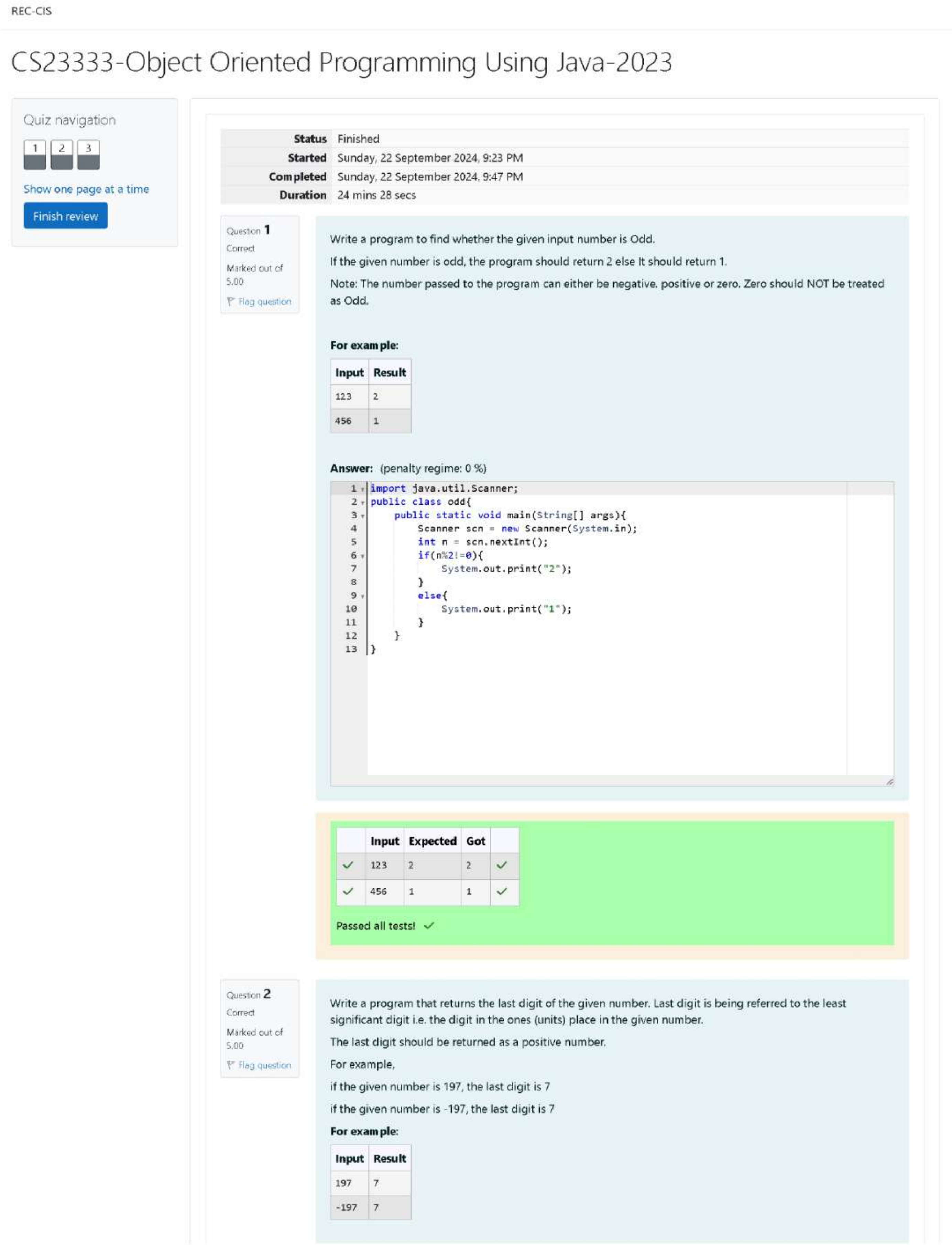


REC-CIS

CS23333-Object Oriented Programming Using Java-2023

Status	Finished
Started	Sunday, 22 September 2024, 9:23 PM
Completed	Sunday, 22 September 2024, 9:47 PM
Duration	24 mins 28 secs

Question **1**

Correct

Marked out of 5.00

Write a program to find whether the given input number is Odd.

If the given number is odd, the program should return 2 else It should return 1.

Note: The number passed to the program can either be negative. positive or zero. Zero should NOT be treated as Odd.

For example:

Input	Result
123	2
456	1

Answer: (penalty regime: 0 %)

```java
import java.util.Scanner;
public class odd{
    public static void main(String[] args){
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        if(n%2!=0){
            System.out.print("2");
        }
        else{
            System.out.print("1");
        }
    }
}
```

	Input	Expected	Got	
✓	123	2	2	✓
✓	456	1	1	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Write a program that returns the last digit of the given number. Last digit is being referred to the least significant digit i.e. the digit in the ones (units) place in the given number.

The last digit should be returned as a positive number.

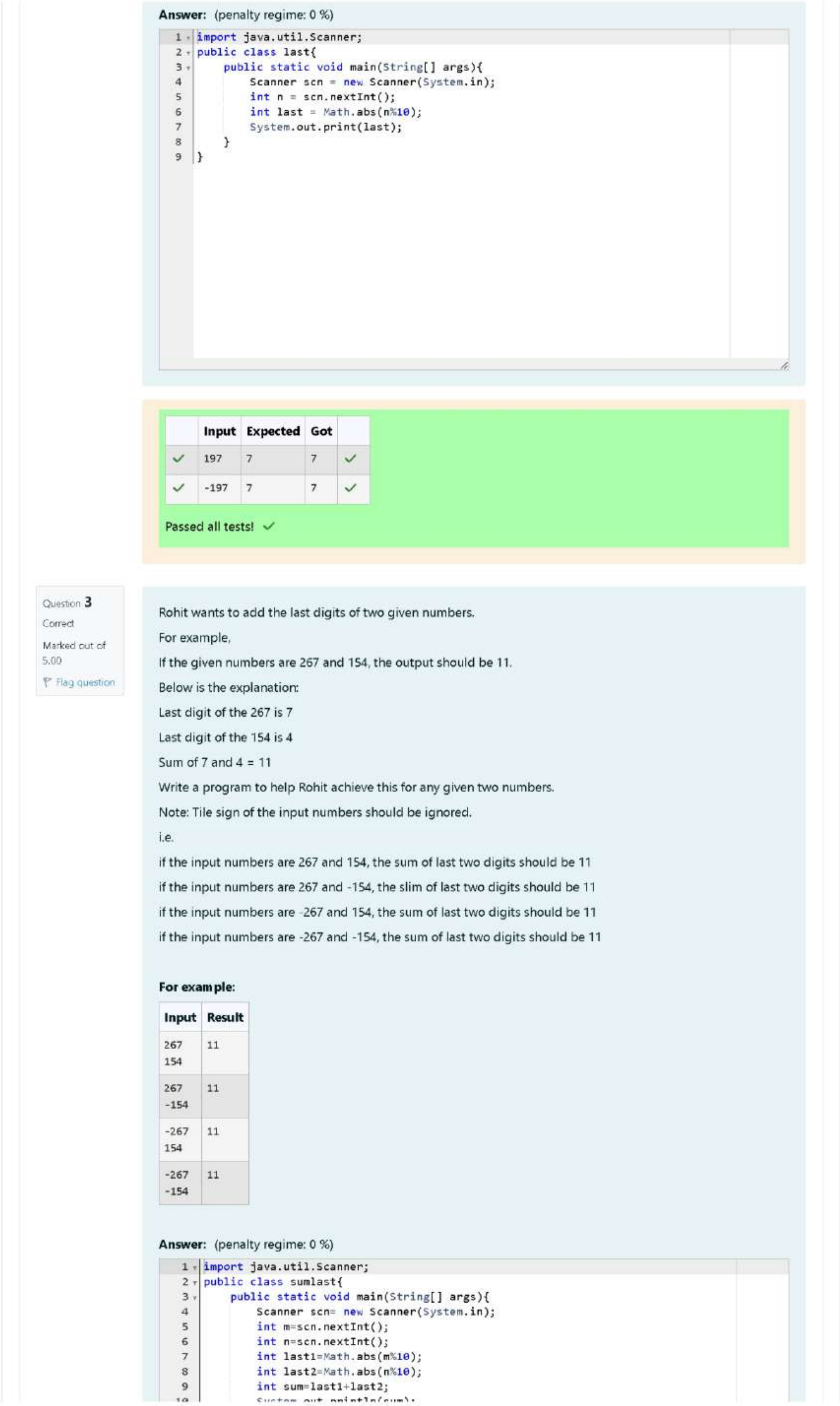
For example,

if the given number is 197, the last digit is 7

if the given number is -197, the last digit is 7

For example:

Input	Result
197	7
-197	7



Answer: (penalty regime: 0 %)

```java
import java.util.Scanner;
public class last{
    public static void main(String[] args){
        Scanner scn = new Scanner(System.in);
        int n = scn.nextInt();
        int last = Math.abs(n%10);
        System.out.print(last);
    }
}
```

	Input	Expected	Got	
✓	197	7	7	✓
✓	-197	7	7	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

Flag question

Rohit wants to add the last digits of two given numbers.

For example,

If the given numbers are 267 and 154, the output should be 11.

Below is the explanation:

Last digit of the 267 is 7

Last digit of the 154 is 4

Sum of 7 and 4 = 11

Write a program to help Rohit achieve this for any given two numbers.

Note: Tile sign of the input numbers should be ignored.

i.e.

if the input numbers are 267 and 154, the sum of last two digits should be 11

if the input numbers are 267 and -154, the slim of last two digits should be 11

if the input numbers are -267 and 154, the sum of last two digits should be 11

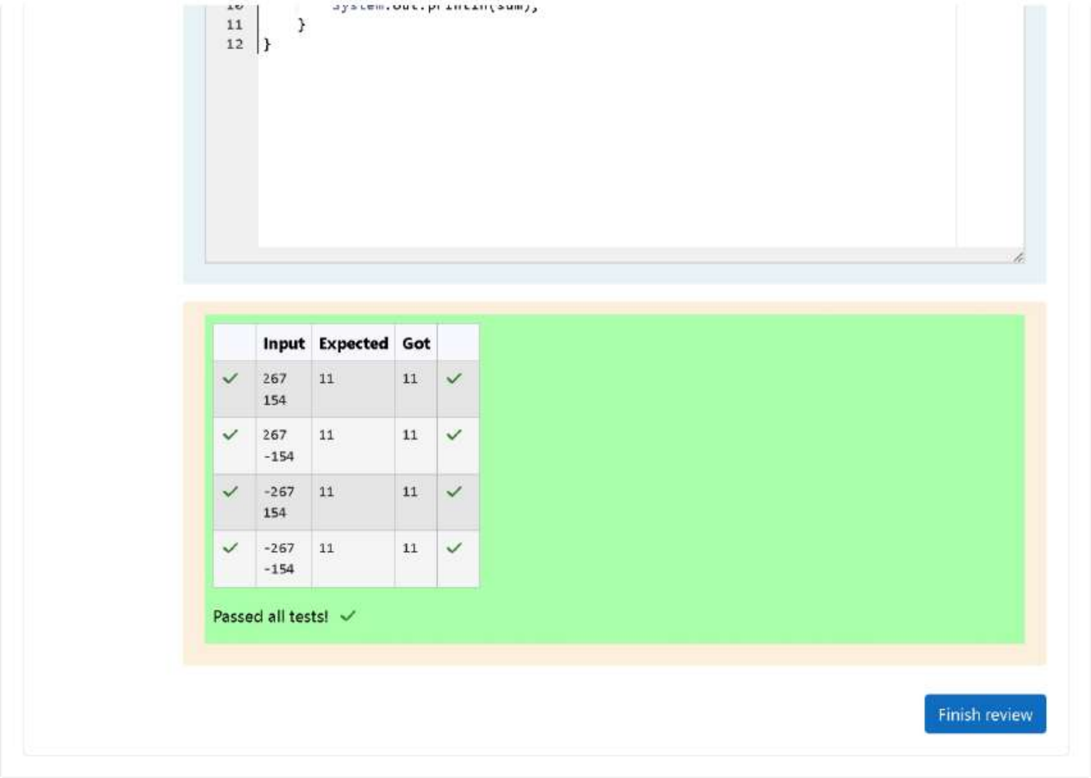
if the input numbers are -267 and -154, the sum of last two digits should be 11

For example:

Input	Result
267 154	11
267 -154	11
-267 154	11
-267 -154	11

Answer: (penalty regime: 0 %)

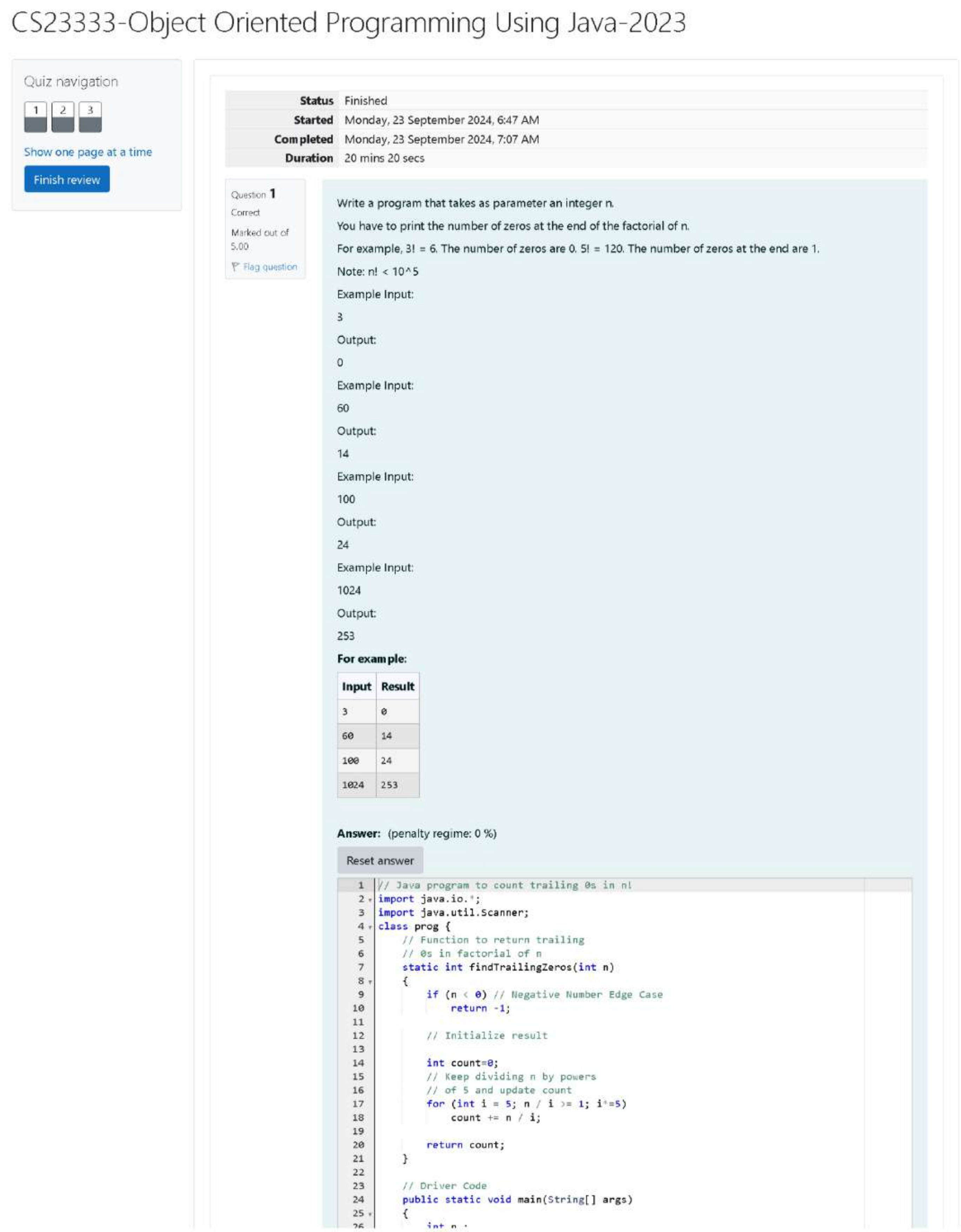
```java
import java.util.Scanner;
public class sumlast{
    public static void main(String[] args){
        Scanner scn= new Scanner(System.in);
        int m=scn.nextInt();
        int n=scn.nextInt();
        int last1=Math.abs(m%10);
        int last2=Math.abs(n%10);
        int sum=last1+last2;
        System.out.println(sum);
```



```
            System.out.println(sum);
        }
}
```

	Input	Expected	Got	
✓	267 154	11	11	✓
✓	267 -154	11	11	✓
✓	-267 154	11	11	✓
✓	-267 -154	11	11	✓

Passed all tests! ✓



CS23333-Object Oriented Programming Using Java-2023

Status	Finished
Started	Monday, 23 September 2024, 6:47 AM
Completed	Monday, 23 September 2024, 7:07 AM
Duration	20 mins 20 secs

Question **1**
Correct
Marked out of 5.00

Write a program that takes as parameter an integer n.

You have to print the number of zeros at the end of the factorial of n.

For example, 3! = 6. The number of zeros are 0. 5! = 120. The number of zeros at the end are 1.

Note: n! < 10^5

Example Input:

3

Output:

0

Example Input:

60

Output:

14

Example Input:

100

Output:

24

Example Input:

1024

Output:

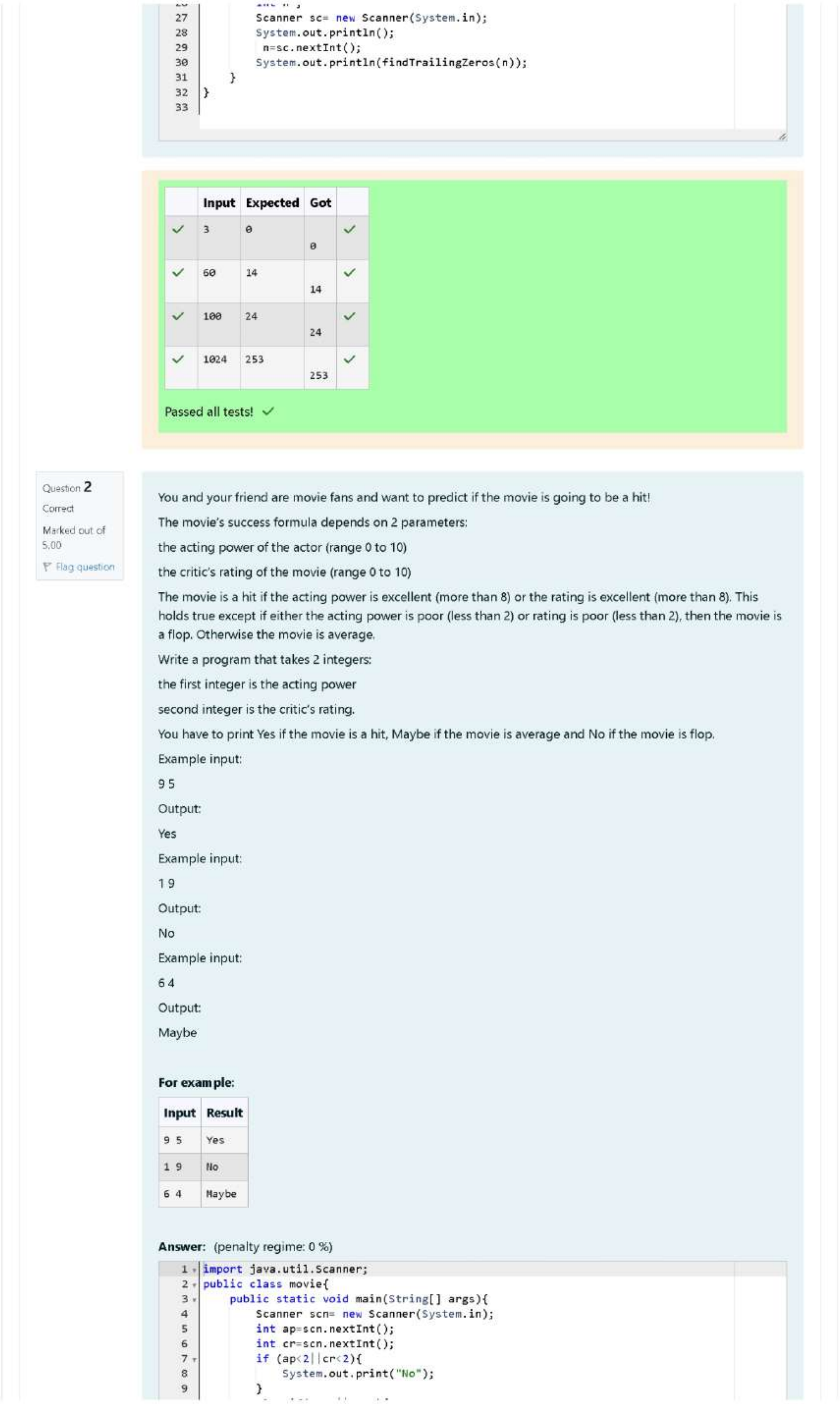
253

For example:

Input	Result
3	0
60	14
100	24
1024	253

Answer: (penalty regime: 0 %)

```
// Java program to count trailing 0s in n!
import java.io.*;
import java.util.Scanner;
class prog {
    // Function to return trailing
    // 0s in factorial of n
    static int findTrailingZeros(int n)
    {
        if (n < 0) // Negative Number Edge Case
            return -1;

        // Initialize result

        int count=0;
        // Keep dividing n by powers
        // of 5 and update count
        for (int i = 5; n / i >= 1; i*=5)
            count += n / i;

        return count;
    }

    // Driver Code
    public static void main(String[] args)
    {
        int n ;
```



```
        Scanner sc= new Scanner(System.in);
        System.out.println();
         n=sc.nextInt();
        System.out.println(findTrailingZeros(n));
    }
}
```

	Input	Expected	Got	
✓	3	0	0	✓
✓	60	14	14	✓
✓	100	24	24	✓
✓	1024	253	253	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Flag question

You and your friend are movie fans and want to predict if the movie is going to be a hit!

The movie's success formula depends on 2 parameters:

the acting power of the actor (range 0 to 10)

the critic's rating of the movie (range 0 to 10)

The movie is a hit if the acting power is excellent (more than 8) or the rating is excellent (more than 8). This holds true except if either the acting power is poor (less than 2) or rating is poor (less than 2), then the movie is a flop. Otherwise the movie is average.

Write a program that takes 2 integers:

the first integer is the acting power

second integer is the critic's rating.

You have to print Yes if the movie is a hit, Maybe if the movie is average and No if the movie is flop.

Example input:

9 5

Output:

Yes

Example input:

1 9

Output:

No

Example input:

6 4

Output:

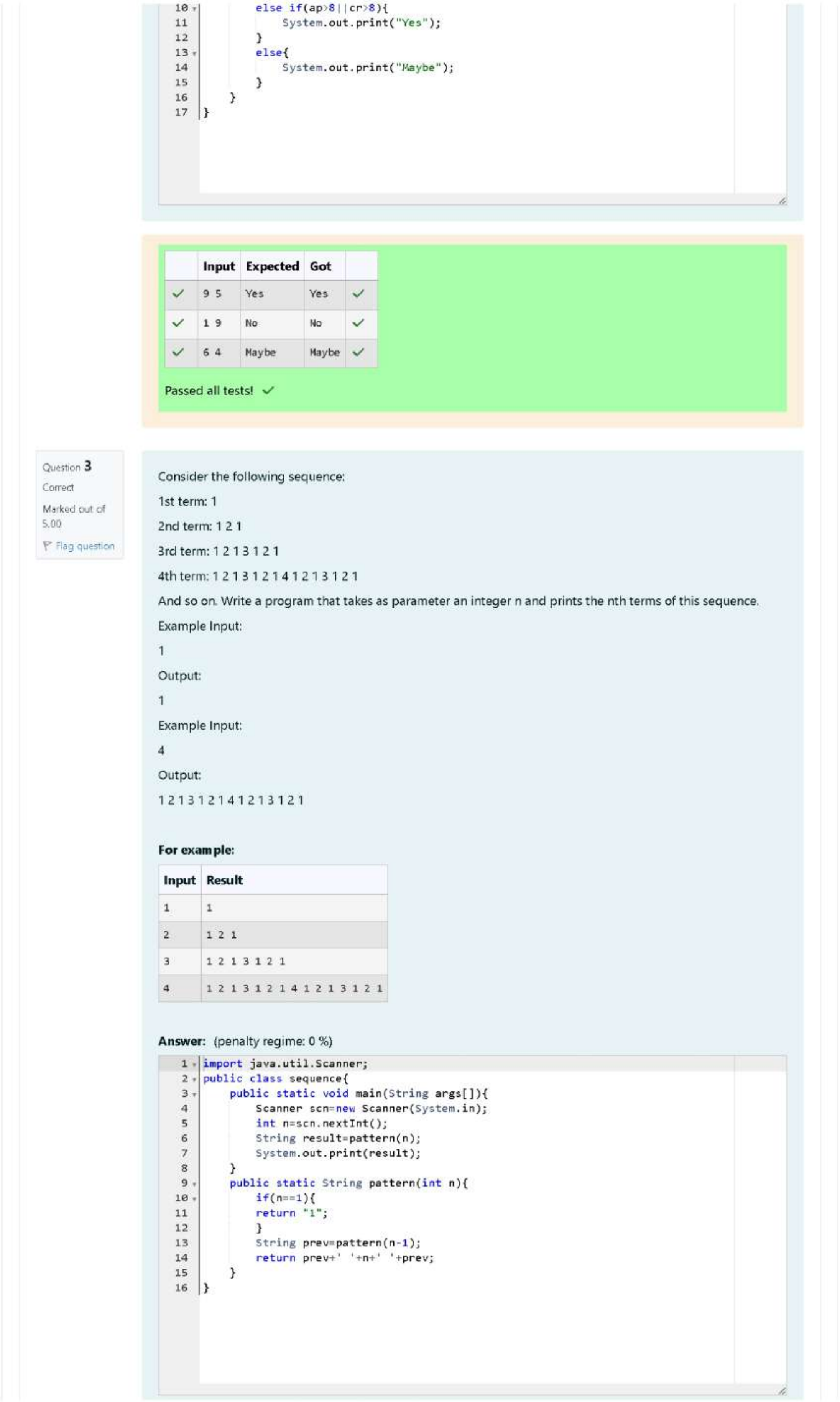
Maybe

For example:

Input	Result
9 5	Yes
1 9	No
6 4	Maybe

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class movie{
    public static void main(String[] args){
        Scanner scn= new Scanner(System.in);
        int ap=scn.nextInt();
        int cr=scn.nextInt();
        if (ap<2||cr<2){
            System.out.print("No");
        }
```



