Shreyansh Saha | 17BCE1148

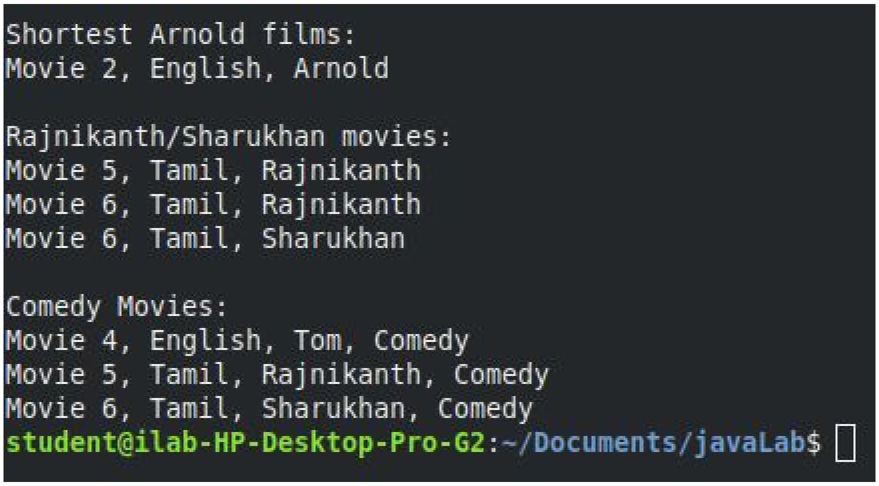
Java Lab 3

Question 1

Code

```
class Film {
 String name;
 String language;
 String leadActor;
 String category;
 int duration;

 Film(String name, String language, String leadActor, String category, int
duration) {
   this.name = name;
   this.language = language;
   this.leadActor = leadActor;
   this.category = category;
   this.duration = duration;
 }

 Film() {
   this.name = "";
   this.language = "";
   this.leadActor = "";
   this.category = "";
   this.duration = 0;
 }

 String getName() { return this.name; }
 String getLanguage() { return this.language; }
 String getleadActor() { return this.leadActor; }
 String getCategory() { return this.category; }
 int getDuration() { return this.duration; }

 void setName(String name) { this.name = name; }
 void setLanguage(String language) { this.language = language; }
 void setLeadActor(String leadActor) { this.leadActor = leadActor; }
 void setCategory(String category) { this.category = category; }
 void setDuration(int duration) { this.duration = duration;}
}

class FilmMain {
 public static void main(String[] args) {
   Film[] films = new Film[7];
   films[0] = new Film("Movie 1", "English", "Arnold", "Action", 120);
   films[1] = new Film("Movie 2", "English", "Arnold", "Action", 110);
   films[2] = new Film("Movie 3", "English", "Tom", "Action", 118);
   films[3] = new Film("Movie 4", "English", "Tom", "Comedy", 147);
   films[4] = new Film("Movie 5", "Tamil", "Rajnikanth", "Comedy", 167);
   films[5] = new Film("Movie 6", "Tamil", "Rajnikanth", "Action", 157);
   films[6] = new Film("Movie 6", "Tamil", "Sharukhan", "Comedy", 155);

   // Shortest arnold films
   int shortest = Integer.MAX_VALUE;
   for (Film film : films) {
     if (film.getleadActor().equals("Arnold") && film.getDuration() <
shortest)
       shortest = film.getDuration();
   }
   System.out.println("\nShortest Arnold films:");
   for (Film film : films) {
     if (film.getleadActor().equals("Arnold") && film.getDuration() ==
shortest)
       System.out.println(film.getName() + ", " + film.getLanguage() + ",
" + film.getleadActor());
   }
   // Tamil films with Rajnikanth/ Sharukhan
   System.out.println("\nRajnikanth/Sharukhan movies:");
   for (Film film : films) {
     if (film.getleadActor().equals("Rajnikanth") ||
film.getleadActor().equals("Sharukhan"))
       System.out.println(film.getName() + ", " + film.getLanguage() + ",
" + film.getleadActor());
   }

   // All comedy movies
   System.out.println("\nComedy Movies:");
   for (Film film : films) {
     if (film.getCategory().equals("Comedy"))
       System.out.println(film.getName() + ", " + film.getLanguage() + ",
" + film.getleadActor() + ", " + film.getCategory());
   }
 }
}
```

