        }else{
            System.out.println("Inventory:"); for
            (int i = 0; i < inventory.size(); i++) {
                System.out.println((i+1)+"."+inventory.get(i));
            }
        }
    }

    publicvoidrentVideo(Stringtitle){ for
        (Video video : inventory) {
            if(video.getTitle().equalsIgnoreCase(title)){ if
                (video.isAvailable()) {
                    video.rent();
                    System.out.println("Yourented:"+title);
                }else {
                    System.out.println("Videoiscurrentlyunavailable.");
                }
                return;
            }
        }
        System.out.println("Error:Videonotfoundininventory.");
    }

    publicvoidreturnVideo(Stringtitle){
        for (Video video : inventory) {
            if(video.getTitle().equalsIgnoreCase(title)){ if
                (!video.isAvailable()) {
                     video.returnVideo();
                     System.out.println("Youreturned:"+title);
                 }else {
                     System.out.println("Error:Videowasnotrented.");
                 }
                 return;
             }
         }
        System.out.println("Error:Videonotfoundininventory."); }
}
```

```
publicclassVideoRentalSystem{
   public static void main(String[] args) {
        VideoStore store = new VideoStore();
        Scannerscanner=newScanner(System.in);
        while (true) {
            System.out.println("\n---VideoRentalStore---");
            System.out.println("1. Add Video");
            System.out.println("2. List Inventory");
            System.out.println("3. Rent Video");
            System.out.println("4. Return Video");
            System.out.println("5. Exit");
            System.out.print("Enter your choice:
            "); intchoice=-1; if
            (scanner.hasNextInt()) {
                choice=scanner.nextInt();
            }else {
                System.out.println("Invalidchoice.Pleaseenteranumber.")
            ; scanner.next(); continue; } scanner.nextLine();

           switch(choice){ case 1:
                   System.out.print("Enter video title to add:
                   ");
                   StringtitleToAdd=scanner.nextLine().trim();
                   store.addVideo(titleToAdd); break;
               case 2:
                   store.listInventory();
                   break;
               case3:
                   System.out.print("Enter video title to rent:
                   ");
                   StringtitleToRent=scanner.nextLine().trim();
                   store.rentVideo(titleToRent); break;
                case 4:
                    System.out.print("Enter video title to return:
                    ");
                    StringtitleToReturn=scanner.nextLine().trim();
                    store.returnVideo(titleToReturn); break;
                case 5:
                    System.out.println("Exitingthesystem.Goodbye!"); scanner.close();
                    return;
 default:
                    System.out.println("Invalidchoice.Pleasetryagain.");
            }
        }
    }
}
```

4. Output



```
--- Video Rental Store ---
1. Add Video
2. List Inventory
3. Rent Video
4. Return Video
5. Exit
Enter your choice: 1
Enter video title to add: UDIT
Video added successfully: UDIT

--- Video Rental Store ---
1. Add Video
2. List Inventory
3. Rent Video
4. Return Video
5. Exit
Enter your choice: 2
Inventory:
1. Title: UDIT | Available: Yes

--- Video Rental Store ---
1. Add Video
2. List Inventory
3. Rent Video
4. Return Video
5. Exit
Enter your choice: 5
Exiting the system. Goodbye!


...Program finished with exit code 0
```

5. Learning Outcomes

- Object-Oriented Design: Learn to create and use classes for real-world entities.
- Core Programming Skills: Practice loops, conditionals, and methods for inventory operations.
- Data Structure Usage: Use ArrayList to manage dynamic data effectively.
- User-Friendly Systems: Design intuitive interfaces and handle errors smoothly.