```
        else if(ap>8||cr>8){
            System.out.print("Yes");
        }
        else{
            System.out.print("Maybe");
        }
    }
}
```

	Input	Expected	Got	
✓	9 5	Yes	Yes	✓
✓	1 9	No	No	✓
✓	6 4	Maybe	Maybe	✓

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

Flag question

Consider the following sequence:

1st term: 1

2nd term: 1 2 1

3rd term: 1 2 1 3 1 2 1

4th term: 1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

And so on. Write a program that takes as parameter an integer n and prints the nth terms of this sequence.

Example Input:

1

Output:

1

Example Input:

4

Output:

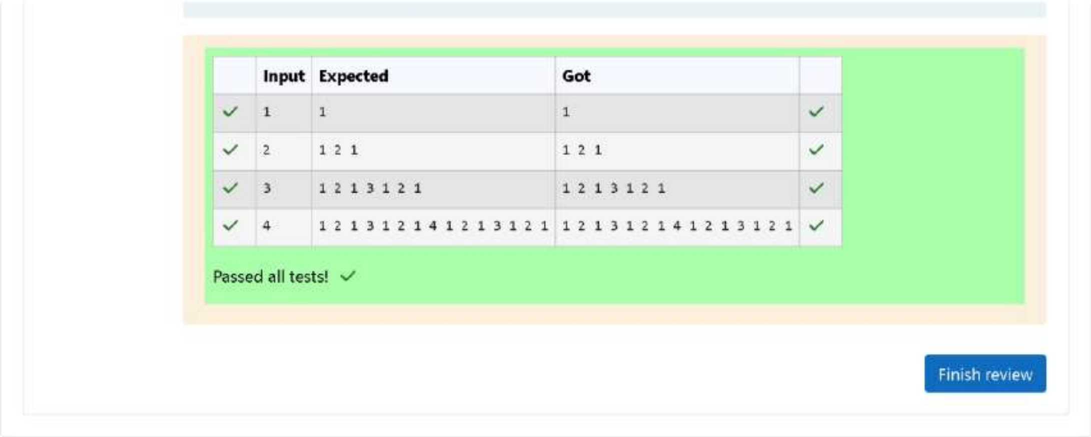
1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

For example:

Input	Result
1	1
2	1 2 1
3	1 2 1 3 1 2 1
4	1 2 1 3 1 2 1 4 1 2 1 3 1 2 1

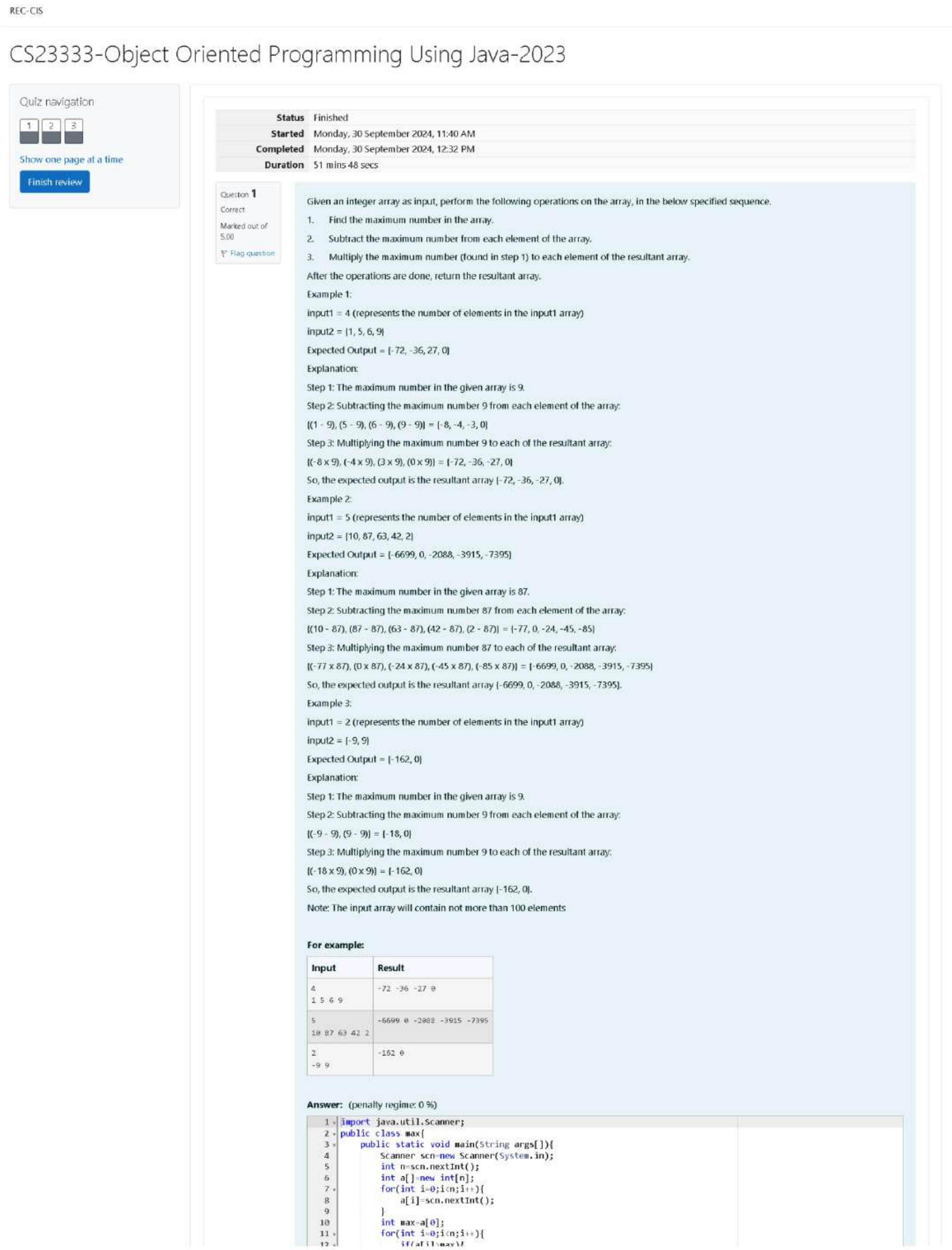
Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class sequence{
    public static void main(String args[]){
        Scanner scn=new Scanner(System.in);
        int n=scn.nextInt();
        String result=pattern(n);
        System.out.print(result);
    }
    public static String pattern(int n){
        if(n==1){
        return "1";
        }
        String prev=pattern(n-1);
        return prev+' '+n+' '+prev;
    }
}
```



	Input	Expected	Got	
✓	1	1	1	✓
✓	2	1 2 1	1 2 1	✓
✓	3	1 2 1 3 1 2 1	1 2 1 3 1 2 1	✓
✓	4	1 2 1 3 1 2 1 4 1 2 1 3 1 2 1	1 2 1 3 1 2 1 4 1 2 1 3 1 2 1	✓

Passed all tests! ✓



CS23333-Object Oriented Programming Using Java-2023

Status	Finished
Started	Monday, 30 September 2024, 11:40 AM
Completed	Monday, 30 September 2024, 12:32 PM
Duration	51 mins 48 secs

Given an integer array as input, perform the following operations on the array, in the below specified sequence.

1. Find the maximum number in the array.
2. Subtract the maximum number from each element of the array.
3. Multiply the maximum number (found in step 1) to each element of the resultant array.

After the operations are done, return the resultant array.

Example 1:

input1 = 4 (represents the number of elements in the input1 array)

input2 = [1, 5, 6, 9]

Expected Output = [-72, -36, 27, 0]

Explanation:

Step 1: The maximum number in the given array is 9.

Step 2: Subtracting the maximum number 9 from each element of the array:

[(1 - 9), (5 - 9), (6 - 9), (9 - 9)] = [-8, -4, -3, 0]

Step 3: Multiplying the maximum number 9 to each of the resultant array:

[(-8 x 9), (-4 x 9), (3 x 9), (0 x 9)] = [-72, -36, -27, 0]

So, the expected output is the resultant array [-72, -36, -27, 0].

Example 2:

input1 = 5 (represents the number of elements in the input1 array)

input2 = [10, 87, 63, 42, 2]

Expected Output = [-6699, 0, -2088, -3915, -7395]

Explanation:

Step 1: The maximum number in the given array is 87.

Step 2: Subtracting the maximum number 87 from each element of the array:

[(10 - 87), (87 - 87), (63 - 87), (42 - 87), (2 - 87)] = [-77, 0, -24, -45, -85]

Step 3: Multiplying the maximum number 87 to each of the resultant array:

[(-77 x 87), (0 x 87), (-24 x 87), (-45 x 87), (-85 x 87)] = [-6699, 0, -2088, -3915, -7395]

So, the expected output is the resultant array [-6699, 0, -2088, -3915, -7395].

Example 3:

input1 = 2 (represents the number of elements in the input1 array)

input2 = [-9, 9]

Expected Output = [-162, 0]

Explanation:

Step 1: The maximum number in the given array is 9.

Step 2: Subtracting the maximum number 9 from each element of the array:

[(-9 - 9), (9 - 9)] = [-18, 0]

Step 3: Multiplying the maximum number 9 to each of the resultant array:

[(-18 x 9), (0 x 9)] = [-162, 0]

So, the expected output is the resultant array [-162, 0].

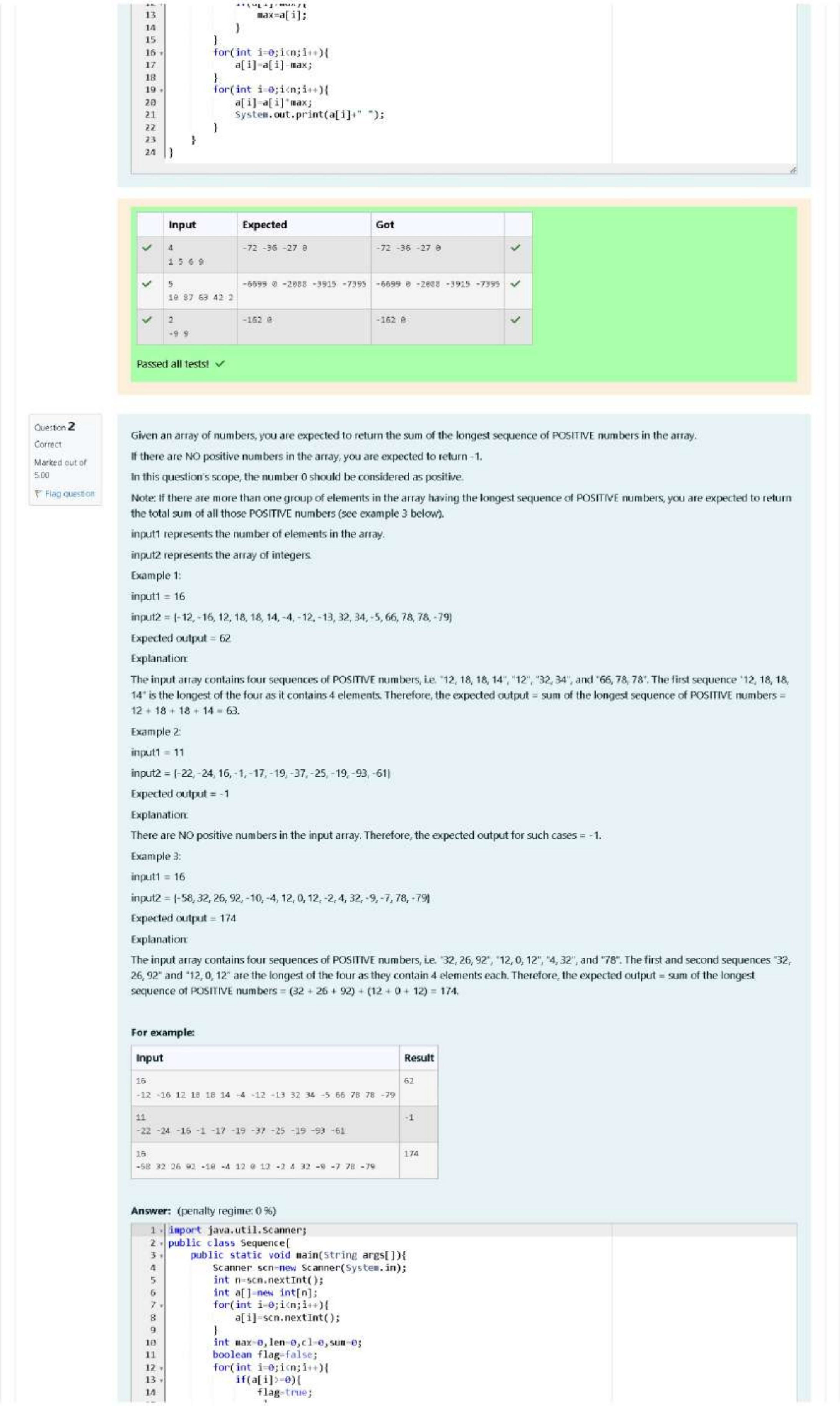
Note: The input array will contain not more than 100 elements

For example:

Input	Result
4 1 5 6 9	-72 -36 -27 0
5 10 87 63 42 2	-6699 0 -2088 -3915 -7395
2 -9 9	-162 0

Answer: (penalty regime: 0 %)

```java
import java.util.Scanner;
public class max{
    public static void main(String args[]){
        Scanner scn=new Scanner(System.in);
        int n=scn.nextInt();
        int a[]=new int[n];
        for(int i=0;i<n;i++){
            a[i]=scn.nextInt();
        }
        int max=a[0];
        for(int i=0;i<n;i++){
            if(a[i]>max){
```



```
                if(a[i]>max){
                    max=a[i];
                }
            }
            for(int i=0;i<n;i++){
                a[i]=a[i]-max;
            }
            for(int i=0;i<n;i++){
                a[i]=a[i]*max;
                System.out.print(a[i]+" ");
            }
        }
}
```

	Input	Expected	Got	
✓	4 1 5 6 9	-72 -36 -27 0	-72 -36 -27 0	✓
✓	5 10 87 63 42 2	-6699 0 -2088 -3915 -7395	-6699 0 -2088 -3915 -7395	✓
✓	2 -9 9	-162 0	-162 0	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Flag question

Given an array of numbers, you are expected to return the sum of the longest sequence of POSITIVE numbers in the array.

If there are NO positive numbers in the array, you are expected to return -1.

In this question's scope, the number 0 should be considered as positive.

Note: If there are more than one group of elements in the array having the longest sequence of POSITIVE numbers, you are expected to return the total sum of all those POSITIVE numbers (see example 3 below).

input1 represents the number of elements in the array.

input2 represents the array of integers.

Example 1:

input1 = 16

input2 = {-12, -16, 12, 18, 18, 14, -4, -12, -13, 32, 34, -5, 66, 78, 78, -79}

Expected output = 62

Explanation:

The input array contains four sequences of POSITIVE numbers, i.e. "12, 18, 18, 14", "12", "32, 34", and "66, 78, 78". The first sequence "12, 18, 18, 14" is the longest of the four as it contains 4 elements. Therefore, the expected output = sum of the longest sequence of POSITIVE numbers = 12 + 18 + 18 + 14 = 63.

Example 2:

input1 = 11

input2 = {-22, -24, 16, -1, -17, -19, -37, -25, -19, -93, -61}

Expected output = -1

Explanation:

There are NO positive numbers in the input array. Therefore, the expected output for such cases = -1.

Example 3:

input1 = 16

input2 = {-58, 32, 26, 92, -10, -4, 12, 0, 12, -2, 4, 32, -9, -7, 78, -79}

Expected output = 174

Explanation:

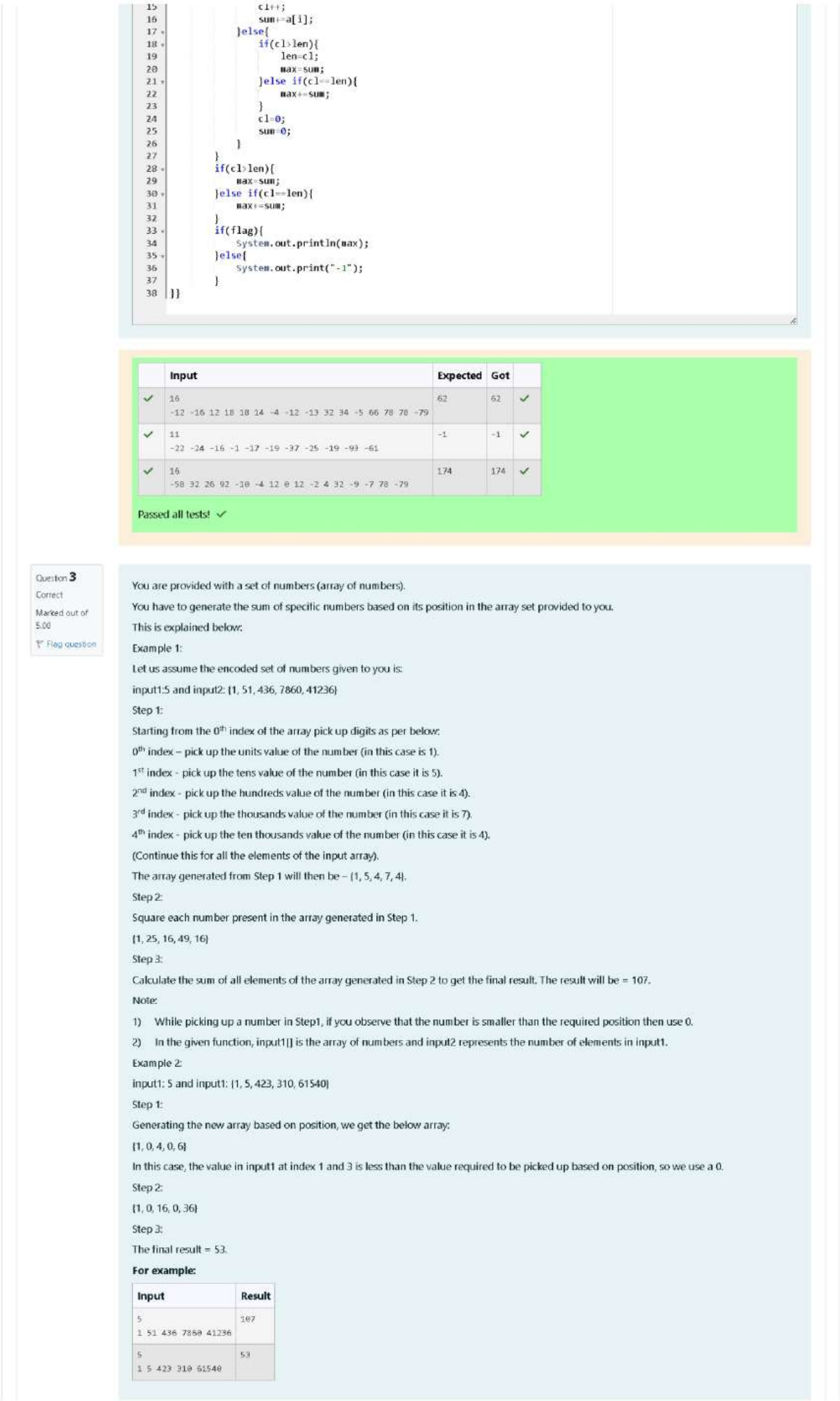
The input array contains four sequences of POSITIVE numbers, i.e. "32, 26, 92", "12, 0, 12", "4, 32", and "78". The first and second sequences "32, 26, 92" and "12, 0, 12" are the longest of the four as they contain 4 elements each. Therefore, the expected output = sum of the longest sequence of POSITIVE numbers = (32 + 26 + 92) + (12 + 0 + 12) = 174.

For example:

Input	Result
16 -12 -16 12 18 18 14 -4 -12 -13 32 34 -5 66 78 78 -79	62
11 -22 -24 -16 -1 -17 -19 -37 -25 -19 -93 -61	-1
16 -58 32 26 92 -10 -4 12 0 12 -2 4 32 -9 -7 78 -79	174

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class Sequence{
    public static void main(String args[]){
        Scanner scn=new Scanner(System.in);
        int n=scn.nextInt();
        int a[]=new int[n];
        for(int i=0;i<n;i++){
            a[i]=scn.nextInt();
        }
        int max=0,len=0,cl=0,sum=0;
        boolean flag=false;
        for(int i=0;i<n;i++){
            if(a[i]>=0){
                flag=true;
```



```
                cl++;
                sum+=a[i];
            }else{
                if(cl>len){
                    len=cl;
                    max=sum;
                }else if(cl==len){
                    max+=sum;
                }
                cl=0;
                sum=0;
            }
        }
        if(cl>len){
            max=sum;
        }else if(cl==len){
            max+=sum;
        }
        if(flag){
            System.out.println(max);
        }else{
            System.out.print("-1");
        }
}}
```

	Input	Expected	Got	
✓	16 -12 -16 12 18 18 14 -4 -12 -13 32 34 -5 66 78 78 -79	62	62	✓
✓	11 -22 -24 -16 -1 -17 -19 -37 -25 -19 -93 -61	-1	-1	✓
✓	16 -58 32 26 92 -10 -4 12 0 12 -2 4 32 -9 -7 78 -79	174	174	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

Flag question

You are provided with a set of numbers (array of numbers).

You have to generate the sum of specific numbers based on its position in the array set provided to you.

This is explained below:

Example 1:

Let us assume the encoded set of numbers given to you is:

input1:5 and input2: {1, 51, 436, 7860, 41236}

Step 1:

Starting from the 0th index of the array pick up digits as per below:

0th index – pick up the units value of the number (in this case is 1).

1st index - pick up the tens value of the number (in this case it is 5).

2nd index - pick up the hundreds value of the number (in this case it is 4).

3rd index - pick up the thousands value of the number (in this case it is 7).

4th index - pick up the ten thousands value of the number (in this case it is 4).

(Continue this for all the elements of the input array).

The array generated from Step 1 will then be – {1, 5, 4, 7, 4}.

Step 2:

Square each number present in the array generated in Step 1.

{1, 25, 16, 49, 16}

Step 3:

Calculate the sum of all elements of the array generated in Step 2 to get the final result. The result will be = 107.

Note:

1) While picking up a number in Step1, if you observe that the number is smaller than the required position then use 0.

2) In the given function, input1[] is the array of numbers and input2 represents the number of elements in input1.

Example 2:

input1: 5 and input1: {1, 5, 423, 310, 61540}

Step 1:

Generating the new array based on position, we get the below array:

{1, 0, 4, 0, 6}

In this case, the value in input1 at index 1 and 3 is less than the value required to be picked up based on position, so we use a 0.

Step 2:

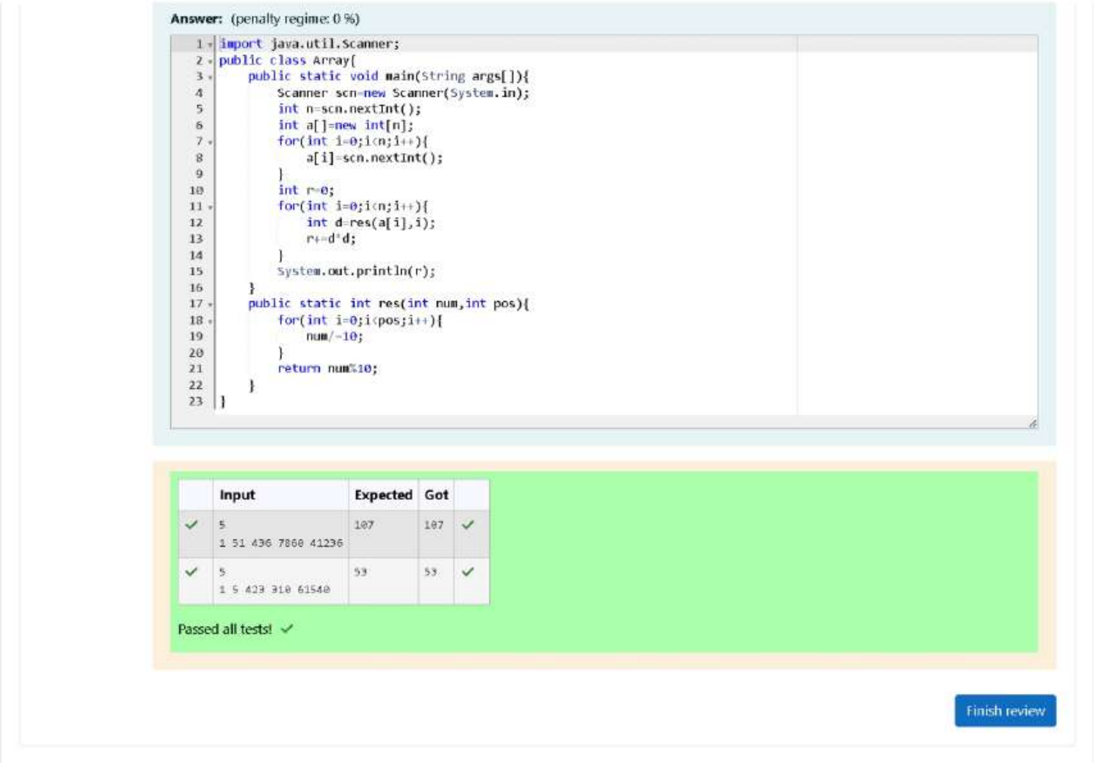
{1, 0, 16, 0, 36}

Step 3:

The final result = 53.

For example:

Input	Result
5 1 51 436 7860 41236	107
5 1 5 423 310 61540	53

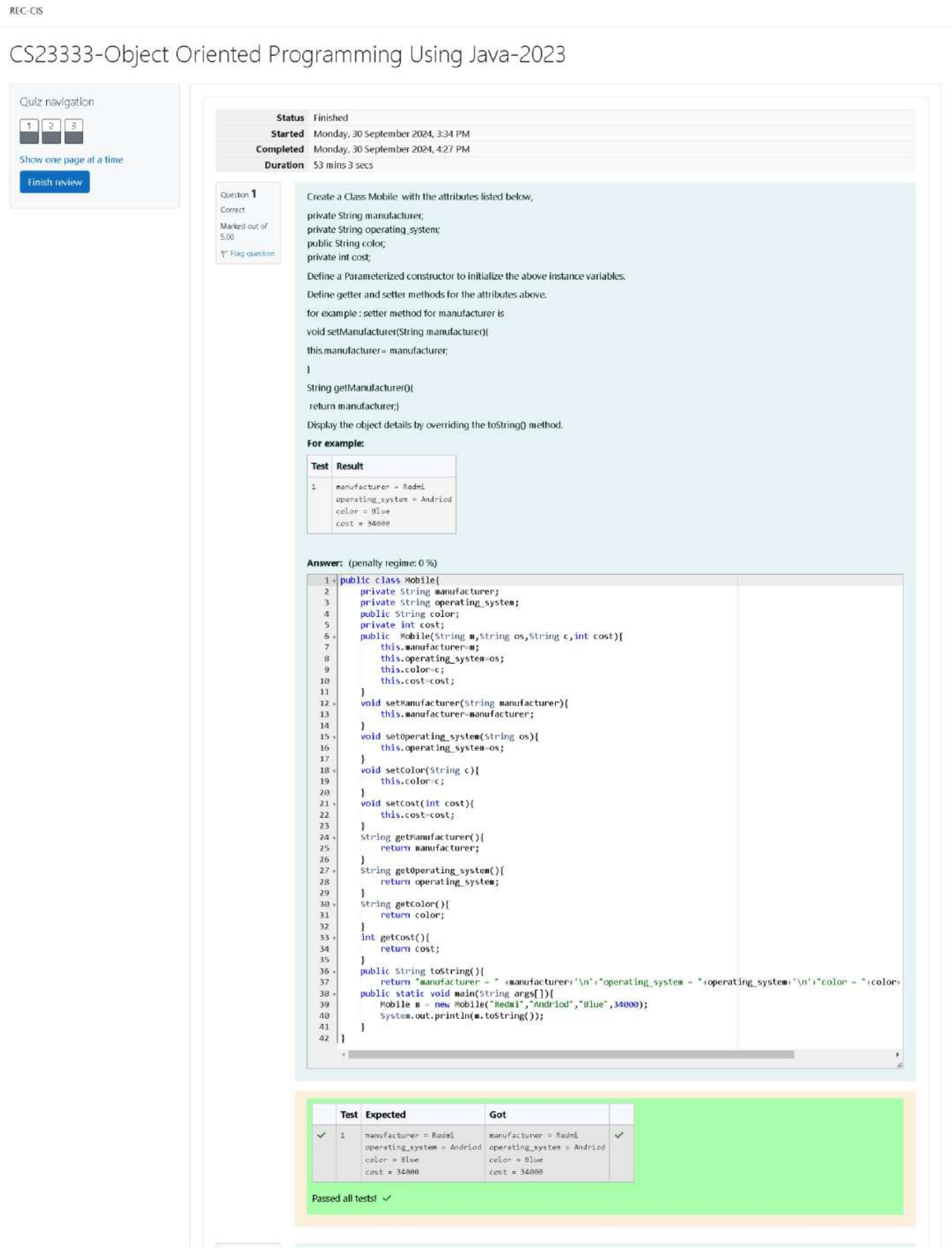


Answer: (penalty regime: 0 %)