Output



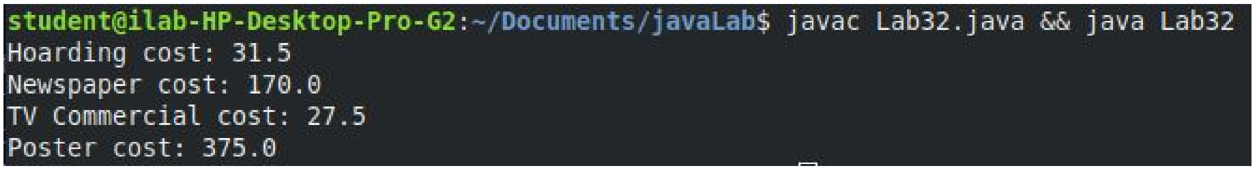
```
Shortest Arnold films:
Movie 2, English, Arnold

Rajnikanth/Sharukhan movies:
Movie 5, Tamil, Rajnikanth
Movie 6, Tamil, Rajnikanth
Movie 6, Tamil, Sharukhan

Comedy Movies:
Movie 4, English, Tom, Comedy
Movie 5, Tamil, Rajnikanth, Comedy
Movie 6, Tamil, Sharukhan, Comedy
student@ilab-HP-Desktop-Pro-G2:~/Documents/javaLab$ 
```

Question 2

Code

```
class Advertcampain {
 double cost(int param) {
   return 0;
 };
}

class Hoarding extends Advertcampain {
 double feesPerDay = 0.7;
 double primeLocation = 0.5;

 double cost(int days, Boolean prime) {
   if (prime)
     return days * feesPerDay + (days * feesPerDay) * primeLocation;

   return days * feesPerDay;
 }
}

class Poster extends Advertcampain {
 double feesPerCopy = 0.3;

 double cost(int copies, int dimensionsH, int dimensionsW) {
   return copies * feesPerCopy * (dimensionsH * dimensionsW) / 100;
 }
}

class Newspaper extends Advertcampain {
 double feesPerColumn = 1.7;

 double cost(int columns) {
   return columns * feesPerColumn;
 }
}

class TVCommercial extends Advertcampain {
 double feesPerSecond = 5.5;
 double peakTime = 2;

 double cost(int seconds, Boolean peak) {
   if (peak)
     return seconds * feesPerSecond * peakTime;
   return seconds * feesPerSecond;
 }
}

class Lab32 {
 public static void main(String[] args) {
   Hoarding campaign1 = new Hoarding();
   Newspaper campaign2 = new Newspaper();
   TVCommercial campaign3 = new TVCommercial();
   Poster campaign4 = new Poster();

   System.out.println("Hoarding cost: " + campaign1.cost(30, true));
   System.out.println("Newspaper cost: " + campaign2.cost(100));
   System.out.println("TV Commercial cost: " + campaign3.cost(5, false));
   System.out.println("Poster cost: " + campaign4.cost(25, 100, 50));
 }
}
```

Output



```
student@ilab-HP-Desktop-Pro-G2:~/Documents/javaLab$ javac Lab32.java && java Lab32
Hoarding cost: 31.5
Newspaper cost: 170.0
TV Commercial cost: 27.5
Poster cost: 375.0
```