```java
import java.util.Scanner;
public class Array{
    public static void main(String args[]){
        Scanner scn=new Scanner(System.in);
        int n=scn.nextInt();
        int a[]=new int[n];
        for(int i=0;i<n;i++){
            a[i]=scn.nextInt();
        }
        int r=0;
        for(int i=0;i<n;i++){
            int d=res(a[i],i);
            r+=d*d;
        }
        System.out.println(r);
    }
    public static int res(int num,int pos){
        for(int i=0;i<pos;i++){
            num/=10;
        }
        return num%10;
    }
}
```

	Input	Expected	Got	
✓	5 1 51 436 7868 41236	107	107	✓
✓	5 1 5 423 310 61540	53	53	✓

Passed all tests! ✓



REC-CIS

CS23333-Object Oriented Programming Using Java-2023

Status	Finished
Started	Monday, 30 September 2024, 3:34 PM
Completed	Monday, 30 September 2024, 4:27 PM
Duration	53 mins 3 secs

Question **1**

Correct

Marked out of 5.00

Create a Class Mobile with the attributes listed below,

private String manufacturer;
private String operating_system;
public String color;
private int cost;

Define a Parameterized constructor to initialize the above instance variables.

Define getter and setter methods for the attributes above.

for example : setter method for manufacturer is

void setManufacturer(String manufacturer){

this.manufacturer= manufacturer;

}

String getManufacturer(){

return manufacturer;}

Display the object details by overriding the toString() method.

For example:

Test	Result

1	manufacturer = Redmi
	operating_system = Andriod
	color = Blue
	cost = 34000

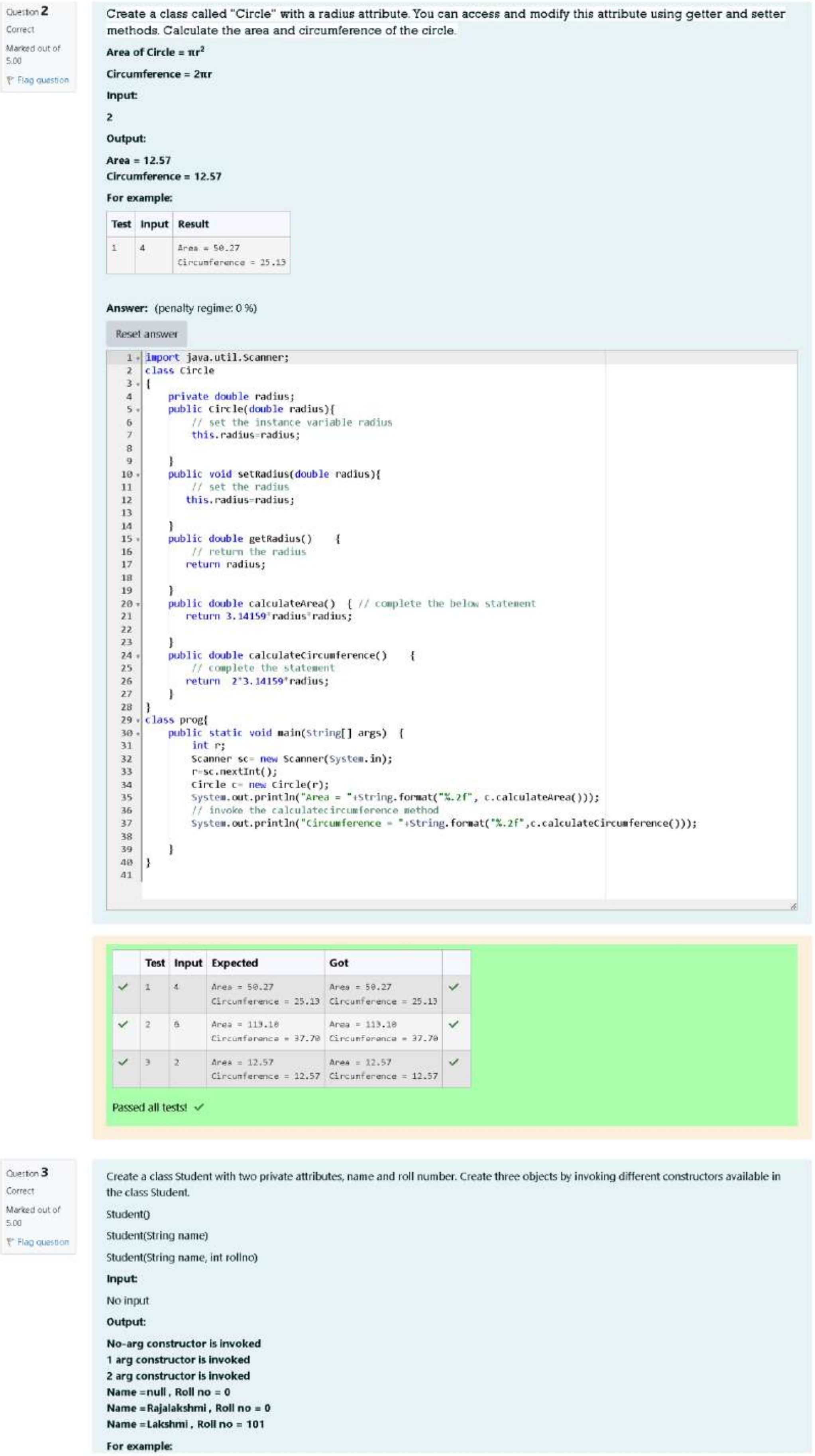
Answer: (penalty regime: 0 %)

```java
public class Mobile{
    private String manufacturer;
    private String operating_system;
    public String color;
    private int cost;
    public  Mobile(String m,String os,String c,int cost){
        this.manufacturer=m;
        this.operating_system=os;
        this.color=c;
        this.cost=cost;
    }
    void setManufacturer(String manufacturer){
        this.manufacturer=manufacturer;
    }
    void setOperating_system(String os){
        this.operating_system=os;
    }
    void setColor(String c){
        this.color=c;
    }
    void setCost(int cost){
        this.cost=cost;
    }
    String getManufacturer(){
        return manufacturer;
    }
    String getOperating_system(){
        return operating_system;
    }
    String getColor(){
        return color;
    }
    int getCost(){
        return cost;
    }
    public String toString(){
        return "manufacturer = " +manufacturer+'\n'+"operating_system = "+operating_system+'\n'+"color = "+color+
    public static void main(String args[]){
        Mobile m = new Mobile("Redmi","Andriod","Blue",34000);
        System.out.println(m.toString());
    }
}
```

	Test	Expected	Got	

✓	1	manufacturer = Redmi
		operating_system = Andriod
		color = Blue
		cost = 34000
			manufacturer = Redmi
			operating_system = Andriod
			color = Blue
			cost = 34000
				✓

Passed all tests! ✓



Question **2**

Correct

Marked out of 5.00

Flag question

Create a class called "Circle" with a radius attribute. You can access and modify this attribute using getter and setter methods. Calculate the area and circumference of the circle.

Area of Circle = πr^2

Circumference = $2\pi r$

Input:

2

Output:

Area = 12.57
Circumference = 12.57

For example:

Test	Input	Result

1	4	Area = 50.27
		Circumference = 25.13

Answer: (penalty regime: 0 %)

Reset answer

```
import java.util.Scanner;
class Circle
{
    private double radius;
    public Circle(double radius){
        // set the instance variable radius
        this.radius=radius;

    }
    public void setRadius(double radius){
        // set the radius
       this.radius=radius;

    }
    public double getRadius()    {
        // return the radius
       return radius;

    }
    public double calculateArea()  { // complete the below statement
       return 3.14159*radius*radius;

    }
    public double calculateCircumference()    {
        // complete the statement
       return  2*3.14159*radius;
    }
}
class prog{
    public static void main(String[] args)  {
        int r;
        Scanner sc= new Scanner(System.in);
        r=sc.nextInt();
        Circle c= new Circle(r);
        System.out.println("Area = "+String.format("%.2f", c.calculateArea()));
        // invoke the calculatecircumference method
        System.out.println("Circumference = "+String.format("%.2f",c.calculateCircumference()));

    }
}
```

	Test	Input	Expected	Got	
✓	1	4	Area = 50.27 Circumference = 25.13	Area = 50.27 Circumference = 25.13	✓
✓	2	6	Area = 113.10 Circumference = 37.70	Area = 113.10 Circumference = 37.70	✓
✓	3	2	Area = 12.57 Circumference = 12.57	Area = 12.57 Circumference = 12.57	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

Flag question

Create a class Student with two private attributes, name and roll number. Create three objects by invoking different constructors available in the class Student.

Student()

Student(String name)

Student(String name, int rollno)

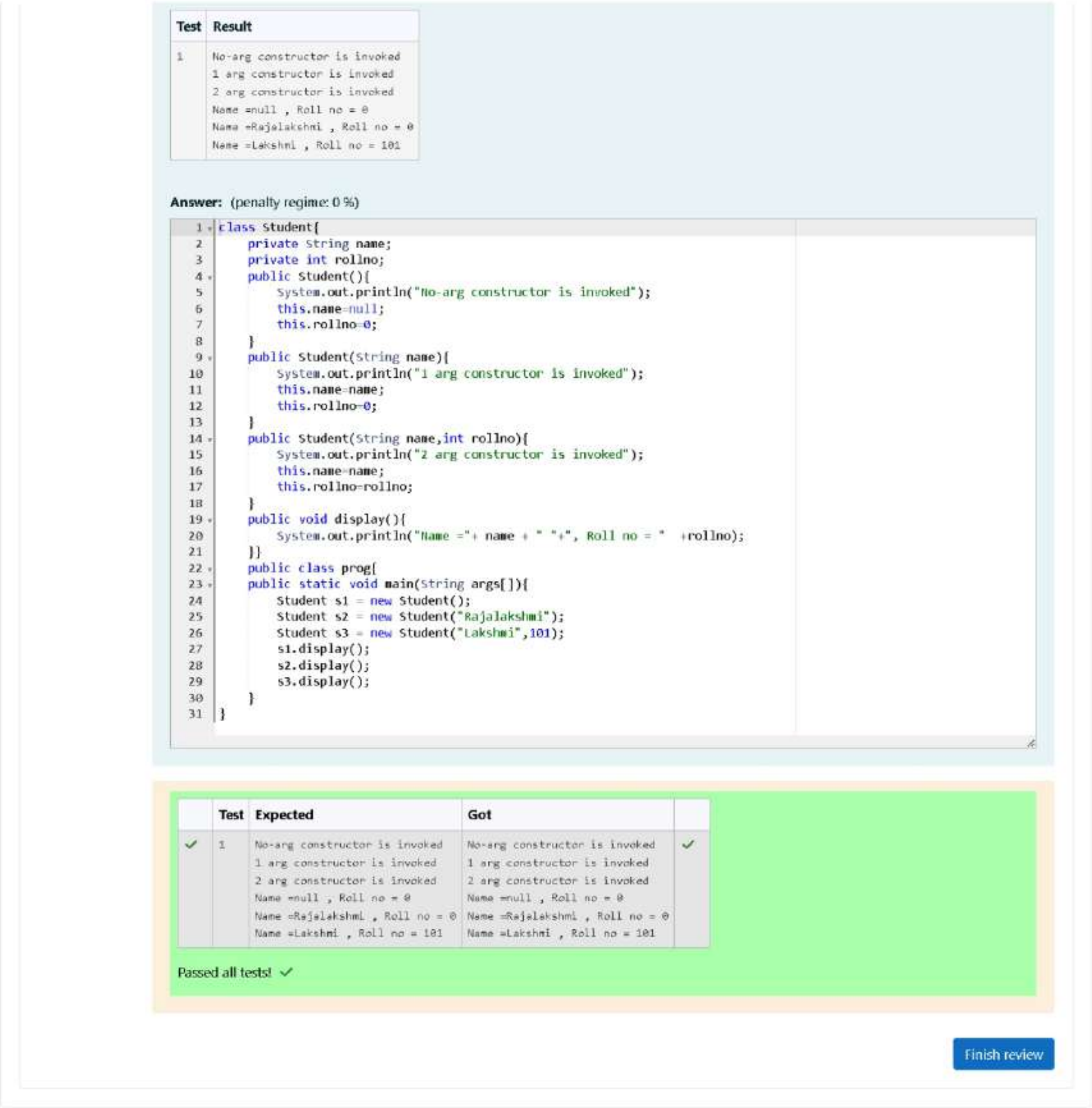
Input:

No input

Output:

No-arg constructor is invoked
1 arg constructor is invoked
2 arg constructor is invoked
Name =null , Roll no = 0
Name =Rajalakshmi , Roll no = 0
Name =Lakshmi , Roll no = 101

For example:



Test	Result

1	No-arg constructor is invoked
	1 arg constructor is invoked
	2 arg constructor is invoked
	Name =null , Roll no = 0
	Name =Rajalakshmi , Roll no = 0
	Name =Lakshmi , Roll no = 101

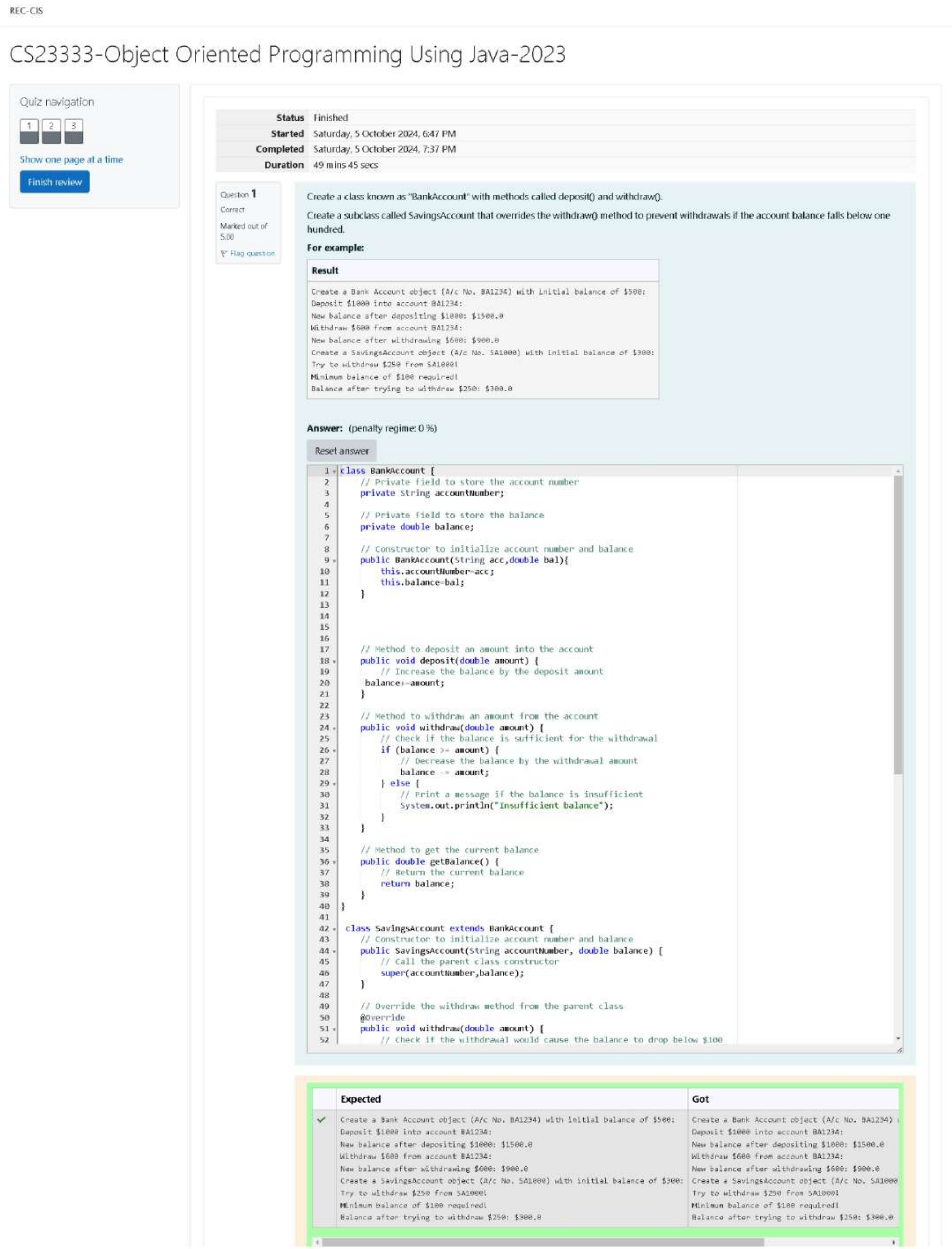
Answer: (penalty regime: 0 %)

```java
class Student{
    private String name;
    private int rollno;
    public Student(){
        System.out.println("No-arg constructor is invoked");
        this.name=null;
        this.rollno=0;
    }
    public Student(String name){
        System.out.println("1 arg constructor is invoked");
        this.name=name;
        this.rollno=0;
    }
    public Student(String name,int rollno){
        System.out.println("2 arg constructor is invoked");
        this.name=name;
        this.rollno=rollno;
    }
    public void display(){
        System.out.println("Name ="+ name + " "+", Roll no = "  +rollno);
    }}
    public class prog{
    public static void main(String args[]){
        Student s1 = new Student();
        Student s2 = new Student("Rajalakshmi");
        Student s3 = new Student("Lakshmi",101);
        s1.display();
        s2.display();
        s3.display();
    }
}
```

	Test	Expected	Got	

✓	1	No-arg constructor is invoked
		1 arg constructor is invoked
		2 arg constructor is invoked
		Name =null , Roll no = 0
		Name =Rajalakshmi , Roll no = 0
		Name =Lakshmi , Roll no = 101
			No-arg constructor is invoked
			1 arg constructor is invoked
			2 arg constructor is invoked
			Name =null , Roll no = 0
			Name =Rajalakshmi , Roll no = 0
			Name =Lakshmi , Roll no = 101
				✓

Passed all tests! ✓



REC-CIS

CS23333-Object Oriented Programming Using Java-2023

Status	Finished
Started	Saturday, 5 October 2024, 6:47 PM
Completed	Saturday, 5 October 2024, 7:37 PM
Duration	49 mins 45 secs

Question **1**
Correct
Marked out of 5.00

Create a class known as "BankAccount" with methods called deposit() and withdraw().

Create a subclass called SavingsAccount that overrides the withdraw() method to prevent withdrawals if the account balance falls below one hundred.

For example:

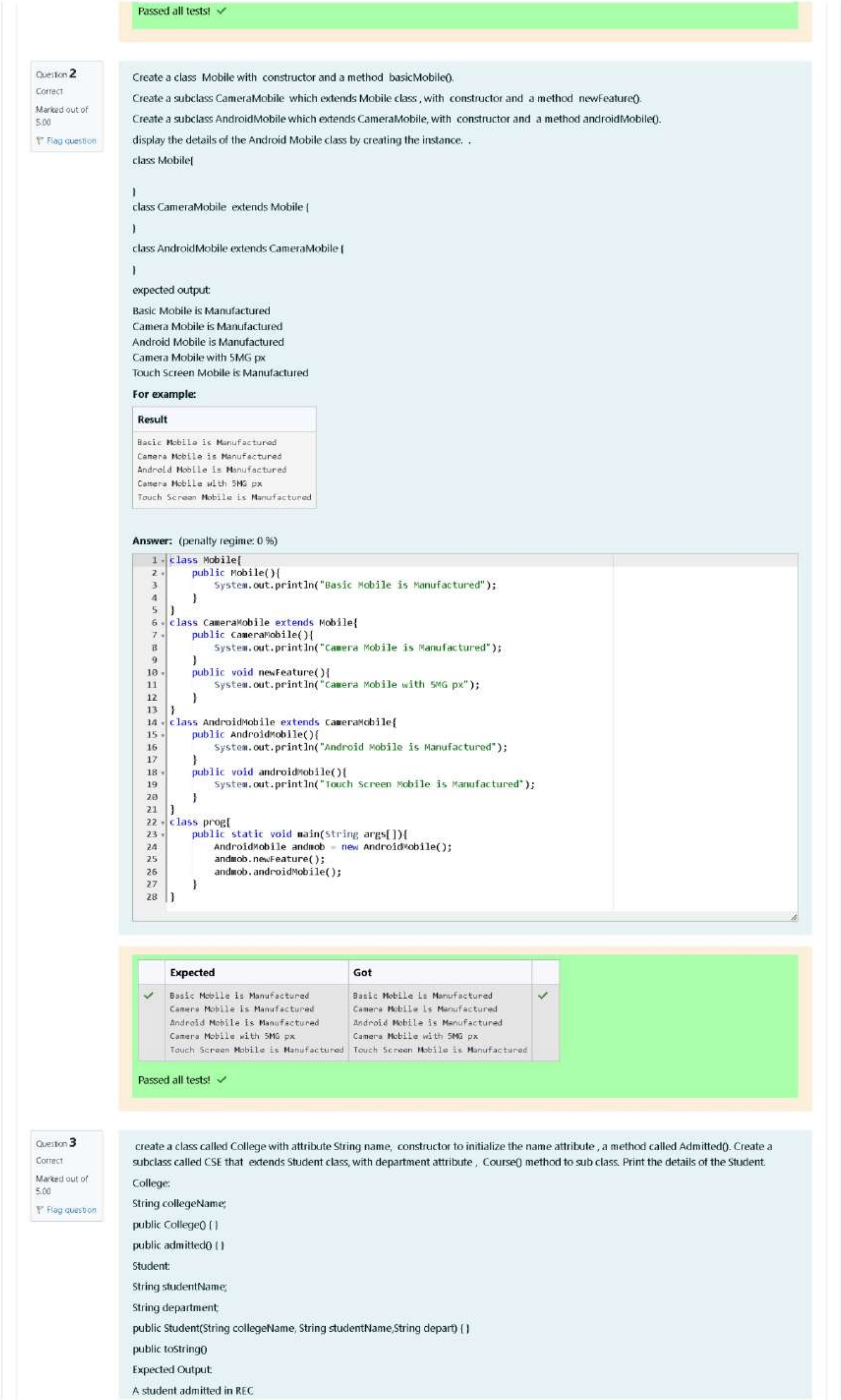
Result

```
Create a Bank Account object (A/c No. BA1234) with initial balance of $500:
Deposit $1000 into account BA1234:
New balance after depositing $1000: $1500.0
Withdraw $600 from account BA1234:
New balance after withdrawing $600: $900.0
Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300:
Try to withdraw $250 from SA1000!
Minimum balance of $100 required!
Balance after trying to withdraw $250: $300.0
```

Answer: (penalty regime: 0 %)

```java
class BankAccount {
    // Private field to store the account number
    private String accountNumber;

    // Private field to store the balance
    private double balance;

    // Constructor to initialize account number and balance
    public BankAccount(String acc,double bal){
        this.accountNumber=acc;
        this.balance=bal;
    }




    // Method to deposit an amount into the account
    public void deposit(double amount) {
        // Increase the balance by the deposit amount
     balance+=amount;
    }

    // Method to withdraw an amount from the account
    public void withdraw(double amount) {
        // Check if the balance is sufficient for the withdrawal
        if (balance >= amount) {
            // Decrease the balance by the withdrawal amount
            balance -= amount;
        } else {
            // Print a message if the balance is insufficient
            System.out.println("Insufficient balance");
        }
    }

    // Method to get the current balance
    public double getBalance() {
        // Return the current balance
        return balance;
    }
}

 class SavingsAccount extends BankAccount {
    // Constructor to initialize account number and balance
    public SavingsAccount(String accountNumber, double balance) {
        // Call the parent class constructor
        super(accountNumber,balance);
    }

    // Override the withdraw method from the parent class
    @Override
    public void withdraw(double amount) {
        // Check if the withdrawal would cause the balance to drop below $100
```

	Expected	Got
✓	Create a Bank Account object (A/c No. BA1234) with initial balance of $500: Deposit $1000 into account BA1234: New balance after depositing $1000: $1500.0 Withdraw $600 from account BA1234: New balance after withdrawing $600: $900.0 Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300: Try to withdraw $250 from SA1000! Minimum balance of $100 required! Balance after trying to withdraw $250: $300.0	Create a Bank Account object (A/c No. BA1234) Deposit $1000 into account BA1234: New balance after depositing $1000: $1500.0 Withdraw $600 from account BA1234: New balance after withdrawing $600: $900.0 Create a SavingsAccount object (A/c No. SA1000 Try to withdraw $250 from SA1000! Minimum balance of $100 required! Balance after trying to withdraw $250: $300.0



Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Flag question

Create a class Mobile with constructor and a method basicMobile().

Create a subclass CameraMobile which extends Mobile class , with constructor and a method newFeature().

Create a subclass AndroidMobile which extends CameraMobile, with constructor and a method androidMobile().

display the details of the Android Mobile class by creating the instance. .

class Mobile{

}

class CameraMobile extends Mobile {

}

class AndroidMobile extends CameraMobile {

}

expected output:

Basic Mobile is Manufactured
Camera Mobile is Manufactured
Android Mobile is Manufactured
Camera Mobile with 5MG px
Touch Screen Mobile is Manufactured

For example:

Result
Basic Mobile is Manufactured Camera Mobile is Manufactured Android Mobile is Manufactured Camera Mobile with 5MG px Touch Screen Mobile is Manufactured

Answer: (penalty regime: 0 %)

```
class Mobile{
    public Mobile(){
        System.out.println("Basic Mobile is Manufactured");
    }
}
class CameraMobile extends Mobile{
    public CameraMobile(){
        System.out.println("Camera Mobile is Manufactured");
    }
    public void newFeature(){
        System.out.println("Camera Mobile with 5MG px");
    }
}
class AndroidMobile extends CameraMobile{
    public AndroidMobile(){
        System.out.println("Android Mobile is Manufactured");
    }
    public void androidMobile(){
        System.out.println("Touch Screen Mobile is Manufactured");
    }
}
class prog{
    public static void main(String args[]){
        AndroidMobile andmob = new AndroidMobile();
        andmob.newFeature();
        andmob.androidMobile();
    }
}
```

	Expected	Got	
✓	Basic Mobile is Manufactured Camera Mobile is Manufactured Android Mobile is Manufactured Camera Mobile with 5MG px Touch Screen Mobile is Manufactured	Basic Mobile is Manufactured Camera Mobile is Manufactured Android Mobile is Manufactured Camera Mobile with 5MG px Touch Screen Mobile is Manufactured	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

Flag question

create a class called College with attribute String name, constructor to initialize the name attribute , a method called Admitted(). Create a subclass called CSE that extends Student class, with department attribute , Course() method to sub class. Print the details of the Student.

College:

String collegeName;

public College() { }

public admitted() { }

Student:

String studentName;

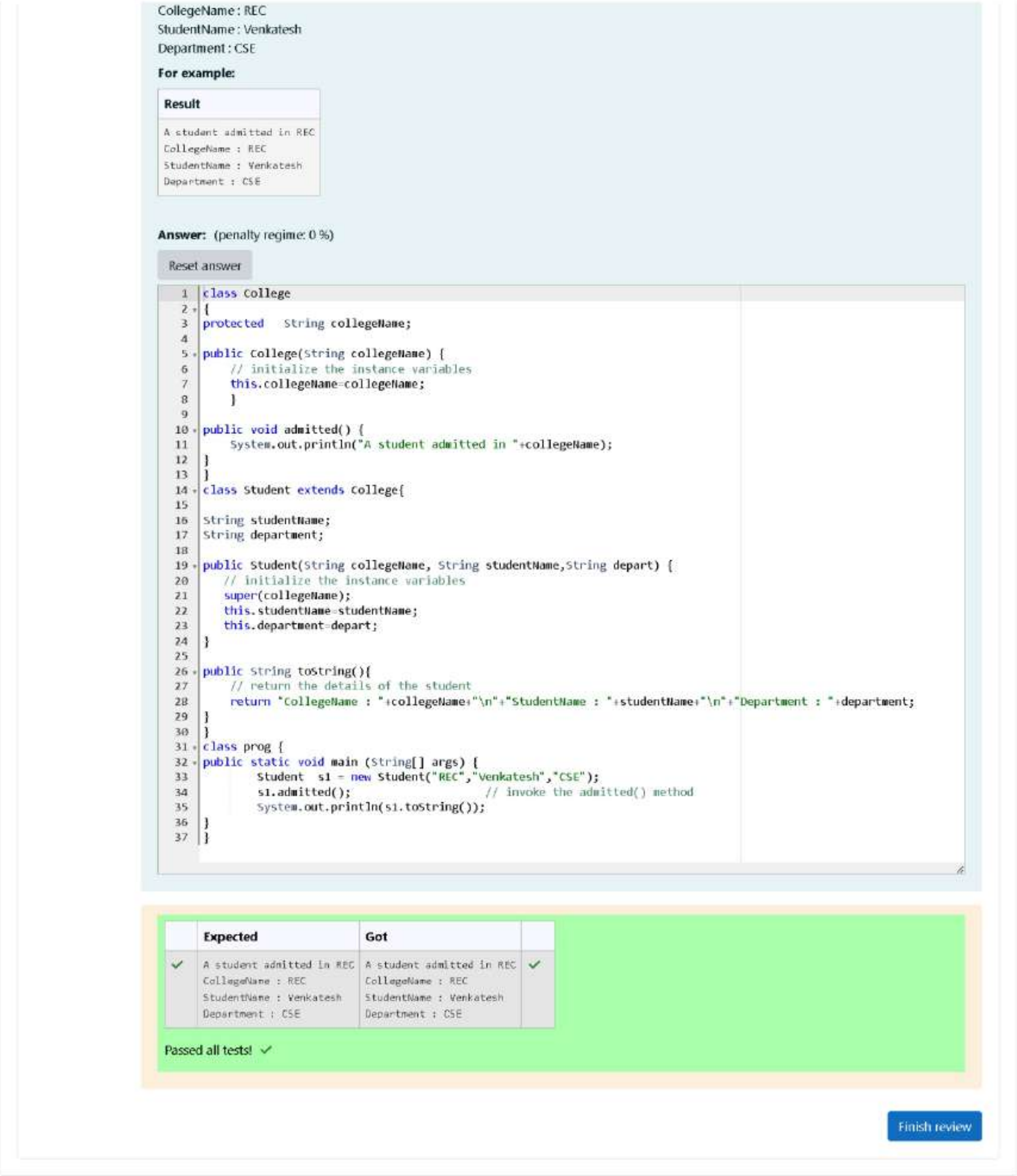
String department;

public Student(String collegeName, String studentName,String depart) { }

public toString()

Expected Output:

A student admitted in REC



CollegeName : REC
StudentName : Venkatesh
Department : CSE

For example:

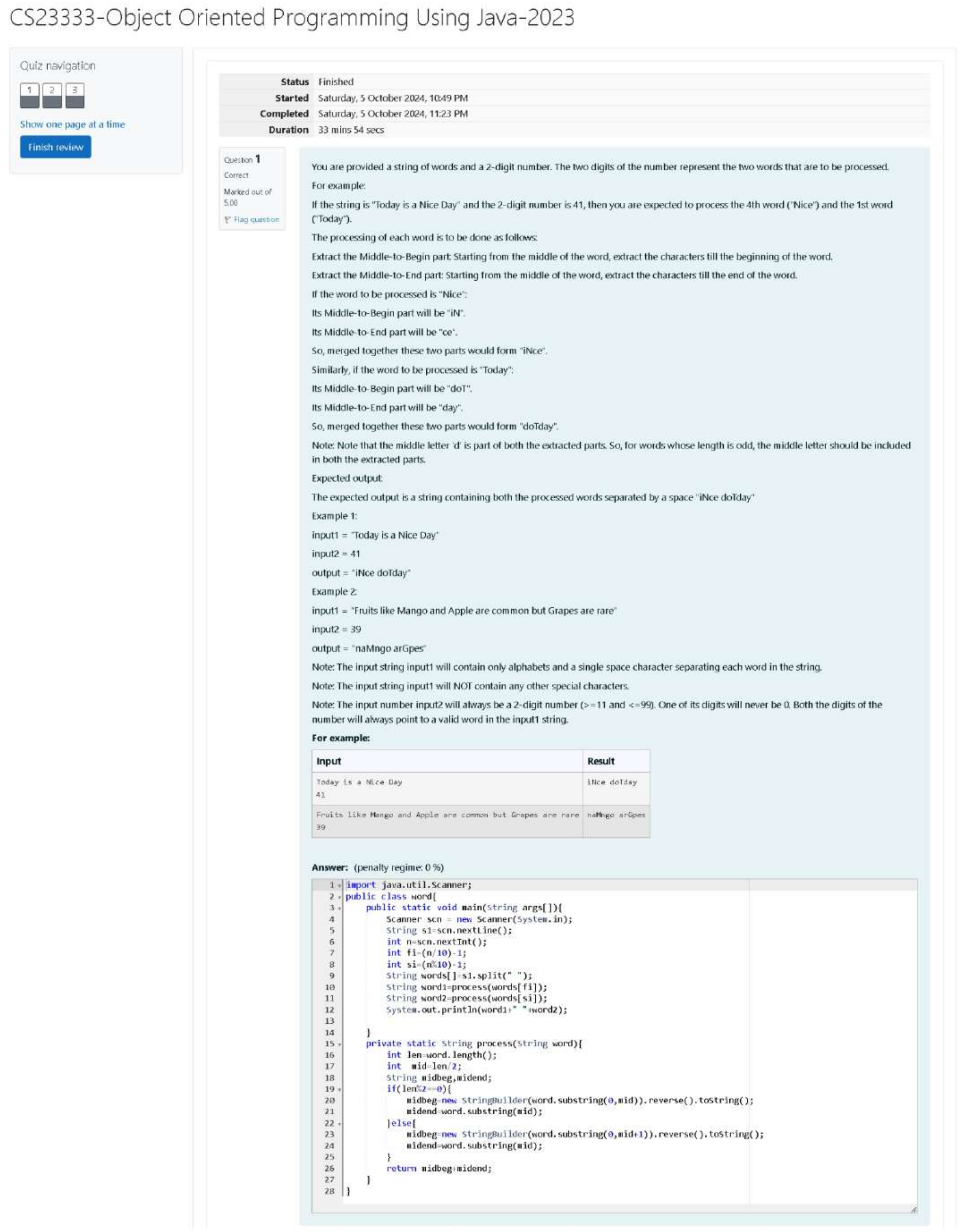
Result
A student admitted in REC CollegeName : REC StudentName : Venkatesh Department : CSE

Answer: (penalty regime: 0 %)

```
class College
{
protected   String collegeName;

public College(String collegeName) {
    // initialize the instance variables
    this.collegeName=collegeName;
    }

public void admitted() {
    System.out.println("A student admitted in "+collegeName);
}
}
class Student extends College{

String studentName;
String department;

public Student(String collegeName, String studentName,String depart) {
   // initialize the instance variables
   super(collegeName);
   this.studentName=studentName;
   this.department=depart;
}

public String toString(){
    // return the details of the student
    return "CollegeName : "+collegeName+"\n"+"StudentName : "+studentName+"\n"+"Department : "+department;
}
}
class prog {
public static void main (String[] args) {
        Student  s1 = new Student("REC","Venkatesh","CSE");
        s1.admitted();                        // invoke the admitted() method
        System.out.println(s1.toString());
}
}
```

	Expected	Got	
✓	A student admitted in REC CollegeName : REC StudentName : Venkatesh Department : CSE	A student admitted in REC CollegeName : REC StudentName : Venkatesh Department : CSE	✓

Passed all tests! ✓



CS23333-Object Oriented Programming Using Java-2023

Status	Finished
Started	Saturday, 5 October 2024, 10:49 PM
Completed	Saturday, 5 October 2024, 11:23 PM
Duration	33 mins 54 secs

Question **1**
Correct
Marked out of 5.00

You are provided a string of words and a 2-digit number. The two digits of the number represent the two words that are to be processed.

For example:

If the string is "Today is a Nice Day" and the 2-digit number is 41, then you are expected to process the 4th word ("Nice") and the 1st word ("Today").

The processing of each word is to be done as follows:

Extract the Middle-to-Begin part: Starting from the middle of the word, extract the characters till the beginning of the word.

Extract the Middle-to-End part: Starting from the middle of the word, extract the characters till the end of the word.

If the word to be processed is "Nice":

Its Middle-to-Begin part will be "iN".

Its Middle-to-End part will be "ce".

So, merged together these two parts would form "iNce".

Similarly, if the word to be processed is "Today":

Its Middle-to-Begin part will be "doT".

Its Middle-to-End part will be "day".

So, merged together these two parts would form "doTday".

Note: Note that the middle letter 'd' is part of both the extracted parts. So, for words whose length is odd, the middle letter should be included in both the extracted parts.

Expected output:

The expected output is a string containing both the processed words separated by a space "iNce doTday"

Example 1:

input1 = "Today is a Nice Day"

input2 = 41

output = "iNce doTday"

Example 2:

input1 = "Fruits like Mango and Apple are common but Grapes are rare"

input2 = 39

output = "naMngo arGpes"

Note: The input string input1 will contain only alphabets and a single space character separating each word in the string.

Note: The input string input1 will NOT contain any other special characters.

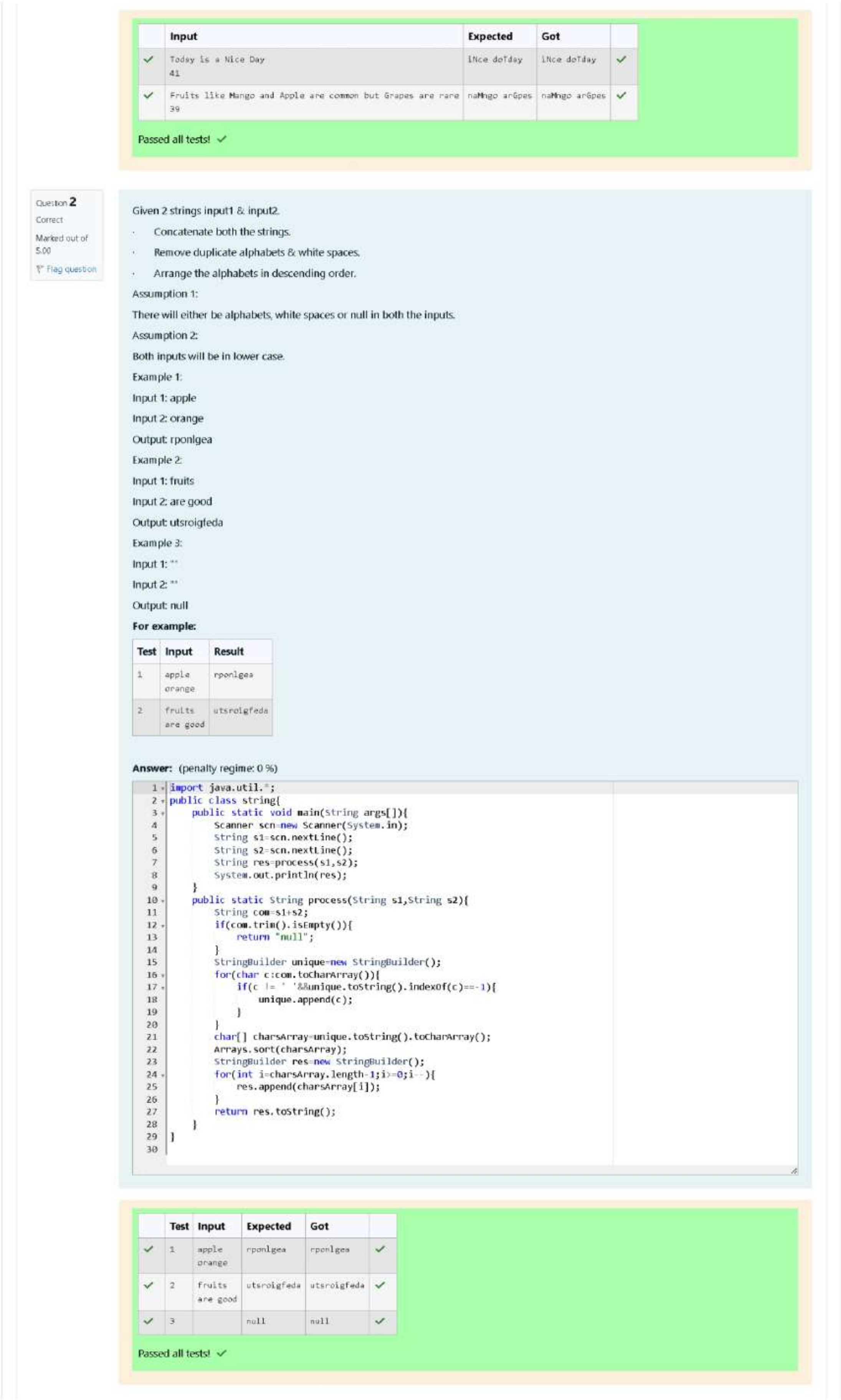
Note: The input number input2 will always be a 2-digit number (>=11 and <=99). One of its digits will never be 0. Both the digits of the number will always point to a valid word in the input1 string.

For example:

Input	Result
Today is a Nice Day 41	iNce doTday
Fruits like Mango and Apple are common but Grapes are rare 39	naMngo arGpes

Answer: (penalty regime: 0 %)

```java
import java.util.Scanner;
public class word{
    public static void main(String args[]){
        Scanner scn = new Scanner(System.in);
        String s1=scn.nextLine();
        int n=scn.nextInt();
        int fi=(n/10)-1;
        int si=(n%10)-1;
        String words[]=s1.split(" ");
        String word1=process(words[fi]);
        String word2=process(words[si]);
        System.out.println(word1+" "+word2);

    }
    private static String process(String word){
        int len=word.length();
        int  mid=len/2;
        String midbeg,midend;
        if(len%2==0){
            midbeg=new StringBuilder(word.substring(0,mid)).reverse().toString();
            midend=word.substring(mid);
        }else{
            midbeg=new StringBuilder(word.substring(0,mid+1)).reverse().toString();
            midend=word.substring(mid);
        }
        return midbeg+midend;
    }
}
```



	Input	Expected	Got	
✓	Today is a Nice Day 41	iNce doTday	iNce doTday	✓
✓	Fruits like Mango and Apple are common but Grapes are rare 39	naMngo arGpes	naMngo arGpes	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Flag question

Given 2 strings input1 & input2.

- Concatenate both the strings.
- Remove duplicate alphabets & white spaces.
- Arrange the alphabets in descending order.

Assumption 1:

There will either be alphabets, white spaces or null in both the inputs.

Assumption 2:

Both inputs will be in lower case.

Example 1:

Input 1: apple

Input 2: orange

Output: rponlgea

Example 2:

Input 1: fruits

Input 2: are good

Output: utsroigfeda

Example 3:

Input 1: ""

Input 2: ""

Output: null

For example:

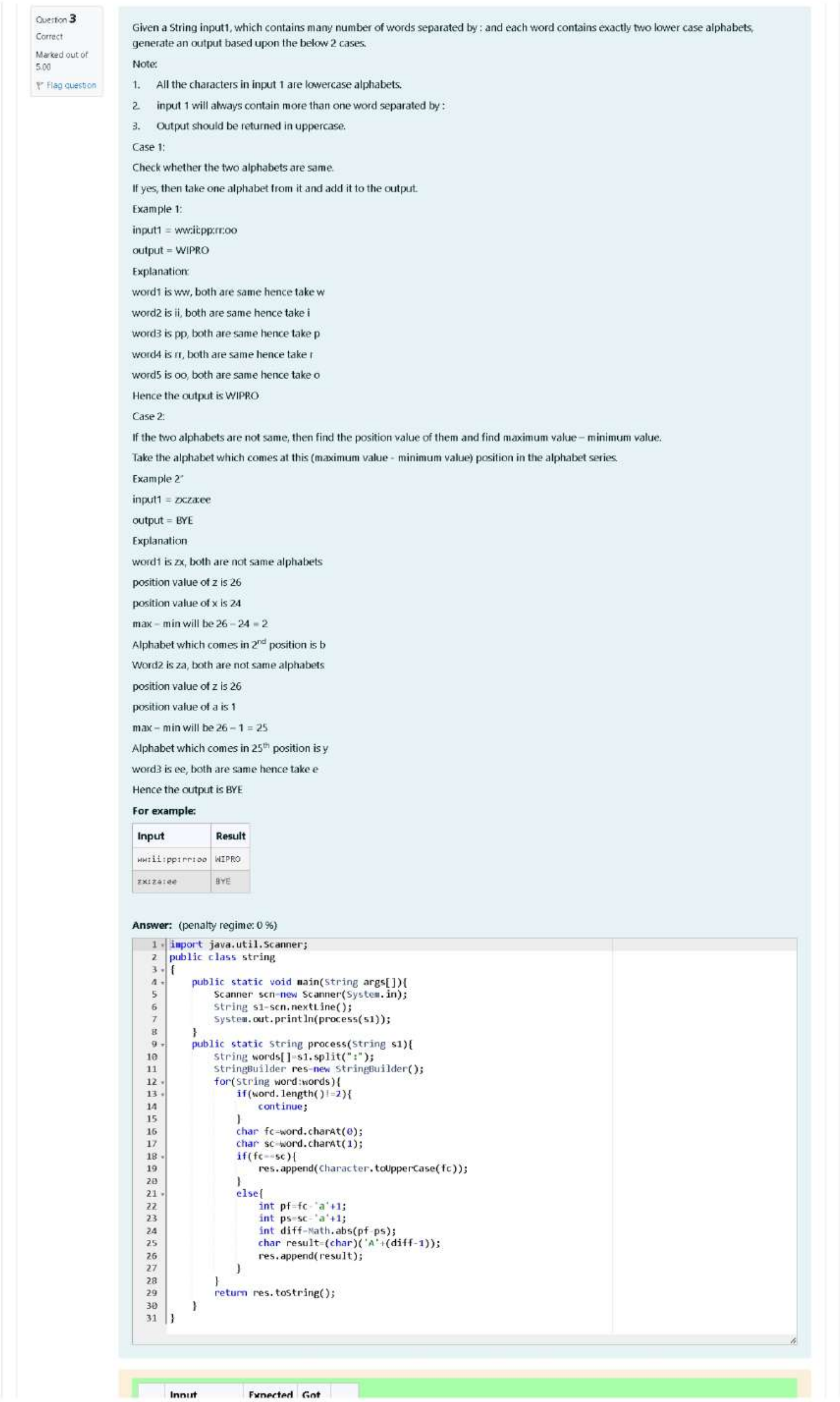
Test	Input	Result
1	apple orange	rponlgea
2	fruits are good	utsroigfeda

Answer: (penalty regime: 0 %)

```java
import java.util.*;
public class string{
    public static void main(String args[]){
        Scanner scn=new Scanner(System.in);
        String s1=scn.nextLine();
        String s2=scn.nextLine();
        String res=process(s1,s2);
        System.out.println(res);
    }
    public static String process(String s1,String s2){
        String com=s1+s2;
        if(com.trim().isEmpty()){
            return "null";
        }
        StringBuilder unique=new StringBuilder();
        for(char c:com.toCharArray()){
            if(c != ' '&&unique.toString().indexOf(c)==-1){
                unique.append(c);
            }
        }
        char[] charsArray=unique.toString().toCharArray();
        Arrays.sort(charsArray);
        StringBuilder res=new StringBuilder();
        for(int i=charsArray.length-1;i>=0;i--){
            res.append(charsArray[i]);
        }
        return res.toString();
    }
}

```

	Test	Input	Expected	Got	
✓	1	apple orange	rponlgea	rponlgea	✓
✓	2	fruits are good	utsroigfeda	utsroigfeda	✓
✓	3		null	null	✓

Passed all tests! ✓



Question 3

Correct

Marked out of 5.00

Flag question

Given a String input1, which contains many number of words separated by : and each word contains exactly two lower case alphabets, generate an output based upon the below 2 cases.

Note:

1. All the characters in input 1 are lowercase alphabets.
2. input 1 will always contain more than one word separated by :
3. Output should be returned in uppercase.

Case 1:

Check whether the two alphabets are same.

If yes, then take one alphabet from it and add it to the output.

Example 1:

input1 = ww:ii:pp:rr:oo

output = WIPRO

Explanation:

word1 is ww, both are same hence take w

word2 is ii, both are same hence take i

word3 is pp, both are same hence take p

word4 is rr, both are same hence take r

word5 is oo, both are same hence take o

Hence the output is WIPRO

Case 2:

If the two alphabets are not same, then find the position value of them and find maximum value – minimum value.

Take the alphabet which comes at this (maximum value - minimum value) position in the alphabet series.

Example 2"

input1 = zx:za:ee

output = BYE

Explanation

word1 is zx, both are not same alphabets

position value of z is 26

position value of x is 24

max – min will be 26 – 24 = 2

Alphabet which comes in 2nd position is b

Word2 is za, both are not same alphabets

position value of z is 26

position value of a is 1

max – min will be 26 – 1 = 25

Alphabet which comes in 25th position is y

word3 is ee, both are same hence take e

Hence the output is BYE

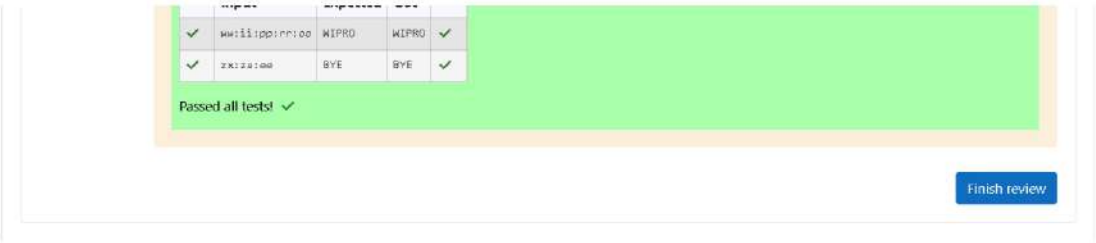
For example:

Input	Result
ww:ii:pp:rr:oo	WIPRO
zx:za:ee	BYE

Answer: (penalty regime: 0 %)

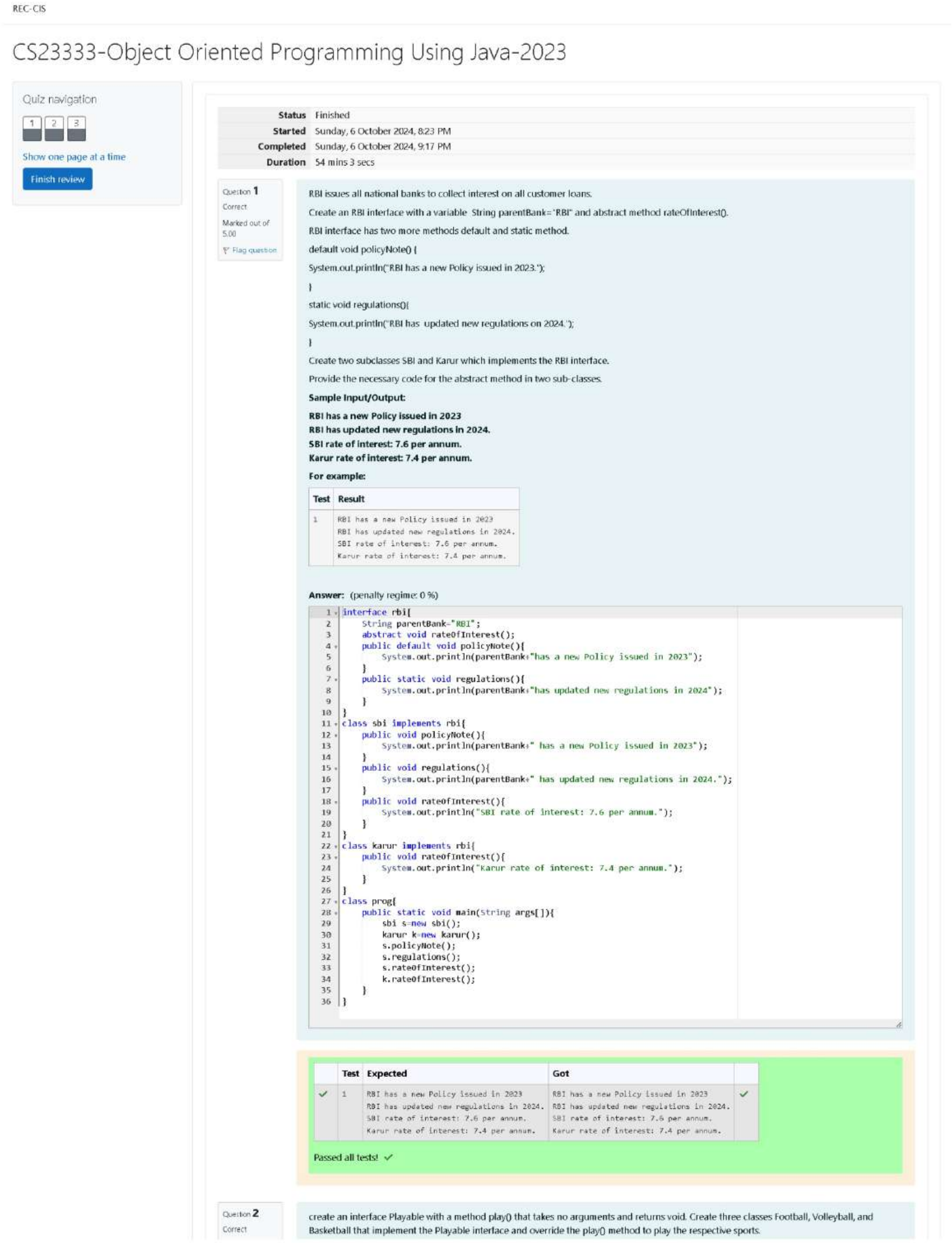
```java
import java.util.Scanner;
public class string
{
    public static void main(String args[]){
        Scanner scn=new Scanner(System.in);
        String s1=scn.nextLine();
        System.out.println(process(s1));
    }
    public static String process(String s1){
        String words[]=s1.split(":");
        StringBuilder res=new StringBuilder();
        for(String word:words){
            if(word.length()!=2){
                continue;
            }
            char fc=word.charAt(0);
            char sc=word.charAt(1);
            if(fc==sc){
                res.append(Character.toUpperCase(fc));
            }
            else{
                int pf=fc-'a'+1;
                int ps=sc-'a'+1;
                int diff=Math.abs(pf-ps);
                char result=(char)('A'+(diff-1));
                res.append(result);
            }
        }
        return res.toString();
    }
}
```

	Input	Expected	Got	



✓	ww:ii:pp:rr:oo	WIPRO	WIPRO	✓
✓	zx:za:ee	BYE	BYE	✓

Passed all tests! ✓



REC-CIS

CS23333-Object Oriented Programming Using Java-2023

Status	Finished
Started	Sunday, 6 October 2024, 8:23 PM
Completed	Sunday, 6 October 2024, 9:17 PM
Duration	54 mins 3 secs

Question **1**
Correct
Marked out of 5.00

RBI issues all national banks to collect interest on all customer loans.

Create an RBI interface with a variable String parentBank="RBI" and abstract method rateOfInterest().

RBI interface has two more methods default and static method.

default void policyNote() {

System.out.println("RBI has a new Policy issued in 2023.");

}

static void regulations(){

System.out.println("RBI has updated new regulations on 2024.");

}

Create two subclasses SBI and Karur which implements the RBI interface.

Provide the necessary code for the abstract method in two sub-classes.

Sample Input/Output:

RBI has a new Policy issued in 2023
RBI has updated new regulations in 2024.
SBI rate of interest: 7.6 per annum.
Karur rate of interest: 7.4 per annum.

For example:

Test	Result
1	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.

Answer: (penalty regime: 0 %)

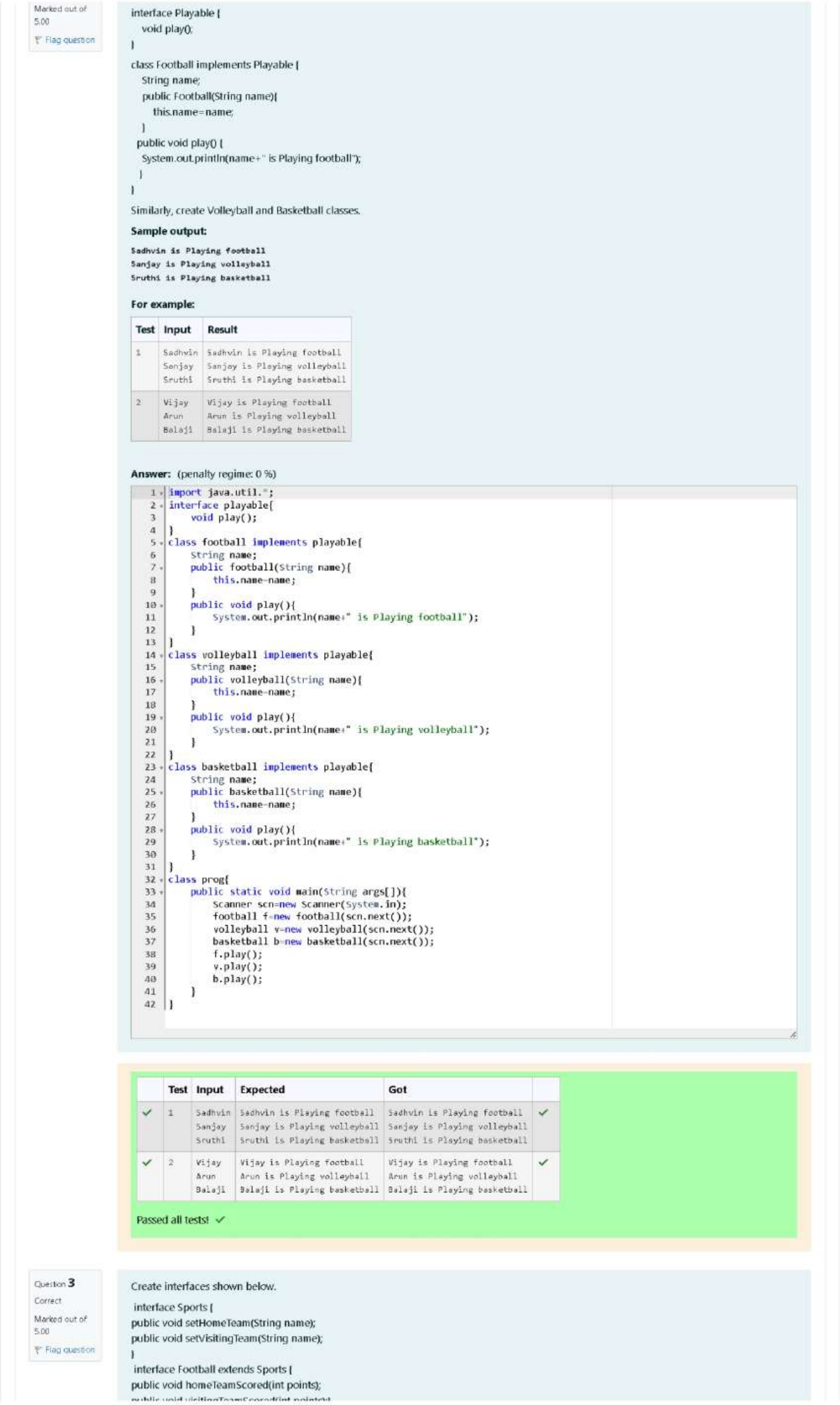
```java
interface rbi{
    String parentBank="RBI";
    abstract void rateOfInterest();
    public default void policyNote(){
        System.out.println(parentBank+"has a new Policy issued in 2023");
    }
    public static void regulations(){
        System.out.println(parentBank+"has updated new regulations in 2024");
    }
}
class sbi implements rbi{
    public void policyNote(){
        System.out.println(parentBank+" has a new Policy issued in 2023");
    }
    public void regulations(){
        System.out.println(parentBank+" has updated new regulations in 2024.");
    }
    public void rateOfInterest(){
        System.out.println("SBI rate of interest: 7.6 per annum.");
    }
}
class karur implements rbi{
    public void rateOfInterest(){
        System.out.println("Karur rate of interest: 7.4 per annum.");
    }
}
class prog{
    public static void main(String args[]){
        sbi s=new sbi();
        karur k=new karur();
        s.policyNote();
        s.regulations();
        s.rateOfInterest();
        k.rateOfInterest();
    }
}
```

	Test	Expected	Got	
✓	1	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.	✓

Passed all tests! ✓

Question **2**
Correct

create an interface Playable with a method play() that takes no arguments and returns void. Create three classes Football, Volleyball, and Basketball that implement the Playable interface and override the play() method to play the respective sports.



Marked out of 5.00

Flag question