Question 3

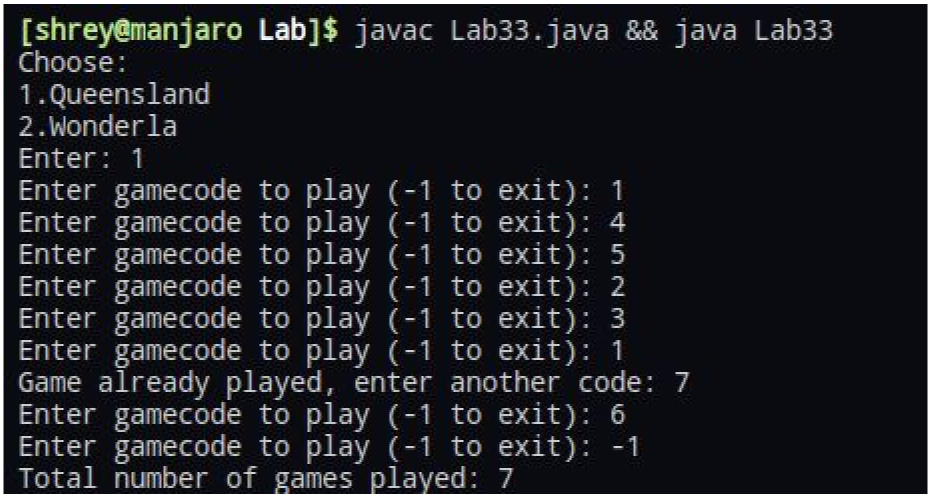
Code

```
import java.io.*;
import java.util.Scanner;
abstract class Themepark {
 int adultFees = 500;
 int childFees = 300;

 float totalCost(int adult, int children) {
   return adultFees * adult + childFees * children;
 }

 abstract void playGame(int gameCode);
}

class Queensland extends Themepark {
 Boolean[] games = new Boolean[30];
 public int playCount;

 Queensland() {
   for (int i = 0; i < 30; i++)
     games[i] = false;
   playCount = 0;
 }

 @Override
 void playGame(int gameCode) {
   if (games[gameCode] == true) {
     Scanner s = new Scanner(System.in);
     System.out.print("Game already played, enter another code: ");
     gameCode = s.nextInt();

     if (games[gameCode] == true)
       return;
   }
   games[gameCode] = true;
   playCount++;
 }
}

class Wonderla extends Themepark {
 Integer[] games = new Integer[40];

 Wonderla() {
   for (int i = 0; i < 40; i++)
     games[i] = 0;
 }

 @Override
 void playGame(int gameCode) {
   games[gameCode]++;
 }

 int getGamesRepeated() {
   int count = 0;
   for (int i = 0; i < games.length; i++){
     if (games[i] > 1)
       count++;
   }
   return count;
 }

 int getGamesNotPlayed() {
   int count = 0;
   for (int i = 0; i < games.length; i++)
     if (games[i] == 0)
       count++;
   return count;
 }
}
```

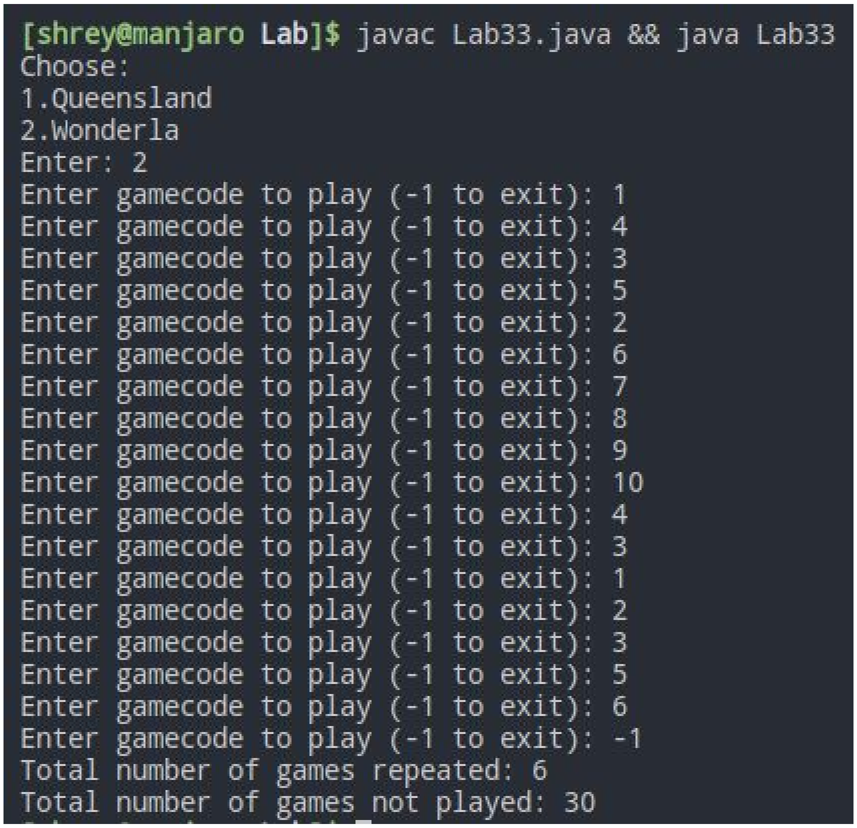
```
class Lab33 {
 public static void main(String[] args) {
   int choice, gameCode;
   System.out.print("Choose:\n1.Queensland\n2.Wonderla\nEnter: ");

   Scanner s = new Scanner(System.in);
   choice = s.nextInt();

   if (choice == 1) {
     Queensland park = new Queensland();
     do {
       System.out.print("Enter gamecode to play (-1 to exit): ");
       gameCode = s.nextInt();
       if (gameCode == -1) {
         System.out.println("Total number of games played: " +
park.playCount);
         return;
       }
       park.playGame(gameCode);
     } while (gameCode != -1);
   } else {
     Wonderla park = new Wonderla();
     do {
       System.out.print("Enter gamecode to play (-1 to exit): ");
       gameCode = s.nextInt();

       if (gameCode == -1) {
         System.out.println("Total number of games repeated: " +
park.getGamesRepeated());
         System.out.println("Total number of games not played: " +
park.getGamesNotPlayed());
         return;
       }
       park.playGame(gameCode);
     } while (gameCode != -1);

   }
```

```
  }
}
```