```
interface Playable {
   void play();
}
class Football implements Playable {
   String name;
   public Football(String name){
      this.name=name;
   }
  public void play() {
   System.out.println(name+" is Playing football");
  }
}
```

Similarly, create Volleyball and Basketball classes.

Sample output:

```
Sadhvin is Playing football
Sanjay is Playing volleyball
Sruthi is Playing basketball
```

For example:

Test	Input	Result
1	Sadhvin Sanjay Sruthi	Sadhvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball
2	Vijay Arun Balaji	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball

Answer: (penalty regime: 0 %)

```java
import java.util.*;
interface playable{
    void play();
}
class football implements playable{
    String name;
    public football(String name){
        this.name=name;
    }
    public void play(){
        System.out.println(name+" is Playing football");
    }
}
class volleyball implements playable{
    String name;
    public volleyball(String name){
        this.name=name;
    }
    public void play(){
        System.out.println(name+" is Playing volleyball");
    }
}
class basketball implements playable{
    String name;
    public basketball(String name){
        this.name=name;
    }
    public void play(){
        System.out.println(name+" is Playing basketball");
    }
}
class prog{
    public static void main(String args[]){
        Scanner scn=new Scanner(System.in);
        football f=new football(scn.next());
        volleyball v=new volleyball(scn.next());
        basketball b=new basketball(scn.next());
        f.play();
        v.play();
        b.play();
    }
}
```

	Test	Input	Expected	Got	
✓	1	Sadhvin Sanjay Sruthi	Sadhvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball	Sadhvin is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball	✓
✓	2	Vijay Arun Balaji	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball	Vijay is Playing football Arun is Playing volleyball Balaji is Playing basketball	✓

Passed all tests! ✓

Question **3**

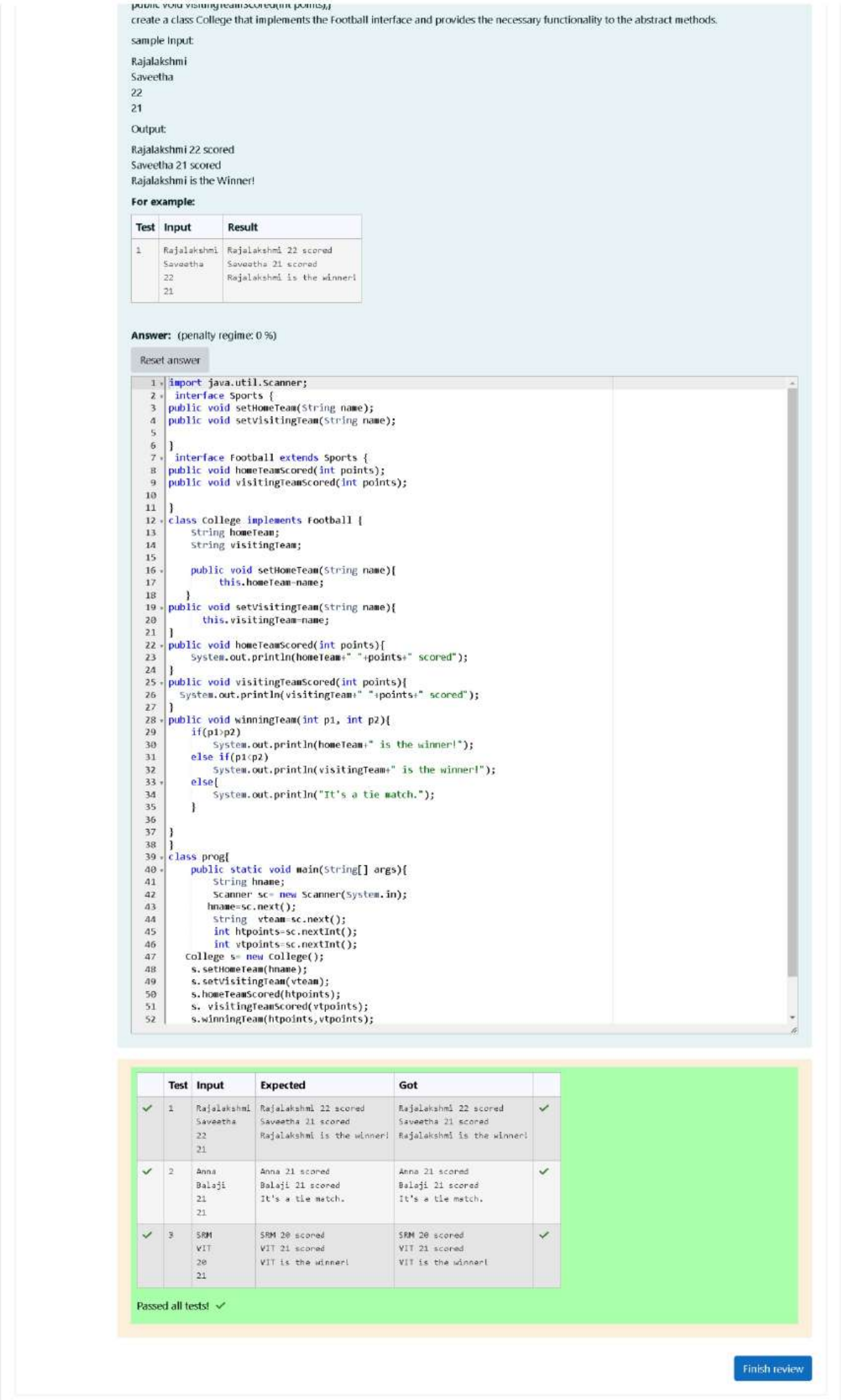
Correct

Marked out of 5.00

Flag question

Create interfaces shown below.

```
interface Sports {
public void setHomeTeam(String name);
public void setVisitingTeam(String name);
}
interface Football extends Sports {
public void homeTeamScored(int points);
public void visitingTeamScored(int points);
```



public void visitingTeamScored(int points);}

create a class College that implements the Football interface and provides the necessary functionality to the abstract methods.

sample Input:

Rajalakshmi
Saveetha
22
21

Output:

Rajalakshmi 22 scored
Saveetha 21 scored
Rajalakshmi is the Winner!

For example:

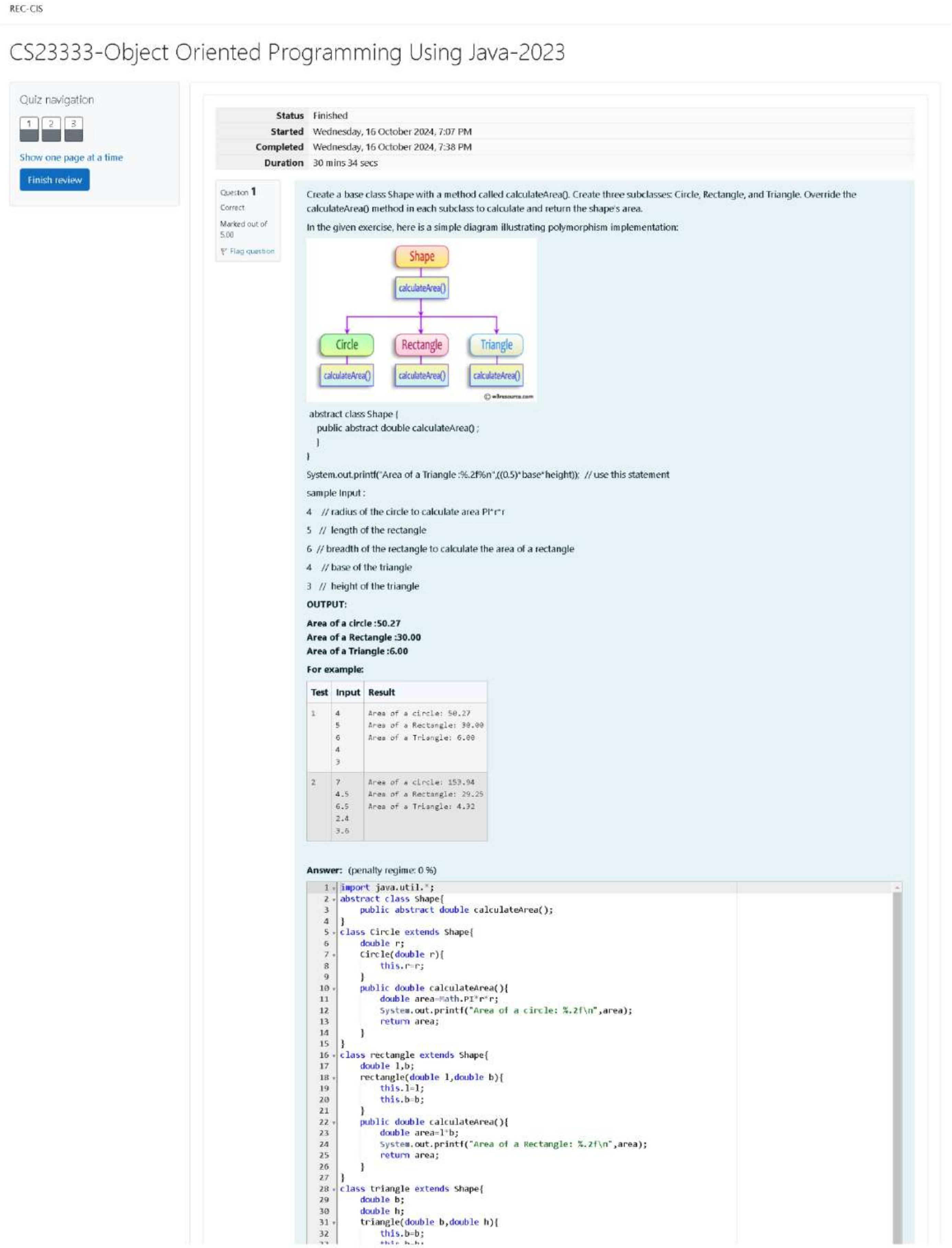
Test	Input	Result
1	Rajalakshmi Saveetha 22 21	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!

Answer: (penalty regime: 0 %)

```java
import java.util.Scanner;
 interface Sports {
public void setHomeTeam(String name);
public void setVisitingTeam(String name);

}
 interface Football extends Sports {
public void homeTeamScored(int points);
public void visitingTeamScored(int points);

}
class College implements Football {
    String homeTeam;
    String visitingTeam;

    public void setHomeTeam(String name){
         this.homeTeam=name;
     }
public void setVisitingTeam(String name){
       this.visitingTeam=name;
}
public void homeTeamScored(int points){
    System.out.println(homeTeam+" "+points+" scored");
}
public void visitingTeamScored(int points){
  System.out.println(visitingTeam+" "+points+" scored");
}
public void winningTeam(int p1, int p2){
    if(p1>p2)
        System.out.println(homeTeam+" is the winner!");
    else if(p1<p2)
        System.out.println(visitingTeam+" is the winner!");
    else{
        System.out.println("It's a tie match.");
    }

}
}
class prog{
    public static void main(String[] args){
        String hname;
        Scanner sc= new Scanner(System.in);
       hname=sc.next();
        String  vteam=sc.next();
        int htpoints=sc.nextInt();
        int vtpoints=sc.nextInt();
   College s= new College();
    s.setHomeTeam(hname);
    s.setVisitingTeam(vteam);
    s.homeTeamScored(htpoints);
    s. visitingTeamScored(vtpoints);
    s.winningTeam(htpoints,vtpoints);
```

	Test	Input	Expected	Got	
✓	1	Rajalakshmi Saveetha 22 21	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!	✓
✓	2	Anna Balaji 21 21	Anna 21 scored Balaji 21 scored It's a tie match.	Anna 21 scored Balaji 21 scored It's a tie match.	✓
✓	3	SRM VIT 20 21	SRM 20 scored VIT 21 scored VIT is the winner!	SRM 20 scored VIT 21 scored VIT is the winner!	✓

Passed all tests! ✓



CS23333-Object Oriented Programming Using Java-2023

Status	Finished
Started	Wednesday, 16 October 2024, 7:07 PM
Completed	Wednesday, 16 October 2024, 7:38 PM
Duration	30 mins 34 secs

Question **1**

Correct

Marked out of 5.00

Create a base class Shape with a method called calculateArea(). Create three subclasses: Circle, Rectangle, and Triangle. Override the calculateArea() method in each subclass to calculate and return the shape's area.

In the given exercise, here is a simple diagram illustrating polymorphism implementation:

```
 abstract class Shape {
    public abstract double calculateArea();
    }
}
```

System.out.printf("Area of a Triangle :%.2f%n",((0.5)*base*height)); // use this statement

sample Input :

4 // radius of the circle to calculate area PI*r*r

5 // length of the rectangle

6 // breadth of the rectangle to calculate the area of a rectangle

4 // base of the triangle

3 // height of the triangle

OUTPUT:

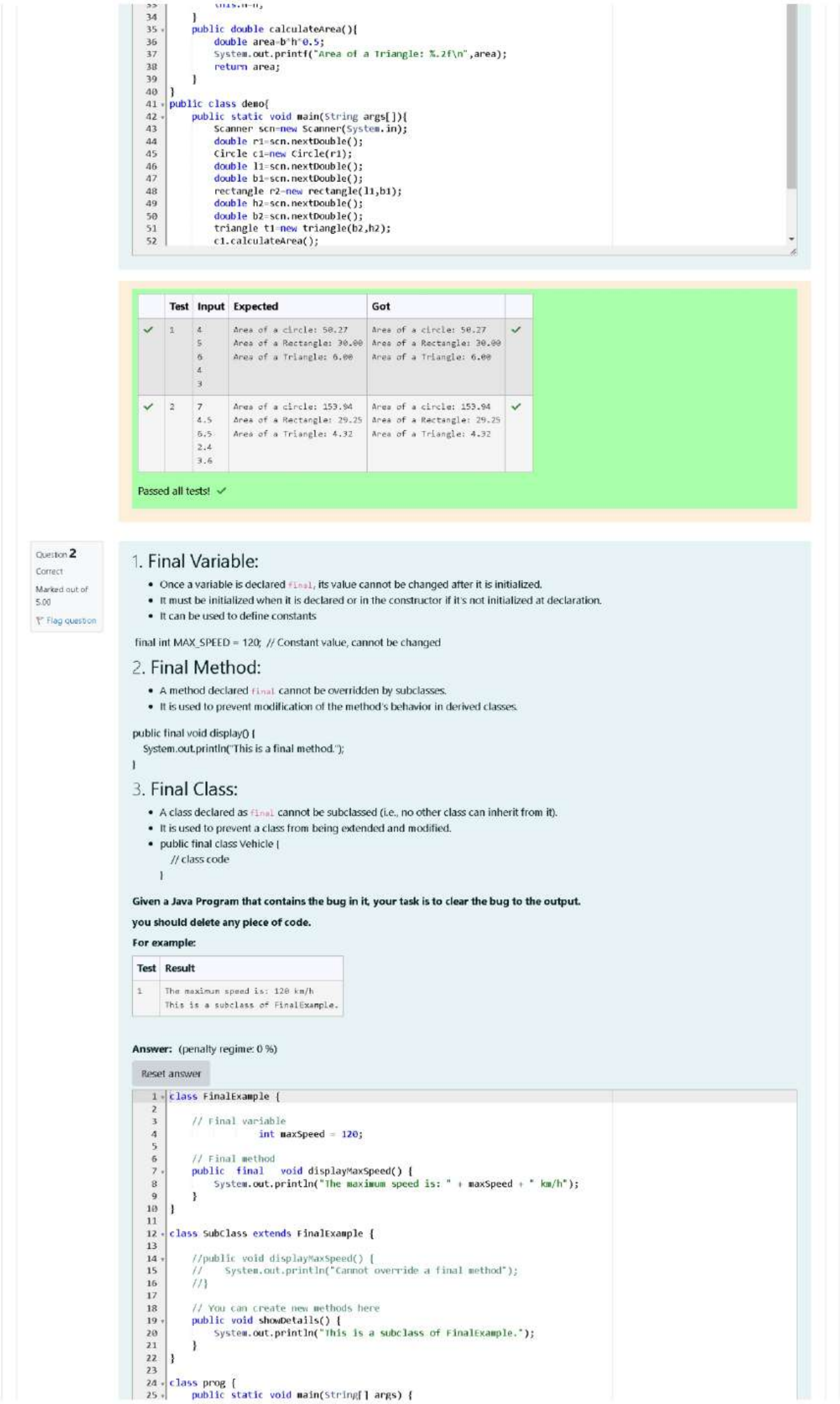
Area of a circle :50.27
Area of a Rectangle :30.00
Area of a Triangle :6.00

For example:

Test	Input	Result
1	4 5 6 4 3	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00
2	7 4.5 6.5 2.4 3.6	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32

Answer: (penalty regime: 0 %)

```java
import java.util.*;
abstract class Shape{
    public abstract double calculateArea();
}
class Circle extends Shape{
    double r;
    Circle(double r){
        this.r=r;
    }
    public double calculateArea(){
        double area=Math.PI*r*r;
        System.out.printf("Area of a circle: %.2f\n",area);
        return area;
    }
}
class rectangle extends Shape{
    double l,b;
    rectangle(double l,double b){
        this.l=l;
        this.b=b;
    }
    public double calculateArea(){
        double area=l*b;
        System.out.printf("Area of a Rectangle: %.2f\n",area);
        return area;
    }
}
class triangle extends Shape{
    double b;
    double h;
    triangle(double b,double h){
        this.b=b;
        this.h=h;
```



```
        this.h=h;
    }
    public double calculateArea(){
        double area=b*h*0.5;
        System.out.printf("Area of a Triangle: %.2f\n",area);
        return area;
    }
}
public class demo{
    public static void main(String args[]){
        Scanner scn=new Scanner(System.in);
        double r1=scn.nextDouble();
        Circle c1=new Circle(r1);
        double l1=scn.nextDouble();
        double b1=scn.nextDouble();
        rectangle r2=new rectangle(l1,b1);
        double h2=scn.nextDouble();
        double b2=scn.nextDouble();
        triangle t1=new triangle(b2,h2);
        c1.calculateArea();
```

	Test	Input	Expected	Got	
✓	1	4 5 6 4 3	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00	Area of a circle: 50.27 Area of a Rectangle: 30.00 Area of a Triangle: 6.00	✓
✓	2	7 4.5 6.5 2.4 3.6	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32	Area of a circle: 153.94 Area of a Rectangle: 29.25 Area of a Triangle: 4.32	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Flag question

1. Final Variable:

- Once a variable is declared `final`, its value cannot be changed after it is initialized.
- It must be initialized when it is declared or in the constructor if it's not initialized at declaration.
- It can be used to define constants

final int MAX_SPEED = 120; // Constant value, cannot be changed

2. Final Method:

- A method declared `final` cannot be overridden by subclasses.
- It is used to prevent modification of the method's behavior in derived classes.

public final void display() {
  System.out.println("This is a final method.");
}

3. Final Class:

- A class declared as `final` cannot be subclassed (i.e., no other class can inherit from it).
- It is used to prevent a class from being extended and modified.
- public final class Vehicle {
      // class code
  }

Given a Java Program that contains the bug in it, your task is to clear the bug to the output.

you should delete any piece of code.

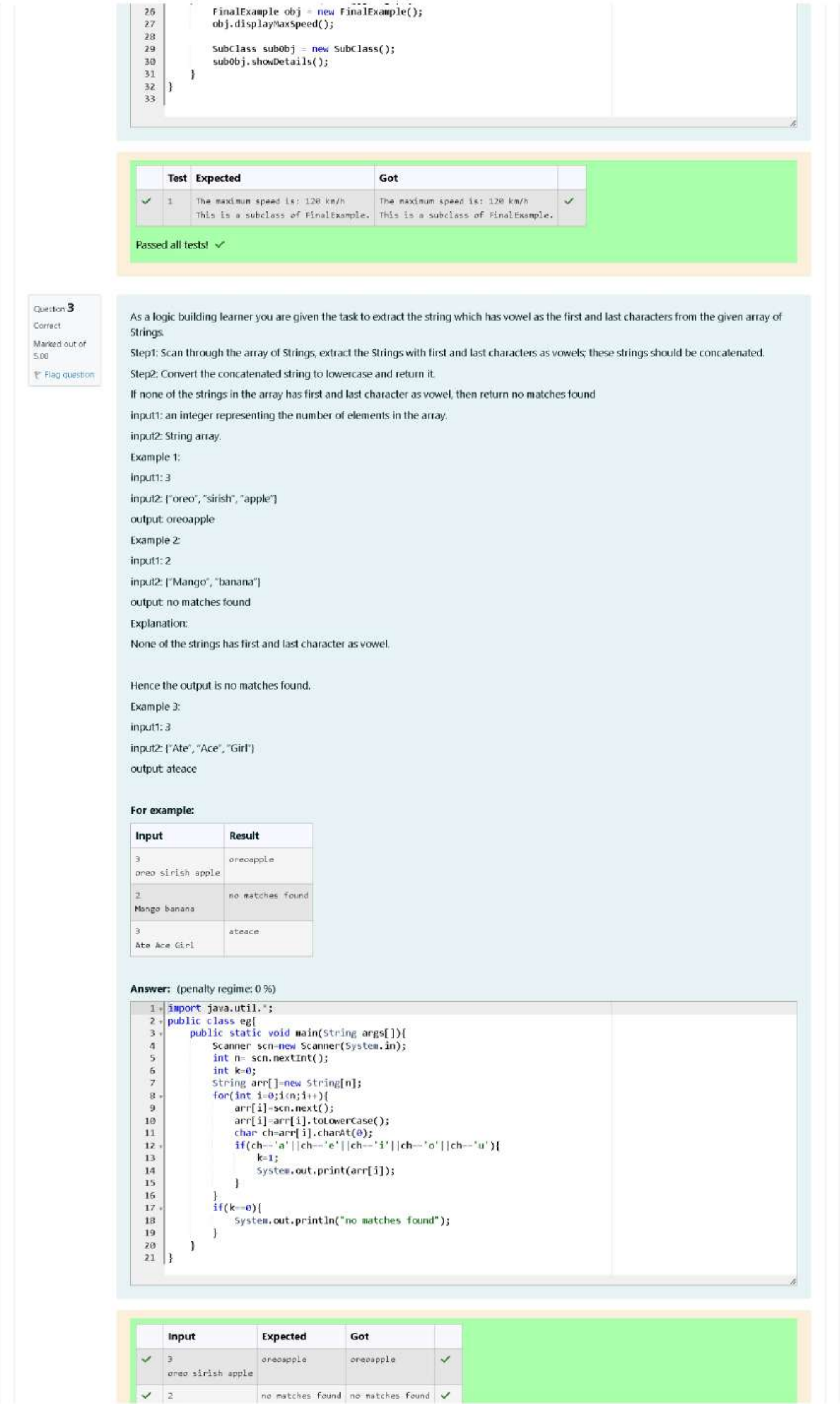
For example:

Test	Result
1	The maximum speed is: 120 km/h This is a subclass of FinalExample.

Answer: (penalty regime: 0 %)

Reset answer

```
class FinalExample {

    // Final variable
                int maxSpeed = 120;

    // Final method
    public  final   void displayMaxSpeed() {
        System.out.println("The maximum speed is: " + maxSpeed + " km/h");
    }
}

class SubClass extends FinalExample {

    //public void displayMaxSpeed() {
    //    System.out.println("Cannot override a final method");
    //}

    // You can create new methods here
    public void showDetails() {
        System.out.println("This is a subclass of FinalExample.");
    }
}

class prog {
    public static void main(String[] args) {
```



```
        FinalExample obj = new FinalExample();
        obj.displayMaxSpeed();

        SubClass subObj = new SubClass();
        subObj.showDetails();
    }
}

```

	Test	Expected	Got	
✓	1	The maximum speed is: 120 km/h This is a subclass of FinalExample.	The maximum speed is: 120 km/h This is a subclass of FinalExample.	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

Flag question

As a logic building learner you are given the task to extract the string which has vowel as the first and last characters from the given array of Strings.

Step1: Scan through the array of Strings, extract the Strings with first and last characters as vowels; these strings should be concatenated.

Step2: Convert the concatenated string to lowercase and return it.

If none of the strings in the array has first and last character as vowel, then return no matches found

input1: an integer representing the number of elements in the array.

input2: String array.

Example 1:

input1: 3

input2: {"oreo", "sirish", "apple"}

output: oreoapple

Example 2:

input1: 2

input2: {"Mango", "banana"}

output: no matches found

Explanation:

None of the strings has first and last character as vowel.

Hence the output is no matches found.

Example 3:

input1: 3

input2: {"Ate", "Ace", "Girl"}

output: ateace

For example:

Input	Result
3 oreo sirish apple	oreoapple
2 Mango banana	no matches found
3 Ate Ace Girl	ateace

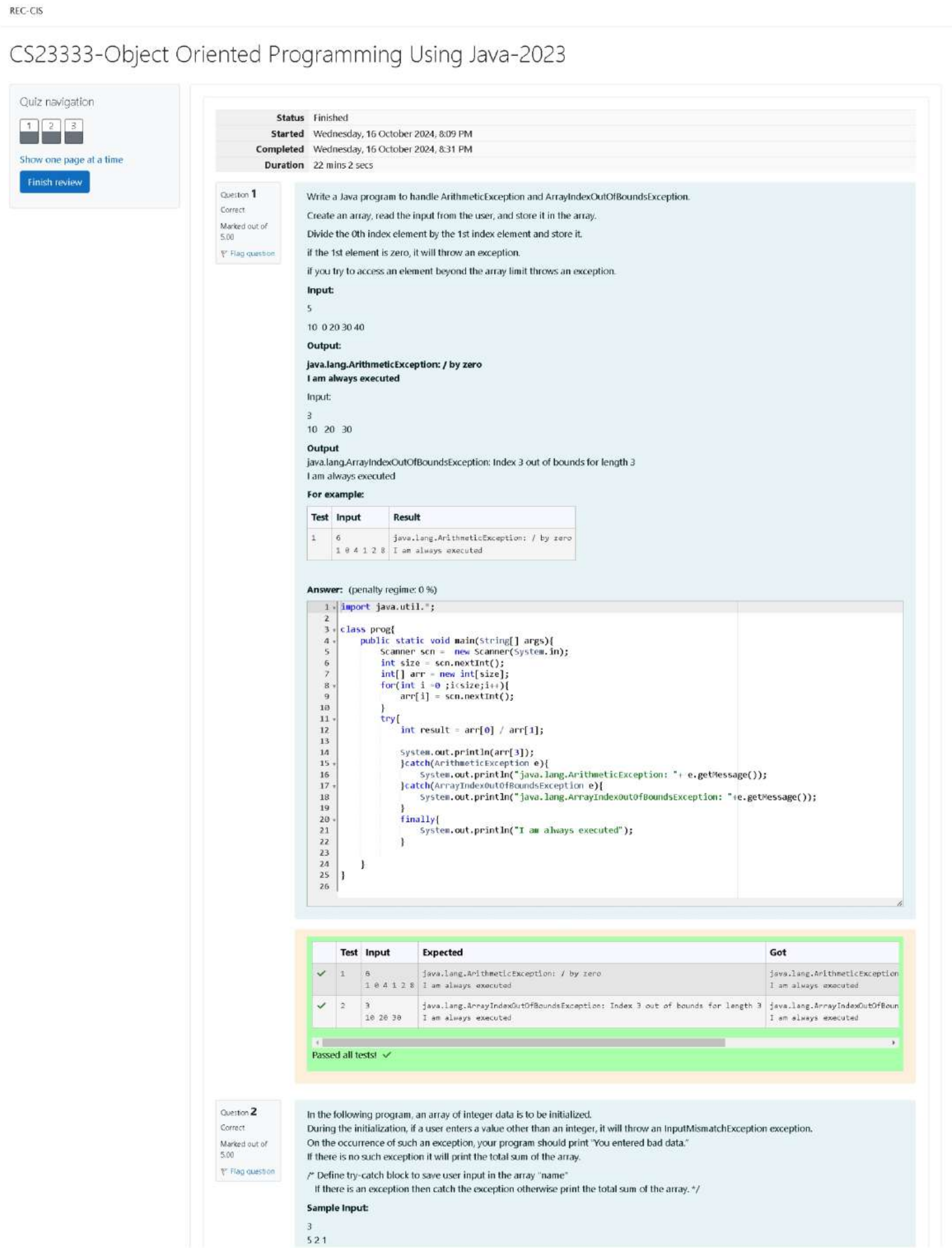
Answer: (penalty regime: 0 %)

```
import java.util.*;
public class eg{
    public static void main(String args[]){
        Scanner scn=new Scanner(System.in);
        int n= scn.nextInt();
        int k=0;
        String arr[]=new String[n];
        for(int i=0;i<n;i++){
            arr[i]=scn.next();
            arr[i]=arr[i].toLowerCase();
            char ch=arr[i].charAt(0);
            if(ch=='a'||ch=='e'||ch=='i'||ch=='o'||ch=='u'){
                k=1;
                System.out.print(arr[i]);
            }
        }
        if(k==0){
            System.out.println("no matches found");
        }
    }
}
```

	Input	Expected	Got	
✓	3 oreo sirish apple	oreoapple	oreoapple	✓
✓	2	no matches found	no matches found	✓

	Mango banana			
✓	3 Ate Ace Girl	ateace	ateace	✓

Passed all tests! ✓



REC-CIS

CS23333-Object Oriented Programming Using Java-2023

Status	Finished
Started	Wednesday, 16 October 2024, 8:09 PM
Completed	Wednesday, 16 October 2024, 8:31 PM
Duration	22 mins 2 secs

Question **1**
Correct
Marked out of 5.00

Write a Java program to handle ArithmeticException and ArrayIndexOutOfBoundsException.

Create an array, read the input from the user, and store it in the array.

Divide the 0th index element by the 1st index element and store it.

if the 1st element is zero, it will throw an exception.

if you try to access an element beyond the array limit throws an exception.

Input:

5

10 0 20 30 40

Output:

java.lang.ArithmeticException: / by zero
I am always executed

Input:

3
10 20 30

Output
java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3
I am always executed

For example:

Test	Input	Result
1	6 1 0 4 1 2 8	java.lang.ArithmeticException: / by zero I am always executed

Answer: (penalty regime: 0 %)

```java
import java.util.*;

class prog{
    public static void main(String[] args){
        Scanner scn =  new Scanner(System.in);
        int size = scn.nextInt();
        int[] arr = new int[size];
        for(int i =0 ;i<size;i++){
            arr[i] = scn.nextInt();
        }
        try{
            int result = arr[0] / arr[1];

            System.out.println(arr[3]);
            }catch(ArithmeticException e){
                System.out.println("java.lang.ArithmeticException: "+ e.getMessage());
            }catch(ArrayIndexOutOfBoundsException e){
                System.out.println("java.lang.ArrayIndexOutOfBoundsException: "+e.getMessage());
            }
            finally{
                System.out.println("I am always executed");
            }

    }
}

```

	Test	Input	Expected	Got
✓	1	6 1 0 4 1 2 8	java.lang.ArithmeticException: / by zero I am always executed	java.lang.ArithmeticException I am always executed
✓	2	3 10 20 30	java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3 I am always executed	java.lang.ArrayIndexOutOfBoun I am always executed

Passed all tests! ✓

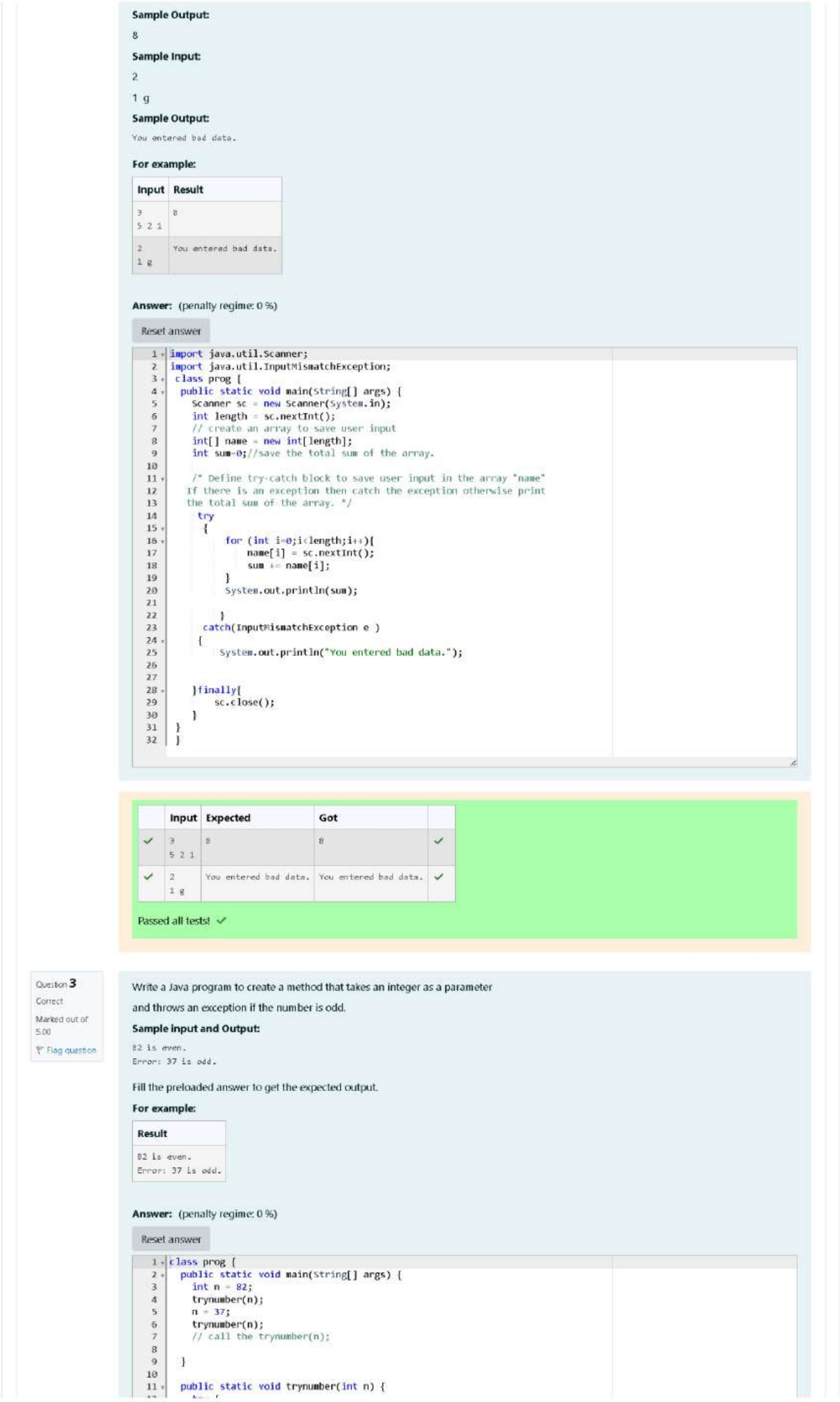
Question **2**
Correct
Marked out of 5.00

In the following program, an array of integer data is to be initialized.
During the initialization, if a user enters a value other than an integer, it will throw an InputMismatchException exception.
On the occurrence of such an exception, your program should print "You entered bad data."
If there is no such exception it will print the total sum of the array.

/* Define try-catch block to save user input in the array "name"
 If there is an exception then catch the exception otherwise print the total sum of the array. */

Sample Input:

3
5 2 1



Sample Output:

8

Sample Input:

2

1 g

Sample Output:

```
You entered bad data.
```

For example:

Input	Result
3 5 2 1	8
2 1 g	You entered bad data.

Answer: (penalty regime: 0 %)

Reset answer

```java
import java.util.Scanner;
import java.util.InputMismatchException;
 class prog {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int length = sc.nextInt();
    // create an array to save user input
    int[] name = new int[length];
    int sum=0;//save the total sum of the array.

    /* Define try-catch block to save user input in the array "name"
   If there is an exception then catch the exception otherwise print
   the total sum of the array. */
     try
      {
          for (int i=0;i<length;i++){
              name[i] = sc.nextInt();
              sum += name[i];
          }
          System.out.println(sum);

        }
      catch(InputMismatchException e )
     {
         System.out.println("You entered bad data.");


    }finally{
        sc.close();
    }
 }
 }
```

	Input	Expected	Got	
✓	3 5 2 1	8	8	✓
✓	2 1 g	You entered bad data.	You entered bad data.	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

Flag question

Write a Java program to create a method that takes an integer as a parameter

and throws an exception if the number is odd.

Sample input and Output:

```
82 is even.
Error: 37 is odd.
```

Fill the preloaded answer to get the expected output.

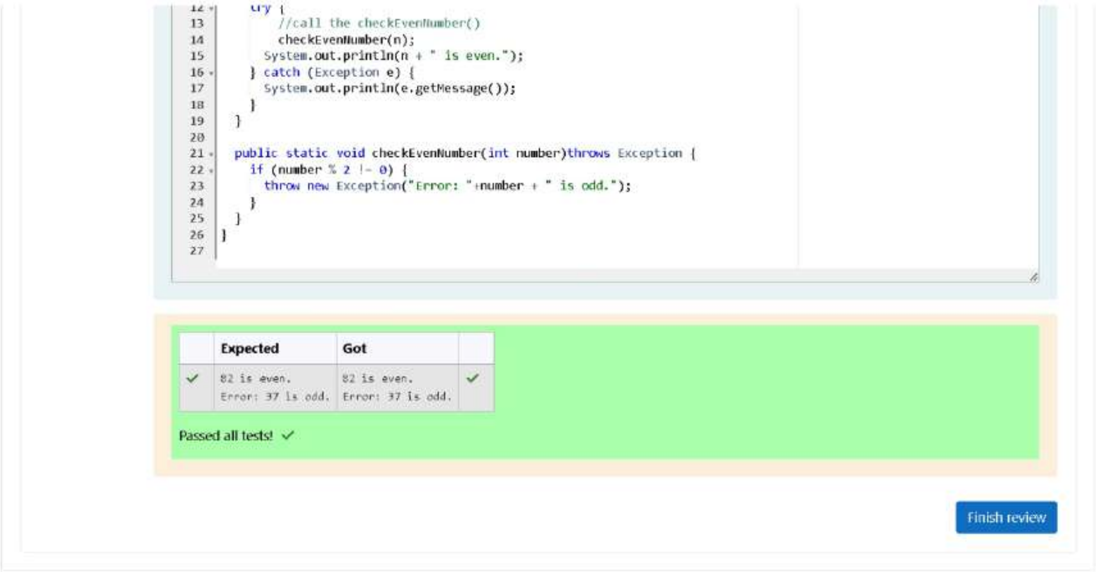
For example:

Result
82 is even. Error: 37 is odd.

Answer: (penalty regime: 0 %)

Reset answer

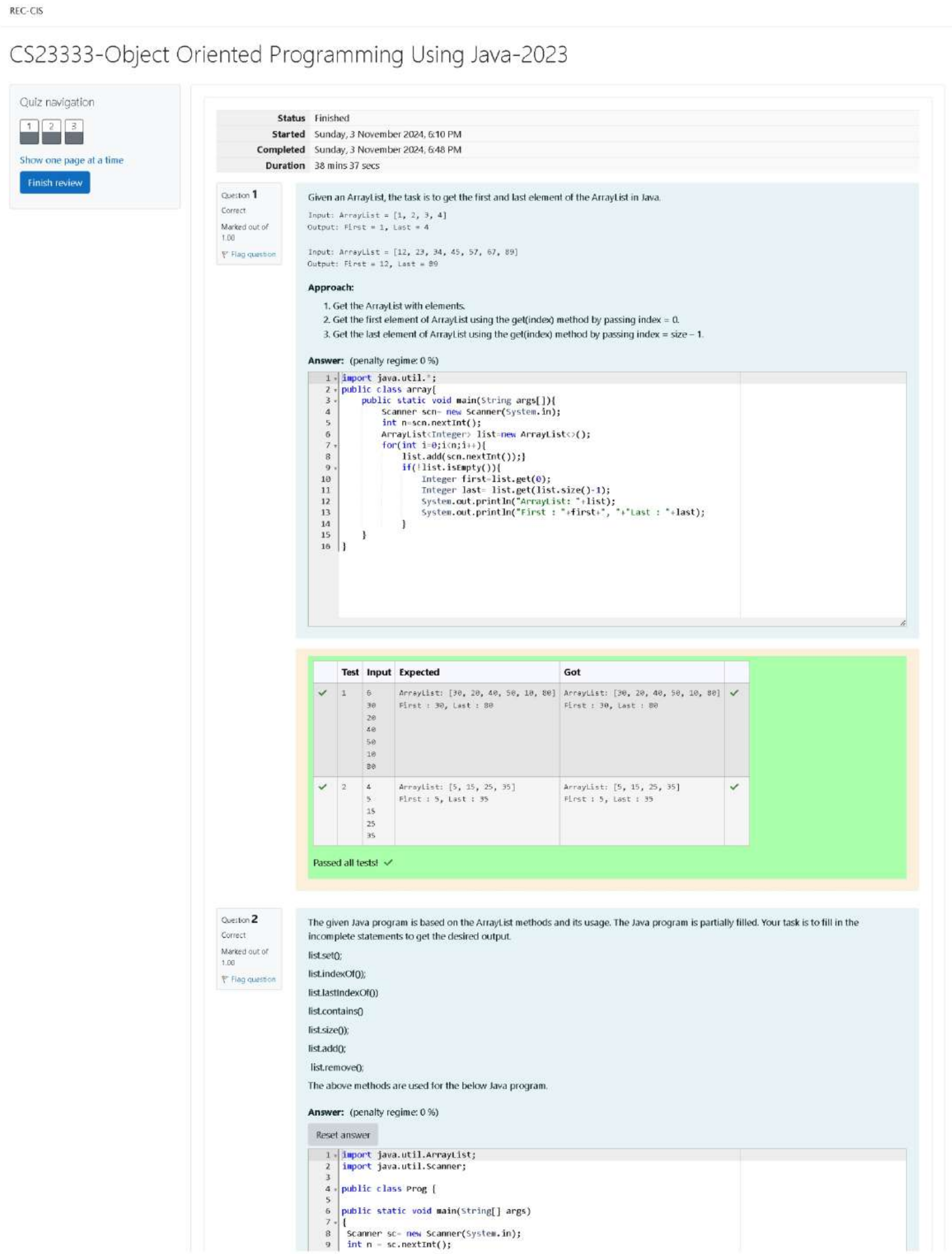
```java
class prog {
  public static void main(String[] args) {
    int n = 82;
    trynumber(n);
    n = 37;
    trynumber(n);
    // call the trynumber(n);

  }

  public static void trynumber(int n) {
```



```java
    try {
        //call the checkEvenNumber()
        checkEvenNumber(n);
      System.out.println(n + " is even.");
    } catch (Exception e) {
      System.out.println(e.getMessage());
    }
  }

  public static void checkEvenNumber(int number)throws Exception {
    if (number % 2 != 0) {
      throw new Exception("Error: "+number + " is odd.");
    }
  }
}
```

	Expected	Got	
✓	82 is even. Error: 37 is odd.	82 is even. Error: 37 is odd.	✓

Passed all tests! ✓



REC-CIS

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

Finish review

Status	Finished
Started	Sunday, 3 November 2024, 6:10 PM
Completed	Sunday, 3 November 2024, 6:48 PM
Duration	38 mins 37 secs

Question **1**
Correct
Marked out of 1.00
Flag question

Given an ArrayList, the task is to get the first and last element of the ArrayList in Java.

```
Input: ArrayList = [1, 2, 3, 4]
Output: First = 1, Last = 4

Input: ArrayList = [12, 23, 34, 45, 57, 67, 89]
Output: First = 12, Last = 89
```

Approach:

1. Get the ArrayList with elements.
2. Get the first element of ArrayList using the get(index) method by passing index = 0.
3. Get the last element of ArrayList using the get(index) method by passing index = size – 1.

Answer: (penalty regime: 0 %)

```java
import java.util.*;
public class array{
    public static void main(String args[]){
        Scanner scn= new Scanner(System.in);
        int n=scn.nextInt();
        ArrayList<Integer> list=new ArrayList<>();
        for(int i=0;i<n;i++){
            list.add(scn.nextInt());}
            if(!list.isEmpty()){
                Integer first=list.get(0);
                Integer last= list.get(list.size()-1);
                System.out.println("ArrayList: "+list);
                System.out.println("First : "+first+", "+"Last : "+last);
            }
    }
}
```

	Test	Input	Expected	Got	
✓	1	6 30 20 40 50 10 80	ArrayList: [30, 20, 40, 50, 10, 80] First : 30, Last : 80	ArrayList: [30, 20, 40, 50, 10, 80] First : 30, Last : 80	✓
✓	2	4 5 15 25 35	ArrayList: [5, 15, 25, 35] First : 5, Last : 35	ArrayList: [5, 15, 25, 35] First : 5, Last : 35	✓

Passed all tests! ✓

Question **2**
Correct
Marked out of 1.00
Flag question

The given Java program is based on the ArrayList methods and its usage. The Java program is partially filled. Your task is to fill in the incomplete statements to get the desired output.

list.set();

list.indexOf());

list.lastIndexOf())

list.contains()

list.size());

list.add();

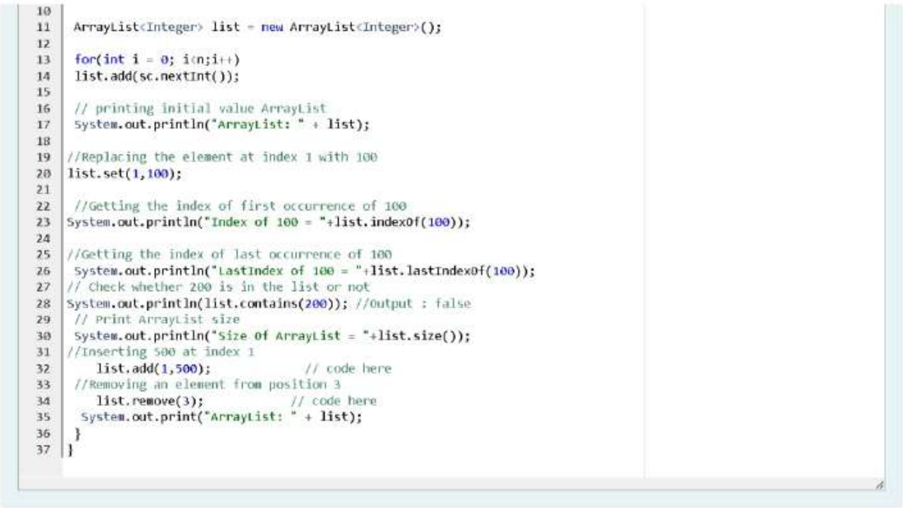
list.remove();

The above methods are used for the below Java program.

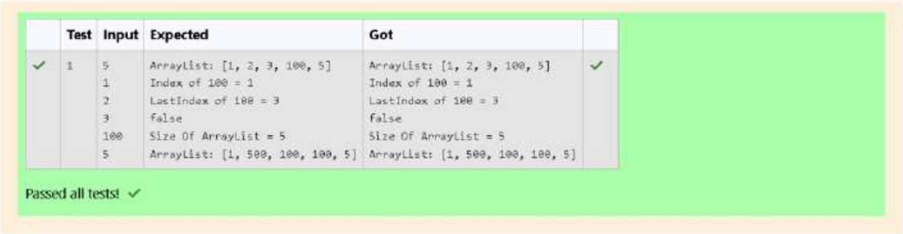
Answer: (penalty regime: 0 %)

Reset answer

```java
import java.util.ArrayList;
import java.util.Scanner;

public class Prog {

public static void main(String[] args)
{
 Scanner sc= new Scanner(System.in);
 int n = sc.nextInt();
```



```
 ArrayList<Integer> list = new ArrayList<Integer>();

 for(int i = 0; i<n;i++)
 list.add(sc.nextInt());

 // printing initial value ArrayList
 System.out.println("ArrayList: " + list);

//Replacing the element at index 1 with 100
list.set(1,100);

 //Getting the index of first occurrence of 100
System.out.println("Index of 100 = "+list.indexOf(100));

//Getting the index of last occurrence of 100
 System.out.println("LastIndex of 100 = "+list.lastIndexOf(100));
// Check whether 200 is in the list or not
System.out.println(list.contains(200)); //Output : false
 // Print ArrayList size
 System.out.println("Size Of ArrayList = "+list.size());
//Inserting 500 at index 1
    list.add(1,500);              // code here
 //Removing an element from position 3
    list.remove(3);               // code here
  System.out.print("ArrayList: " + list);
 }
}
```



	Test	Input	Expected	Got	
✓	1	5 1 2 3 100 5	ArrayList: [1, 2, 3, 100, 5] Index of 100 = 1 LastIndex of 100 = 3 false Size Of ArrayList = 5 ArrayList: [1, 500, 100, 100, 5]	ArrayList: [1, 2, 3, 100, 5] Index of 100 = 1 LastIndex of 100 = 3 false Size Of ArrayList = 5 ArrayList: [1, 500, 100, 100, 5]	✓

Passed all tests! ✓

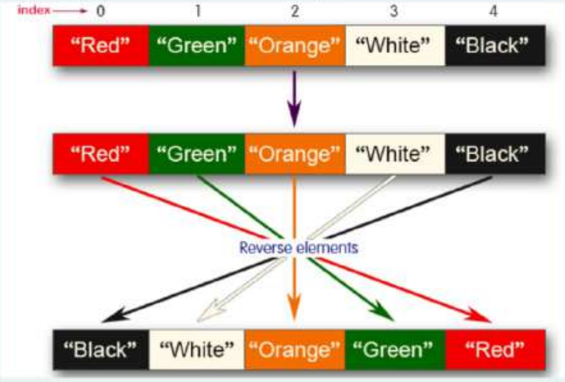
Question **3**

Correct

Marked out of 1.00

Flag question

Write a Java program to reverse elements in an array list.

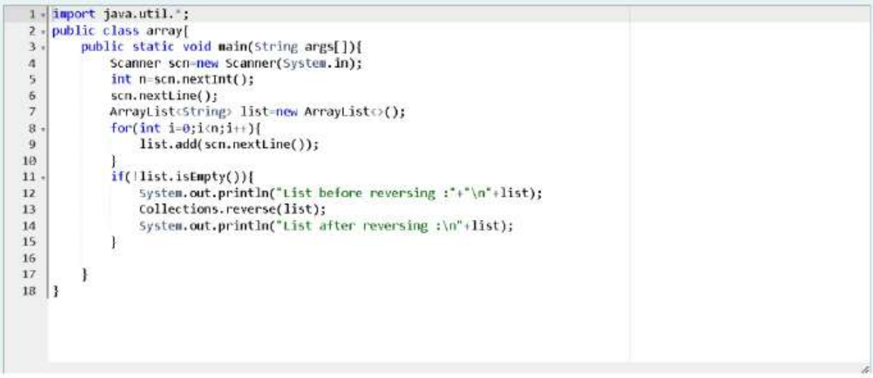


```
Sample input and Output:
Red
Green
Orange
White
Black
```

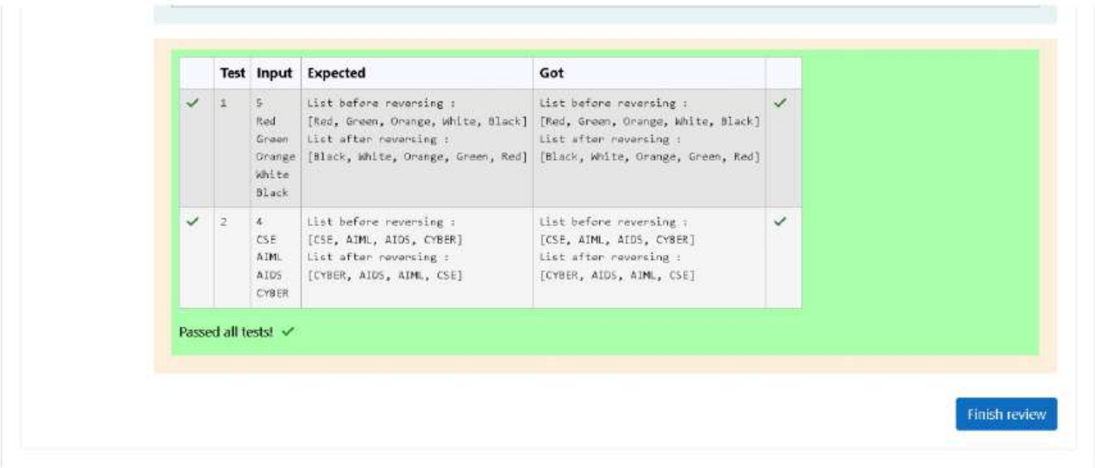
Sample output