Output



```
[shrey@manjaro Lab]$ javac Lab33.java && java Lab33
Choose:
1.Queensland
2.Wonderla
Enter: 1
Enter gamecode to play (-1 to exit): 1
Enter gamecode to play (-1 to exit): 4
Enter gamecode to play (-1 to exit): 5
Enter gamecode to play (-1 to exit): 2
Enter gamecode to play (-1 to exit): 3
Enter gamecode to play (-1 to exit): 1
Game already played, enter another code: 7
Enter gamecode to play (-1 to exit): 6
Enter gamecode to play (-1 to exit): -1
Total number of games played: 7
```



```
[shrey@manjaro Lab]$ javac Lab33.java && java Lab33
Choose:
1.Queensland
2.Wonderla
Enter: 2
Enter gamecode to play (-1 to exit): 1
Enter gamecode to play (-1 to exit): 4
Enter gamecode to play (-1 to exit): 3
Enter gamecode to play (-1 to exit): 5
Enter gamecode to play (-1 to exit): 2
Enter gamecode to play (-1 to exit): 6
Enter gamecode to play (-1 to exit): 7
Enter gamecode to play (-1 to exit): 8
Enter gamecode to play (-1 to exit): 9
Enter gamecode to play (-1 to exit): 10
Enter gamecode to play (-1 to exit): 4
Enter gamecode to play (-1 to exit): 3
Enter gamecode to play (-1 to exit): 1
Enter gamecode to play (-1 to exit): 2
Enter gamecode to play (-1 to exit): 3
Enter gamecode to play (-1 to exit): 5
Enter gamecode to play (-1 to exit): 6
Enter gamecode to play (-1 to exit): -1
Total number of games repeated: 6
Total number of games not played: 30
```

Question 4

Code

```
interface Shape3D {
 double getVolume();
}

class Cubiod implements Shape3D {
 private double breadth, length, height;

 public double getVolume() {
   return breadth * length * height;
 }

 Cubiod(double breadth, double length, double height) {
   this.breadth = breadth;
   this.length = length;
   this.height = height;
 }
}

interface Solid3D extends Shape3D {
 double getMass();
 double getDensity();
}

class SolidCubiod extends Cubiod implements Solid3D {
 private double density;

 public double getMass(){
   return getVolume() * density;
 }

 public double getDensity(){
   return density;
 }

 SolidCubiod(double breadth, double length, double height){
   super(breadth, length, height);
   this.density=1;
 }

 SolidCubiod(double breadth, double length, double height, double
density){
   super(breadth, length, height);
   this.density=density;
 }
}
```