```
List before reversing :
[Red, Green, Orange, White, Black]
List after reversing :
[Black, White, Orange, Green, Red]
```

Answer: (penalty regime: 0 %)

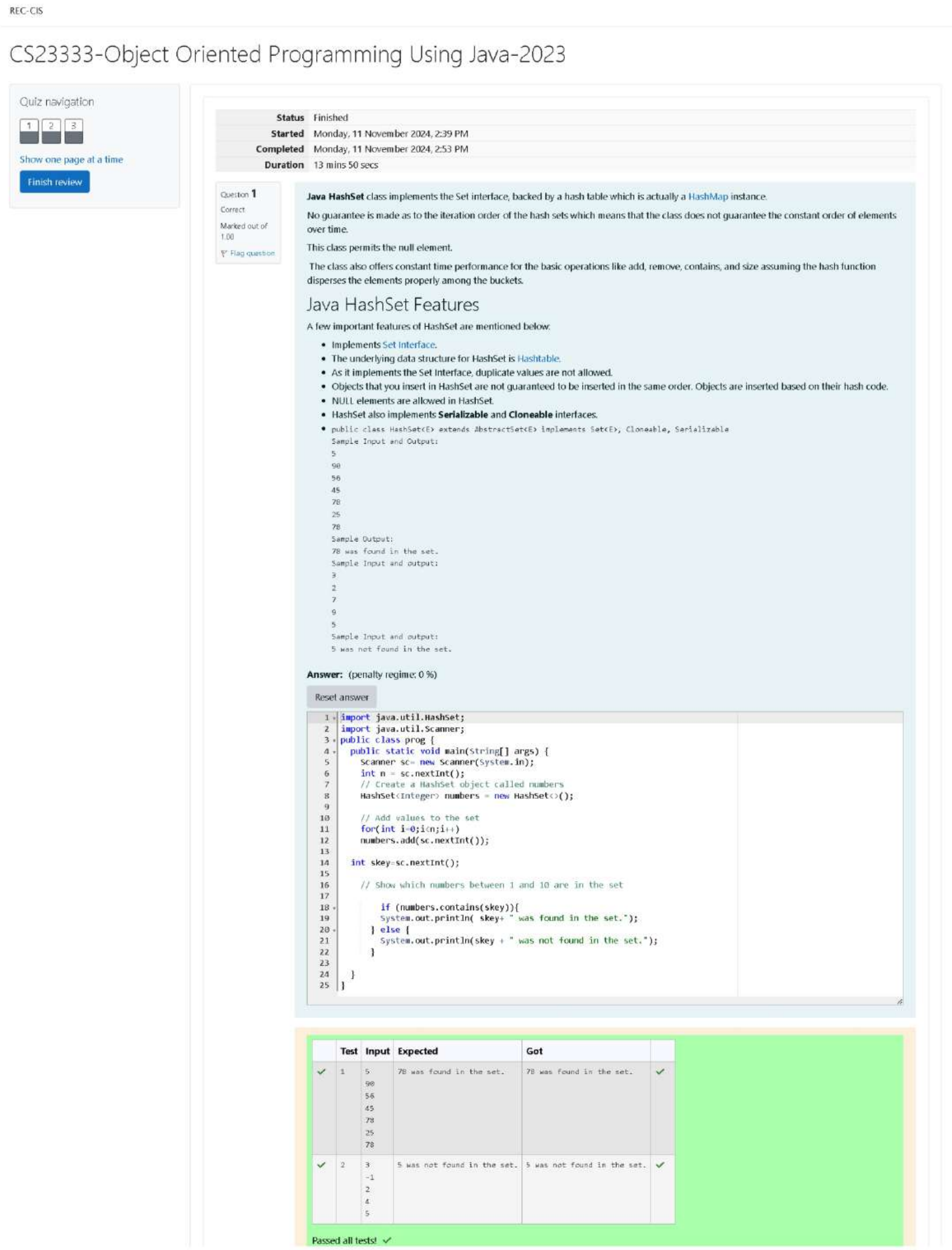


```
import java.util.*;
public class array{
    public static void main(String args[]){
        Scanner scn=new Scanner(System.in);
        int n=scn.nextInt();
        scn.nextLine();
        ArrayList<String> list=new ArrayList<>();
        for(int i=0;i<n;i++){
            list.add(scn.nextLine());
        }
        if(!list.isEmpty()){
            System.out.println("List before reversing :"+"\n"+list);
            Collections.reverse(list);
            System.out.println("List after reversing :\n"+list);
        }

    }
}
```



	Test	Input	Expected	Got	
✓	1	5 Red Green Orange White Black	List before reversing : [Red, Green, Orange, White, Black] List after reversing : [Black, White, Orange, Green, Red]	List before reversing : [Red, Green, Orange, White, Black] List after reversing : [Black, White, Orange, Green, Red]	✓
✓	2	4 CSE AIML AIDS CYBER	List before reversing : [CSE, AIML, AIDS, CYBER] List after reversing : [CYBER, AIDS, AIML, CSE]	List before reversing : [CSE, AIML, AIDS, CYBER] List after reversing : [CYBER, AIDS, AIML, CSE]	✓

Passed all tests! ✓



REC-CIS

CS23333-Object Oriented Programming Using Java-2023

Status	Finished
Started	Monday, 11 November 2024, 2:39 PM
Completed	Monday, 11 November 2024, 2:53 PM
Duration	13 mins 50 secs

Question **1**

Correct

Marked out of 1.00

Java HashSet class implements the Set interface, backed by a hash table which is actually a HashMap instance.

No guarantee is made as to the iteration order of the hash sets which means that the class does not guarantee the constant order of elements over time.

This class permits the null element.

The class also offers constant time performance for the basic operations like add, remove, contains, and size assuming the hash function disperses the elements properly among the buckets.

Java HashSet Features

A few important features of HashSet are mentioned below:

- Implements Set Interface.
- The underlying data structure for HashSet is Hashtable.
- As it implements the Set Interface, duplicate values are not allowed.
- Objects that you insert in HashSet are not guaranteed to be inserted in the same order. Objects are inserted based on their hash code.
- NULL elements are allowed in HashSet.
- HashSet also implements **Serializable** and **Cloneable** interfaces.
- `public class HashSet<E> extends AbstractSet<E> implements Set<E>, Cloneable, Serializable`

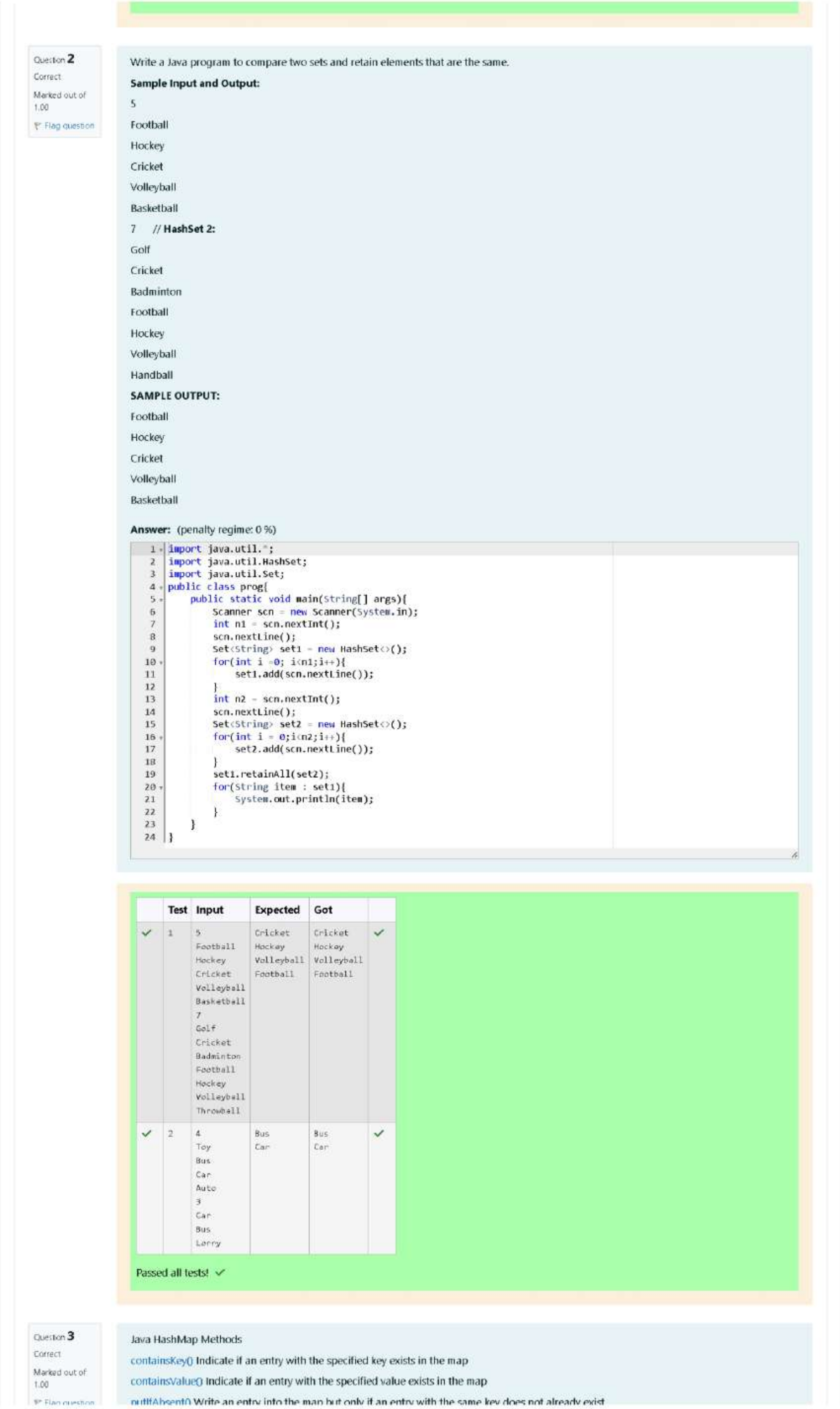
```
Sample Input and Output:
5
90
56
45
78
25
78
Sample Output:
78 was found in the set.
Sample Input and output:
3
2
7
9
5
Sample Input and output:
5 was not found in the set.
```

Answer: (penalty regime: 0 %)

```java
import java.util.HashSet;
import java.util.Scanner;
public class prog {
  public static void main(String[] args) {
    Scanner sc= new Scanner(System.in);
    int n = sc.nextInt();
    // Create a HashSet object called numbers
    HashSet<Integer> numbers = new HashSet<>();

    // Add values to the set
    for(int i=0;i<n;i++)
    numbers.add(sc.nextInt());

  int skey=sc.nextInt();

    // Show which numbers between 1 and 10 are in the set

        if (numbers.contains(skey)){
        System.out.println( skey+ " was found in the set.");
      } else {
        System.out.println(skey + " was not found in the set.");
      }

  }
}
```

	Test	Input	Expected	Got	
✓	1	5 90 56 45 78 25 78	78 was found in the set.	78 was found in the set.	✓
✓	2	3 -1 2 4 5	5 was not found in the set.	5 was not found in the set.	✓

Passed all tests! ✓



Question **2**

Correct

Marked out of 1.00

Write a Java program to compare two sets and retain elements that are the same.

Sample Input and Output:

5

Football

Hockey

Cricket

Volleyball

Basketball

7 // **HashSet 2:**

Golf

Cricket

Badminton

Football

Hockey

Volleyball

Handball

SAMPLE OUTPUT:

Football

Hockey

Cricket

Volleyball

Basketball

Answer: (penalty regime: 0 %)

```java
import java.util.*;
import java.util.HashSet;
import java.util.Set;
public class prog{
    public static void main(String[] args){
        Scanner scn = new Scanner(System.in);
        int n1 = scn.nextInt();
        scn.nextLine();
        Set<String> set1 = new HashSet<>();
        for(int i =0; i<n1;i++){
            set1.add(scn.nextLine());
        }
        int n2 = scn.nextInt();
        scn.nextLine();
        Set<String> set2 = new HashSet<>();
        for(int i = 0;i<n2;i++){
            set2.add(scn.nextLine());
        }
        set1.retainAll(set2);
        for(String item : set1){
            System.out.println(item);
        }
    }
}
```

	Test	Input	Expected	Got	
✓	1	5 Football Hockey Cricket Volleyball Basketball 7 Golf Cricket Badminton Football Hockey Volleyball Throwball	Cricket Hockey Volleyball Football	Cricket Hockey Volleyball Football	✓
✓	2	4 Toy Bus Car Auto 3 Car Bus Lorry	Bus Car	Bus Car	✓

Passed all tests! ✓

Question **3**

Correct

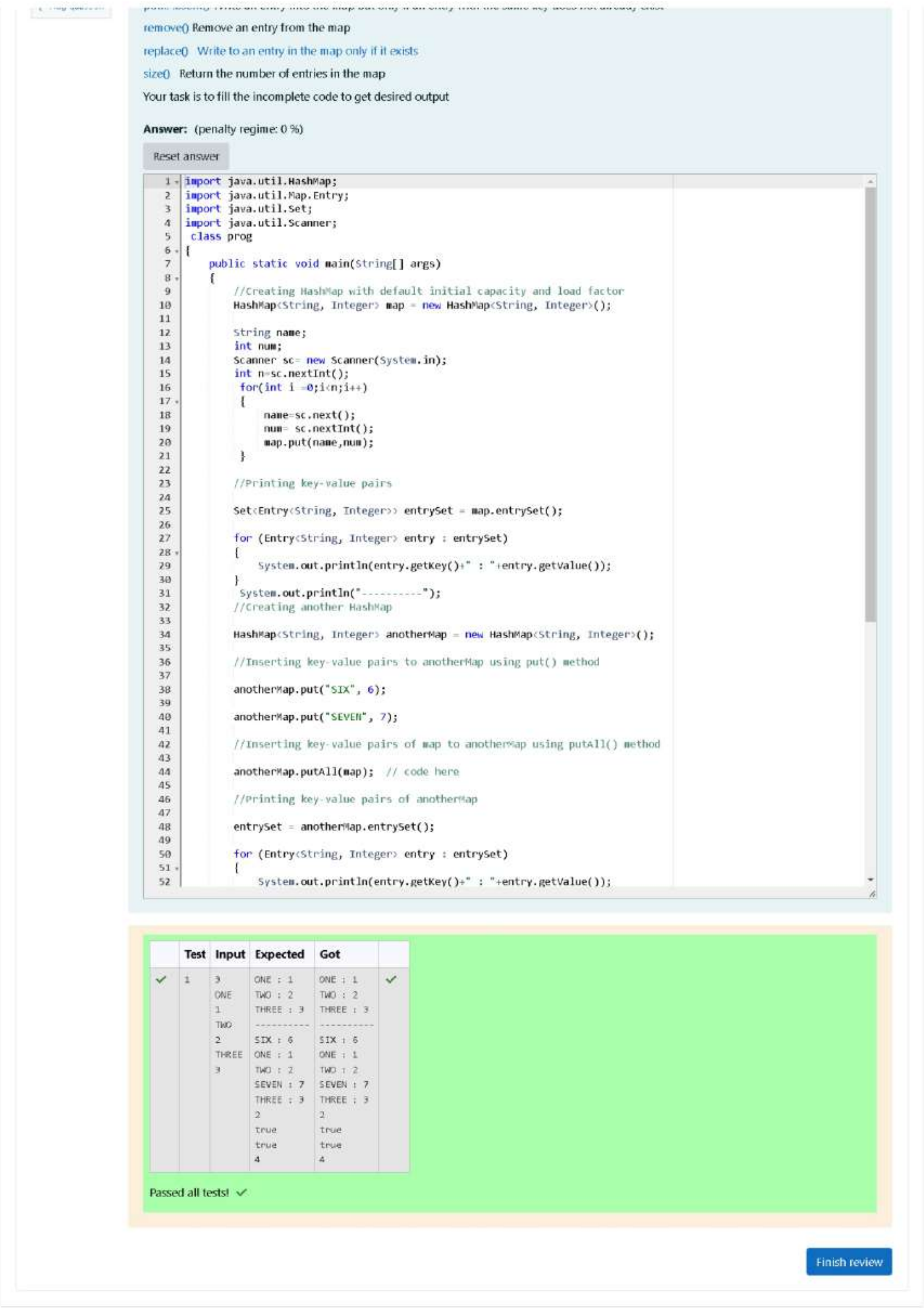
Marked out of 1.00

Java HashMap Methods

containsKey() Indicate if an entry with the specified key exists in the map

containsValue() Indicate if an entry with the specified value exists in the map

putIfAbsent() Write an entry into the map but only if an entry with the same key does not already exist



putIfAbsent() Write an entry into the map but only if an entry with the same key does not already exist

remove() Remove an entry from the map

replace() Write to an entry in the map only if it exists

size() Return the number of entries in the map

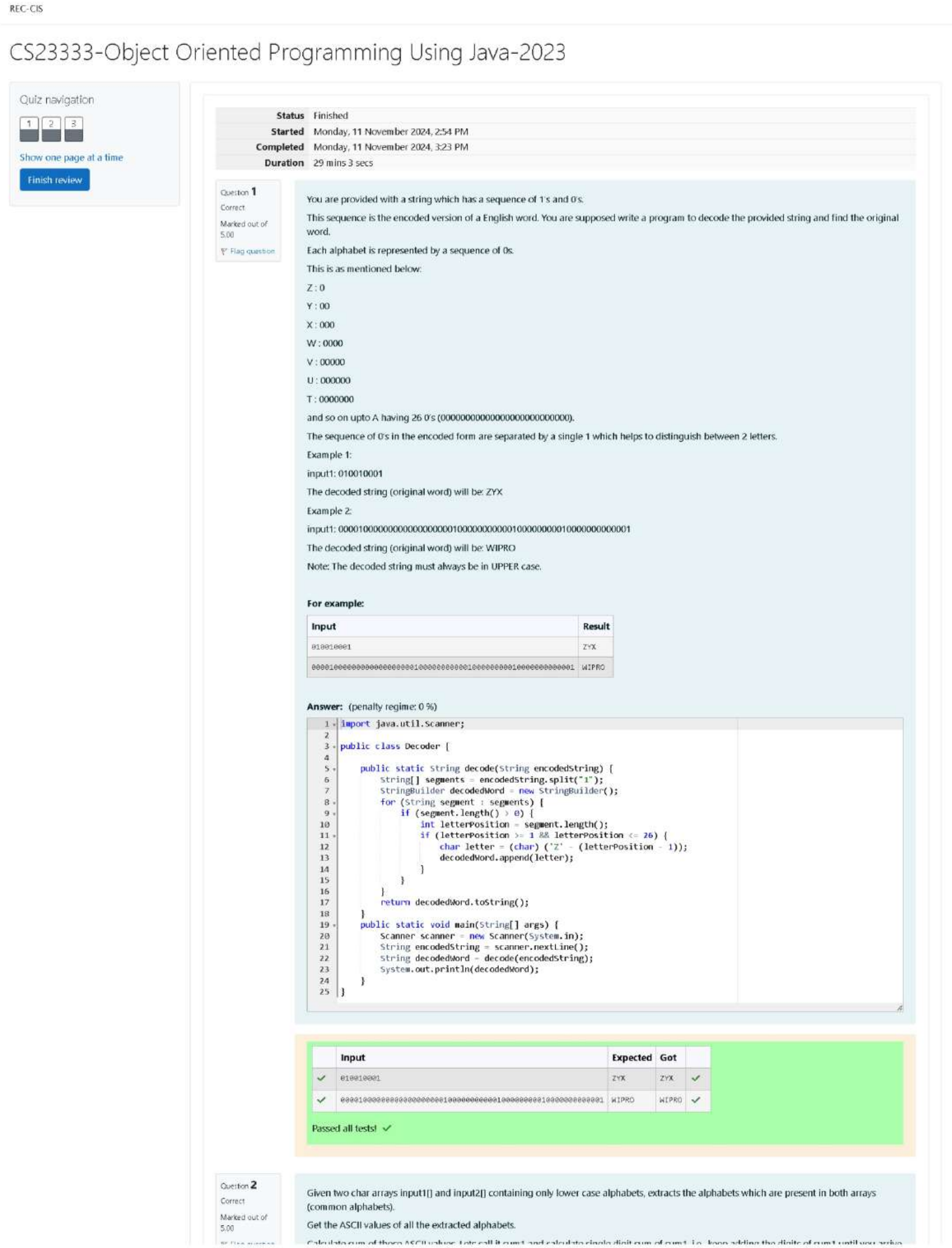
Your task is to fill the incomplete code to get desired output

Answer: (penalty regime: 0 %)

```java
import java.util.HashMap;
import java.util.Map.Entry;
import java.util.Set;
import java.util.Scanner;
 class prog
{
    public static void main(String[] args)
    {
        //Creating HashMap with default initial capacity and load factor
        HashMap<String, Integer> map = new HashMap<String, Integer>();

        String name;
        int num;
        Scanner sc= new Scanner(System.in);
        int n=sc.nextInt();
         for(int i =0;i<n;i++)
         {
             name=sc.next();
             num= sc.nextInt();
             map.put(name,num);
         }

        //Printing key-value pairs

        Set<Entry<String, Integer>> entrySet = map.entrySet();

        for (Entry<String, Integer> entry : entrySet)
        {
            System.out.println(entry.getKey()+" : "+entry.getValue());
        }
         System.out.println("----------");
        //Creating another HashMap

        HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();

        //Inserting key-value pairs to anotherMap using put() method

        anotherMap.put("SIX", 6);

        anotherMap.put("SEVEN", 7);

        //Inserting key-value pairs of map to anotherMap using putAll() method

        anotherMap.putAll(map);  // code here

        //Printing key-value pairs of anotherMap

        entrySet = anotherMap.entrySet();

        for (Entry<String, Integer> entry : entrySet)
        {
            System.out.println(entry.getKey()+" : "+entry.getValue());
```

	Test	Input	Expected	Got	
✓	1	3 ONE 1 TWO 2 THREE 3	ONE : 1 TWO : 2 THREE : 3 ---------- SIX : 6 ONE : 1 TWO : 2 SEVEN : 7 THREE : 3 2 true true 4	ONE : 1 TWO : 2 THREE : 3 ---------- SIX : 6 ONE : 1 TWO : 2 SEVEN : 7 THREE : 3 2 true true 4	✓

Passed all tests! ✓



REC-CIS

CS23333-Object Oriented Programming Using Java-2023

Status	Finished
Started	Monday, 11 November 2024, 2:54 PM
Completed	Monday, 11 November 2024, 3:23 PM
Duration	29 mins 3 secs

Question **1**

Correct

Marked out of 5.00

You are provided with a string which has a sequence of 1's and 0's.

This sequence is the encoded version of a English word. You are supposed write a program to decode the provided string and find the original word.

Each alphabet is represented by a sequence of 0s.

This is as mentioned below:

Z : 0

Y : 00

X : 000

W : 0000

V : 00000

U : 000000

T : 0000000

and so on upto A having 26 0's (00000000000000000000000000).

The sequence of 0's in the encoded form are separated by a single 1 which helps to distinguish between 2 letters.

Example 1:

input1: 010010001

The decoded string (original word) will be: ZYX

Example 2:

input1: 0000100000000000000000010000000000010000000001000000000000001

The decoded string (original word) will be: WIPRO

Note: The decoded string must always be in UPPER case.

For example:

Input	Result
010010001	ZYX
0000100000000000000000010000000000010000000001000000000000001	WIPRO

Answer: (penalty regime: 0 %)

```java
import java.util.Scanner;

public class Decoder {

    public static String decode(String encodedString) {
        String[] segments = encodedString.split("1");
        StringBuilder decodedWord = new StringBuilder();
        for (String segment : segments) {
            if (segment.length() > 0) {
                int letterPosition = segment.length();
                if (letterPosition >= 1 && letterPosition <= 26) {
                    char letter = (char) ('Z' - (letterPosition - 1));
                    decodedWord.append(letter);
                }
            }
        }
        return decodedWord.toString();
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String encodedString = scanner.nextLine();
        String decodedWord = decode(encodedString);
        System.out.println(decodedWord);
    }
}
```

	Input	Expected	Got	
✓	010010001	ZYX	ZYX	✓
✓	0000100000000000000000010000000000010000000001000000000000001	WIPRO	WIPRO	✓

Passed all tests! ✓

Question **2**

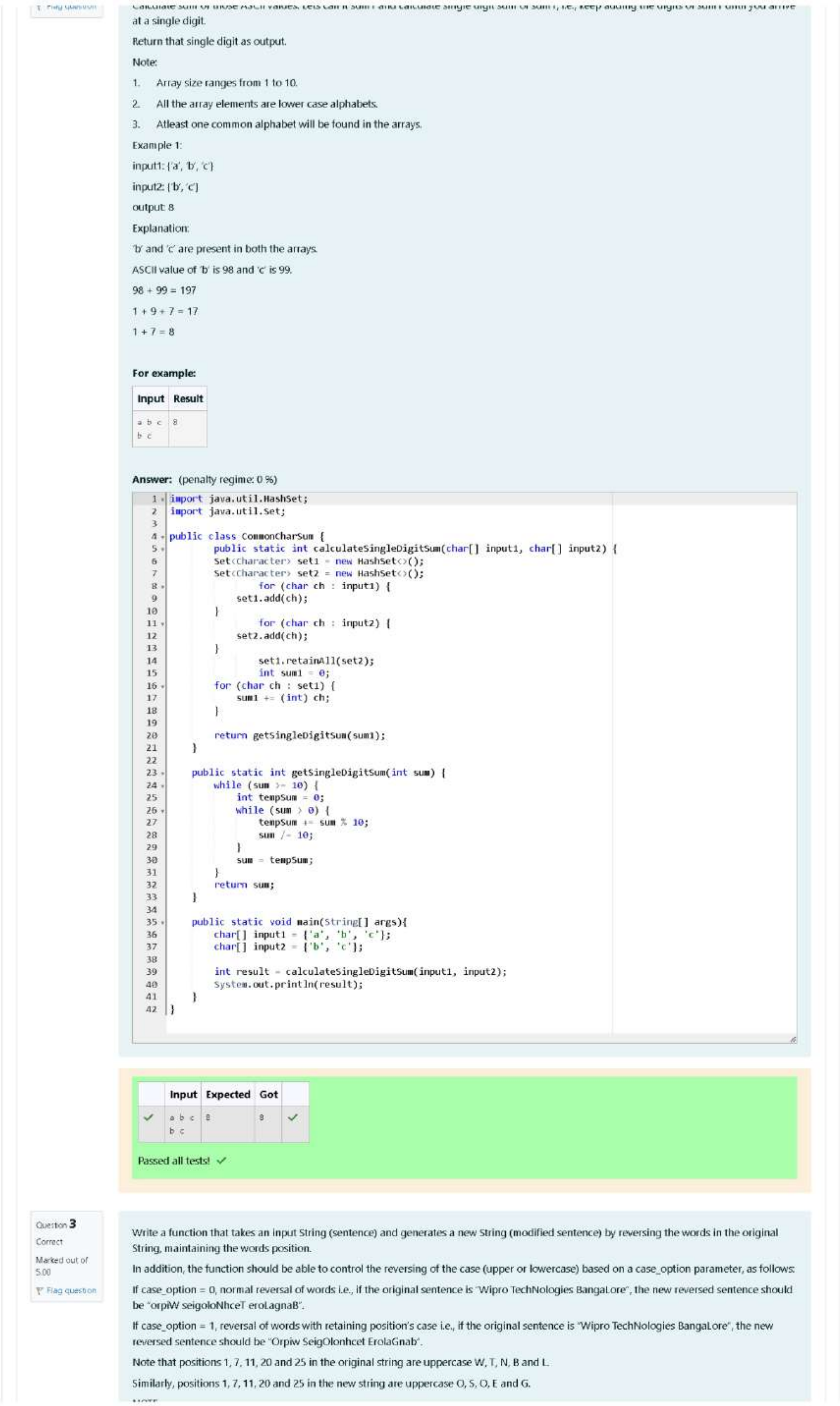
Correct

Marked out of 5.00

Given two char arrays input1[] and input2[] containing only lower case alphabets, extracts the alphabets which are present in both arrays (common alphabets).

Get the ASCII values of all the extracted alphabets.

Calculate sum of those ASCII values. Lets call it sum1 and calculate single digit sum of sum1, i.e., keep adding the digits of sum1 until you arrive



Calculate sum of those ASCII values. Lets call it sum1 and calculate single digit sum of sum1, i.e., keep adding the digits of sum1 until you arrive at a single digit.

Return that single digit as output.

Note:

1. Array size ranges from 1 to 10.
2. All the array elements are lower case alphabets.
3. Atleast one common alphabet will be found in the arrays.

Example 1:

input1: {'a', 'b', 'c'}

input2: {'b', 'c'}

output: 8

Explanation:

'b' and 'c' are present in both the arrays.

ASCII value of 'b' is 98 and 'c' is 99.

98 + 99 = 197

1 + 9 + 7 = 17

1 + 7 = 8

For example:

Input	Result
a b c b c	8

Answer: (penalty regime: 0 %)

```
import java.util.HashSet;
import java.util.Set;

public class CommonCharSum {
        public static int calculateSingleDigitSum(char[] input1, char[] input2) {
        Set<Character> set1 = new HashSet<>();
        Set<Character> set2 = new HashSet<>();
            for (char ch : input1) {
          set1.add(ch);
        }
            for (char ch : input2) {
          set2.add(ch);
        }
            set1.retainAll(set2);
            int sum1 = 0;
        for (char ch : set1) {
            sum1 += (int) ch;
        }

        return getSingleDigitSum(sum1);
    }

    public static int getSingleDigitSum(int sum) {
        while (sum >= 10) {
            int tempSum = 0;
            while (sum > 0) {
                tempSum += sum % 10;
                sum /= 10;
            }
            sum = tempSum;
        }
        return sum;
    }

    public static void main(String[] args){
        char[] input1 = {'a', 'b', 'c'};
        char[] input2 = {'b', 'c'};

        int result = calculateSingleDigitSum(input1, input2);
        System.out.println(result);
    }
}
```

	Input	Expected	Got	
✓	a b c b c	8	8	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

Flag question

Write a function that takes an input String (sentence) and generates a new String (modified sentence) by reversing the words in the original String, maintaining the words position.

In addition, the function should be able to control the reversing of the case (upper or lowercase) based on a case_option parameter, as follows:

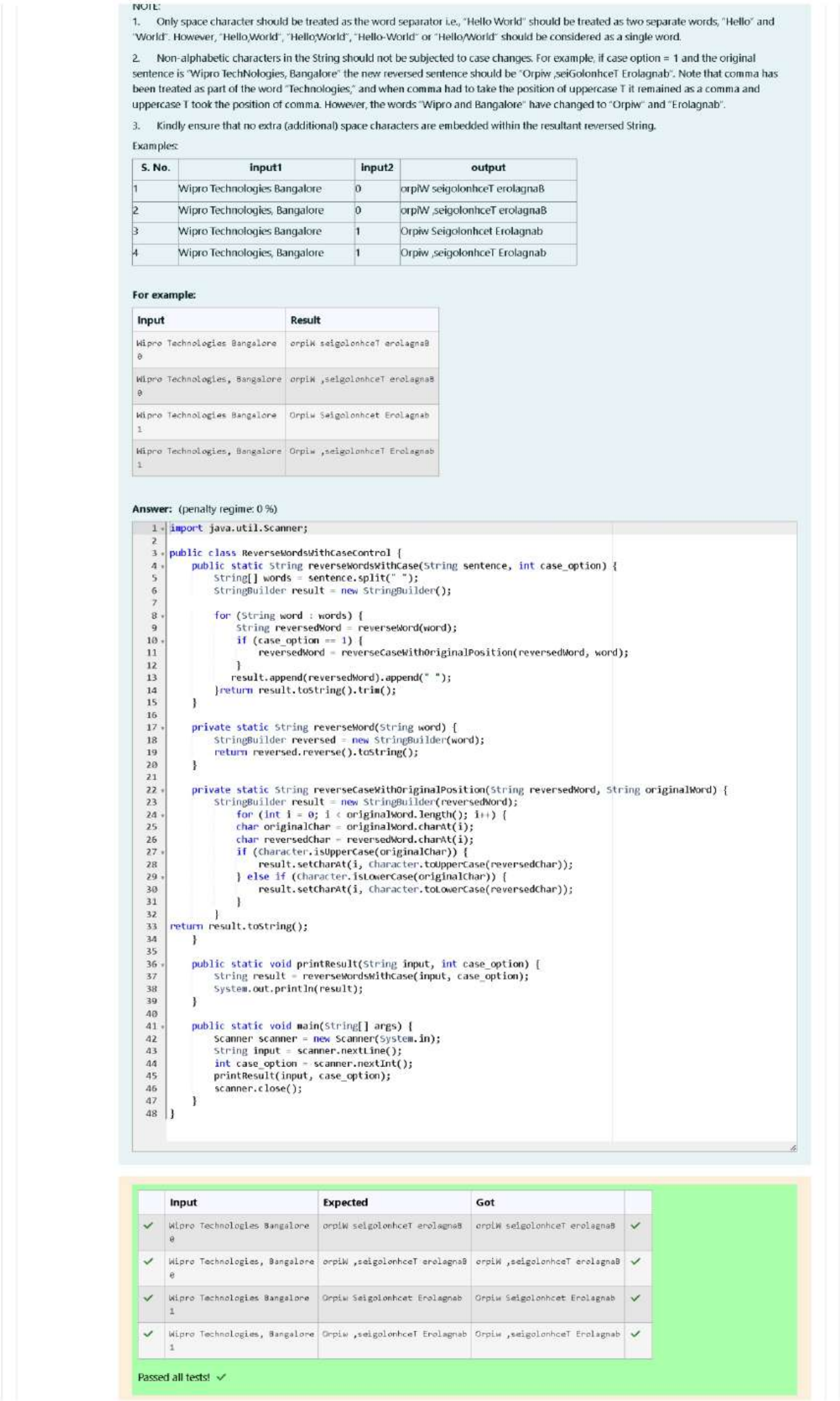
If case_option = 0, normal reversal of words i.e., if the original sentence is "Wipro TechNologies BangaLore", the new reversed sentence should be "orpiW seigoloNhceT eroLagnaB".

If case_option = 1, reversal of words with retaining position's case i.e., if the original sentence is "Wipro TechNologies BangaLore", the new reversed sentence should be "Orpiw SeigOlonhcet ErolaGnab".

Note that positions 1, 7, 11, 20 and 25 in the original string are uppercase W, T, N, B and L.

Similarly, positions 1, 7, 11, 20 and 25 in the new string are uppercase O, S, O, E and G.

NOTE



NOTE:

1. Only space character should be treated as the word separator i.e., "Hello World" should be treated as two separate words, "Hello" and "World". However, "Hello,World", "Hello;World", "Hello-World" or "Hello/World" should be considered as a single word.

2. Non-alphabetic characters in the String should not be subjected to case changes. For example, if case option = 1 and the original sentence is "Wipro TechNologies, Bangalore" the new reversed sentence should be "Orpiw ,seiGolonhceT Erolagnab". Note that comma has been treated as part of the word "Technologies," and when comma had to take the position of uppercase T it remained as a comma and uppercase T took the position of comma. However, the words "Wipro and Bangalore" have changed to "Orpiw" and "Erolagnab".

3. Kindly ensure that no extra (additional) space characters are embedded within the resultant reversed String.

Examples:

S. No.	input1	input2	output
1	Wipro Technologies Bangalore	0	orpiW seigolonhceT erolagnaB
2	Wipro Technologies, Bangalore	0	orpiW ,seigolonhceT erolagnaB
3	Wipro Technologies Bangalore	1	Orpiw Seigolonhcet Erolagnab
4	Wipro Technologies, Bangalore	1	Orpiw ,seigolonhceT Erolagnab

For example:

Input	Result
Wipro Technologies Bangalore 0	orpiW seigolonhceT erolagnaB
Wipro Technologies, Bangalore 0	orpiW ,seigolonhceT erolagnaB
Wipro Technologies Bangalore 1	Orpiw Seigolonhcet Erolagnab
Wipro Technologies, Bangalore 1	Orpiw ,seigolonhceT Erolagnab

Answer: (penalty regime: 0 %)

```java
import java.util.Scanner;

public class ReverseWordsWithCaseControl {
    public static String reverseWordsWithCase(String sentence, int case_option) {
        String[] words = sentence.split(" ");
        StringBuilder result = new StringBuilder();

        for (String word : words) {
            String reversedWord = reverseWord(word);
            if (case_option == 1) {
                reversedWord = reverseCaseWithOriginalPosition(reversedWord, word);
            }
           result.append(reversedWord).append(" ");
        }return result.toString().trim();
    }

    private static String reverseWord(String word) {
        StringBuilder reversed = new StringBuilder(word);
        return reversed.reverse().toString();
    }

    private static String reverseCaseWithOriginalPosition(String reversedWord, String originalWord) {
        StringBuilder result = new StringBuilder(reversedWord);
            for (int i = 0; i < originalWord.length(); i++) {
            char originalChar = originalWord.charAt(i);
            char reversedChar = reversedWord.charAt(i);
            if (Character.isUpperCase(originalChar)) {
                result.setCharAt(i, Character.toUpperCase(reversedChar));
            } else if (Character.isLowerCase(originalChar)) {
                result.setCharAt(i, Character.toLowerCase(reversedChar));
            }
        }
return result.toString();
    }

    public static void printResult(String input, int case_option) {
        String result = reverseWordsWithCase(input, case_option);
        System.out.println(result);
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String input = scanner.nextLine();
        int case_option = scanner.nextInt();
        printResult(input, case_option);
        scanner.close();
    }
}
```

	Input	Expected	Got	
✓	Wipro Technologies Bangalore 0	orpiW seigolonhceT erolagnaB	orpiW seigolonhceT erolagnaB	✓
✓	Wipro Technologies, Bangalore 0	orpiW ,seigolonhceT erolagnaB	orpiW ,seigolonhceT erolagnaB	✓
✓	Wipro Technologies Bangalore 1	Orpiw Seigolonhcet Erolagnab	Orpiw Seigolonhcet Erolagnab	✓
✓	Wipro Technologies, Bangalore 1	Orpiw ,seigolonhceT Erolagnab	Orpiw ,seigolonhceT Erolagnab	✓

Passed all tests! ✓

Finish review

MINI PROJECT

ONLINE MOVIE TICKET BOOKING SYSTEM

CODE :

```
package movieticketbooking;

import javax.swing.*;

import java.awt.*;

import java.util.List;

import java.util.Map;


public class MovieTicketBookingSystem {

  private static DatabaseOperation db = new DatabaseOperation();

  private static int loggedInUserID = -1; // User session management

  public static void main(String[] args) {

    showMainMenu();

  }

  // Main Menu

  private static void showMainMenu() {

    JFrame mainMenu = new JFrame("Movie Ticket Booking System");

    mainMenu.setSize(400, 300);

    mainMenu.setLayout(new GridLayout(3, 1));


    JButton adminButton = new JButton("Admin Login");

    JButton userButton = new JButton("User Login");

    JButton exitButton = new JButton("Exit");


    mainMenu.add(adminButton);

    mainMenu.add(userButton);

    mainMenu.add(exitButton);


    adminButton.addActionListener(e -> {
        mainMenu.dispose();
        showAdminLogin();
    });
    userButton.addActionListener(e -> {
        mainMenu.dispose();
        showUserLogin();
    });
    exitButton.addActionListener(e -> System.exit(0));
    mainMenu.setVisible(true);
}

// ADMIN FUNCTIONS
private static void showAdminLogin() {
    JFrame adminLogin = new JFrame("Admin Login");
    adminLogin.setSize(300, 200);
    adminLogin.setLayout(new GridLayout(3, 2));

    JLabel usernameLabel = new JLabel("Admin Name:");
    JTextField usernameField = new JTextField();
    JLabel passwordLabel = new JLabel("Password:");
    JPasswordField passwordField = new JPasswordField();

    JButton loginButton = new JButton("Login");
    JButton backButton = new JButton("Back");

    adminLogin.add(usernameLabel);
    adminLogin.add(usernameField);
    adminLogin.add(passwordLabel);
    adminLogin.add(passwordField);
```

```
  adminLogin.add(loginButton);
  adminLogin.add(backButton);

  loginButton.addActionListener(e -> {
    String adminName = usernameField.getText();
    String password = new String(passwordField.getPassword());

    String sql = "SELECT Admin_name, Password FROM admin WHERE Admin_name = ?";
    Map<String, Object> admin = db.validatePass(sql, adminName);

    if (admin != null && password.equals(admin.get("Password"))) {
      JOptionPane.showMessageDialog(adminLogin, "Login successful!");
      adminLogin.dispose();
      showAdminDashboard();
    } else {
      JOptionPane.showMessageDialog(adminLogin, "Invalid credentials!");
    }
  });

  backButton.addActionListener(e -> {
    adminLogin.dispose();
    showMainMenu();
  });

  adminLogin.setVisible(true);
}
```

```
private static void showAdminDashboard() {
  JFrame adminDashboard = new JFrame("Admin Dashboard");
  adminDashboard.setSize(400, 400);
  adminDashboard.setLayout(new GridLayout(5, 1));

  JButton addMovieButton = new JButton("Add Movies");
  JButton addTheaterButton = new JButton("Add Theaters");
  JButton addShowtimeButton = new JButton("Add Showtimes");
  JButton viewMoviesButton = new JButton("View Movies and Showtimes");
  JButton logoutButton = new JButton("Logout");

  adminDashboard.add(addMovieButton);
  adminDashboard.add(addTheaterButton);
  adminDashboard.add(addShowtimeButton);
  adminDashboard.add(viewMoviesButton);
  adminDashboard.add(logoutButton);

  addMovieButton.addActionListener(e -> {
    adminDashboard.dispose();
    addMovie();
  });

  addTheaterButton.addActionListener(e -> {
    adminDashboard.dispose();
    addTheater();
  });

  addShowtimeButton.addActionListener(e -> {
    adminDashboard.dispose();
    addShowtime();
  });

  viewMoviesButton.addActionListener(e -> {
    adminDashboard.dispose();
    viewMoviesAndShowtimes();
  });

  logoutButton.addActionListener(e -> {
    adminDashboard.dispose();
    showMainMenu();
  });

  adminDashboard.setVisible(true);
}

private static void addMovie() {
  JFrame addMovieFrame = new JFrame("Add Movie");
  addMovieFrame.setSize(300, 400);
  addMovieFrame.setLayout(new GridLayout(6, 2));
  JLabel movieidLabel = new JLabel("Movie ID:");
  JTextField movieidField = new JTextField();
  JLabel titleLabel = new JLabel("Movie Title:");
  JTextField titleField = new JTextField();
  JLabel genreLabel = new JLabel("Genre:");
  JTextField genreField = new JTextField();
  JLabel durationLabel = new JLabel("Duration (mins):");
  JTextField durationField = new JTextField();
  JLabel synopsisLabel = new JLabel("Synopsis:");
```

```
JTextField synopsisField = new JTextField();
JLabel ratingLabel = new JLabel("Rating (0-10):");
JTextField ratingField = new JTextField();

JButton submitButton = new JButton("Submit");
JButton cancelButton = new JButton("Cancel");
addMovieFrame.add(movieidLabel);
addMovieFrame.add(movieidField);
addMovieFrame.add(titleLabel);
addMovieFrame.add(titleField);
addMovieFrame.add(genreLabel);
addMovieFrame.add(genreField);
addMovieFrame.add(durationLabel);
addMovieFrame.add(durationField);
addMovieFrame.add(synopsisLabel);
addMovieFrame.add(synopsisField);
addMovieFrame.add(ratingLabel);
addMovieFrame.add(ratingField);
addMovieFrame.add(submitButton);
addMovieFrame.add(cancelButton);

submitButton.addActionListener(e -> {
    int movie_id;
   String title = titleField.getText();
   String genre = genreField.getText();
   int duration;
   double rating;
   try {
    movie_id = Integer.parseInt(movieidField.getText());
```

```
        duration = Integer.parseInt(durationField.getText());
        rating = Double.parseDouble(ratingField.getText());
      } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(addMovieFrame, "Invalid duration or rating.
Please enter numbers.");
        return;
      }
      String synopsis = synopsisField.getText();


      String sql = "INSERT INTO movies (movie_ID, title, genre, Duration, Synopsis, rating)
VALUES (?, ?, ?, ?, ?, ?)";
      Object[] values = {movie_id,title, genre, duration, synopsis, rating};
      int rowsAffected = db.executeUpdate(sql, values);


      if (rowsAffected > 0) {
        JOptionPane.showMessageDialog(addMovieFrame, "Movie added successfully!");
      } else {
        JOptionPane.showMessageDialog(addMovieFrame, "Failed to add the movie.");
      }
      addMovieFrame.dispose();
      showAdminDashboard();
    });


    cancelButton.addActionListener(e -> {
      addMovieFrame.dispose();
      showAdminDashboard();
    });


    addMovieFrame.setVisible(true);
  }
```

```
private static void addTheater() {
  JFrame addTheaterFrame = new JFrame("Add Theater");
  addTheaterFrame.setSize(300, 400);
  addTheaterFrame.setLayout(new GridLayout(5, 2));

  JLabel idLabel = new JLabel("Theater ID:");
  JTextField idField = new JTextField();
  JLabel nameLabel = new JLabel("Theater Name:");
  JTextField nameField = new JTextField();
  JLabel capacityLabel = new JLabel("Seating Capacity:");
  JTextField capacityField = new JTextField();
  JLabel locationLabel = new JLabel("Location:");
  JTextField locationField = new JTextField();
  JLabel screenLabel = new JLabel("Screen ID:");
  JTextField screenField = new JTextField();
  JLabel foodLabel = new JLabel("Food Add On:");
  JTextField foodField = new JTextField();

  JButton submitButton = new JButton("Submit");
  JButton cancelButton = new JButton("Cancel");

  addTheaterFrame.add(idLabel);
  addTheaterFrame.add(idField);
  addTheaterFrame.add(nameLabel);
  addTheaterFrame.add(nameField);
  addTheaterFrame.add(capacityLabel);
  addTheaterFrame.add(capacityField);
  addTheaterFrame.add(locationLabel);
```

```
    addTheaterFrame.add(locationField);
    addTheaterFrame.add(screenLabel);
    addTheaterFrame.add(screenField);
    addTheaterFrame.add(foodLabel);
    addTheaterFrame.add(foodField);
    addTheaterFrame.add(submitButton);
    addTheaterFrame.add(cancelButton);

    submitButton.addActionListener(e -> {
      String name = nameField.getText();
      int capacity;
      try {
        capacity = Integer.parseInt(capacityField.getText());
      } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(addTheaterFrame, "Invalid capacity. Please enter
a number.");
        return;
      }
      String location = locationField.getText();
      String foodaddon=foodField.getText();
      int theatreID,screenID;
      try {
       theatreID = Integer.parseInt(screenField.getText());
        screenID = Integer.parseInt(screenField.getText());
      } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(addTheaterFrame, "Invalid screen ID. Please
enter a number.");
        return;
      }
      String sql = "INSERT INTO theatre (theatre_id,theatre_name, seating_capacity,
location, screen_id,food_add_on) VALUES (?,?,?, ?, ?, ?)";
      Object[] values = {theatreID, name, capacity, location, screenID,foodaddon};
      int rowsAffected = db.executeUpdate(sql, values);

      if (rowsAffected > 0) {
        JOptionPane.showMessageDialog(addTheaterFrame, "Theater added
successfully!");
      } else {
        JOptionPane.showMessageDialog(addTheaterFrame, "Failed to add the theater.");
      }
      addTheaterFrame.dispose();
      showAdminDashboard();
    });

    cancelButton.addActionListener(e -> {
      addTheaterFrame.dispose();
      showAdminDashboard();
    });

    addTheaterFrame.setVisible(true);
  }

  private static void addShowtime() {
    JFrame addShowtimeFrame = new JFrame("Add Showtime");
    addShowtimeFrame.setSize(300, 400);
    addShowtimeFrame.setLayout(new GridLayout(6, 2));

    JLabel showtimeIDLabel = new JLabel("Showtime ID:");
    JTextField showtimeIDField = new JTextField();
```

```
JLabel movieIDLabel = new JLabel("Movie ID:");
JTextField movieIDField = new JTextField();
JLabel theaterIDLabel = new JLabel("Theater ID:");
JTextField theaterIDField = new JTextField();
JLabel screenIDLabel = new JLabel("Screen ID:");
JTextField screenIDField = new JTextField();
JLabel dayLabel = new JLabel("Day:");
JTextField dayField = new JTextField();
JLabel dateLabel = new JLabel("Date (YYYY-MM-DD):");
JTextField dateField = new JTextField();

JButton submitButton = new JButton("Submit");
JButton cancelButton = new JButton("Cancel");

addShowtimeFrame.add(showtimeIDLabel);
addShowtimeFrame.add(showtimeIDField);
addShowtimeFrame.add(movieIDLabel);
addShowtimeFrame.add(movieIDField);
addShowtimeFrame.add(theaterIDLabel);
addShowtimeFrame.add(theaterIDField);
addShowtimeFrame.add(screenIDLabel);
addShowtimeFrame.add(screenIDField);
addShowtimeFrame.add(dayLabel);
addShowtimeFrame.add(dayField);
addShowtimeFrame.add(dateLabel);
addShowtimeFrame.add(dateField);
addShowtimeFrame.add(submitButton);
addShowtimeFrame.add(cancelButton);
```

```
    submitButton.addActionListener(e -> {
      int showtimeID, movieID, theaterID, screenID;
      String day = dayField.getText();
      String date = dateField.getText();

      try {
       showtimeID = Integer.parseInt(showtimeIDField.getText());
        movieID = Integer.parseInt(movieIDField.getText());
        theaterID = Integer.parseInt(theaterIDField.getText());
        screenID = Integer.parseInt(screenIDField.getText());
      } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(addShowtimeFrame, "Invalid Movie ID, Theater
ID, or Screen ID.");
        return;
      }

      String sql = "INSERT INTO showtimes (showtime_id, movie_ID, theatre_id, screen_id,
day, date) VALUES (?,?, ?, ?, ?, ?)";
      Object[] values = {showtimeID,movieID, theaterID, screenID, day, date};
      int rowsAffected = db.executeUpdate(sql, values);

      if (rowsAffected > 0) {
        JOptionPane.showMessageDialog(addShowtimeFrame, "Showtime added
successfully!");
      } else {
        JOptionPane.showMessageDialog(addShowtimeFrame, "Failed to add the
showtime.");
      }
      addShowtimeFrame.dispose();
      showAdminDashboard();
    });
    cancelButton.addActionListener(e -> {
      addShowtimeFrame.dispose();
      showAdminDashboard();
    });

    addShowtimeFrame.setVisible(true);
  }

  private static void viewMoviesAndShowtimes() {
    JFrame viewMoviesFrame = new JFrame("Movies and Showtimes");
    viewMoviesFrame.setSize(600, 600);
    viewMoviesFrame.setLayout(new GridLayout(0, 1));

    List<Map<String, Object>> movies = db.getRecords("SELECT * FROM movies");
    for (Map<String, Object> movie : movies) {
      String movieDetails = "Movie ID: " + movie.get("movie_ID") +
                  ", Title: " + movie.get("title") +
                  ", Genre: " + movie.get("genre") +
                  ", Duration: " + movie.get("Duration") +
                  ", Rating: " + movie.get("rating");
      JLabel movieLabel = new JLabel(movieDetails);
      viewMoviesFrame.add(movieLabel);

      int movieID = (int) movie.get("movie_ID");
      List<Map<String, Object>> showtimes = db.getRecords("SELECT * FROM showtimes
WHERE movie_ID = " + movieID);
      for (Map<String, Object> showtime : showtimes) {
        String showtimeDetails = "    Showtime ID: " + showtime.get("showtime_id") +
                    ", Theater ID: " + showtime.get("theatre_id") +
                  ", Screen ID: " + showtime.get("screen_id") +
                  ", Day: " + showtime.get("day") +
                  ", Date: " + showtime.get("date");
      JLabel showtimeLabel = new JLabel(showtimeDetails);
      viewMoviesFrame.add(showtimeLabel);
    }
  }

  JButton backButton = new JButton("Back");
  backButton.addActionListener(e -> {
    viewMoviesFrame.dispose();
    showAdminDashboard();
  });

  viewMoviesFrame.add(backButton);
  viewMoviesFrame.setVisible(true);
}

// USER FUNCTIONS
private static void showUserLogin() {
  JFrame userLogin = new JFrame("User Login");
  userLogin.setSize(300, 200);
  userLogin.setLayout(new GridLayout(3, 2));

  JLabel usernameLabel = new JLabel("Username:");
  JTextField usernameField = new JTextField();
  JLabel passwordLabel = new JLabel("Password:");
  JPasswordField passwordField = new JPasswordField();
```

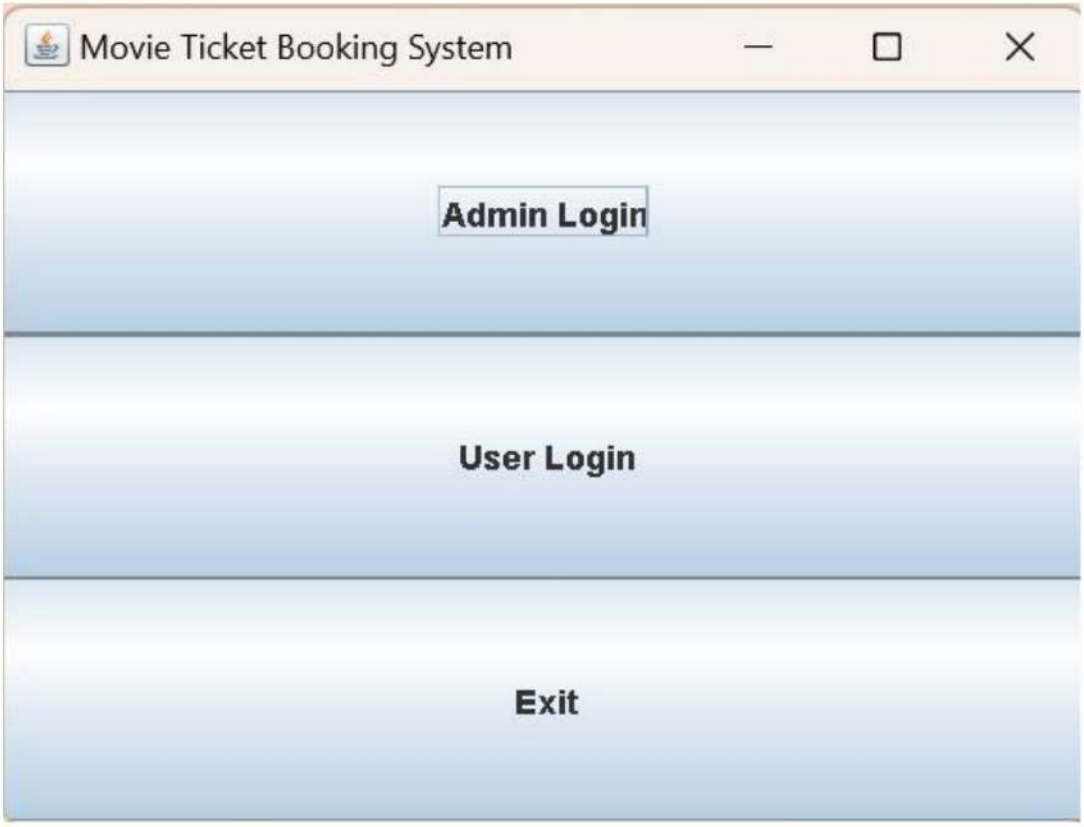
```
JButton loginButton = new JButton("Login");
JButton backButton = new JButton("Back");

userLogin.add(usernameLabel);
userLogin.add(usernameField);
userLogin.add(passwordLabel);
userLogin.add(passwordField);
userLogin.add(loginButton);
userLogin.add(backButton);

loginButton.addActionListener(e -> {
  String username = usernameField.getText();
  String password = new String(passwordField.getPassword());

  String sql = "SELECT UserID, Password FROM users WHERE Username = ?";
  Map<String, Object> user = db.validatePass(sql, username);

  if (user != null && password.equals(user.get("Password"))) {
    loggedInUserID = (int) user.get("UserID");
    JOptionPane.showMessageDialog(userLogin, "Login successful!");
    userLogin.dispose();
    showUserDashboard();
  } else {
    JOptionPane.showMessageDialog(userLogin, "Invalid credentials!");
  }
});

backButton.addActionListener(e -> {
  userLogin.dispose();
```

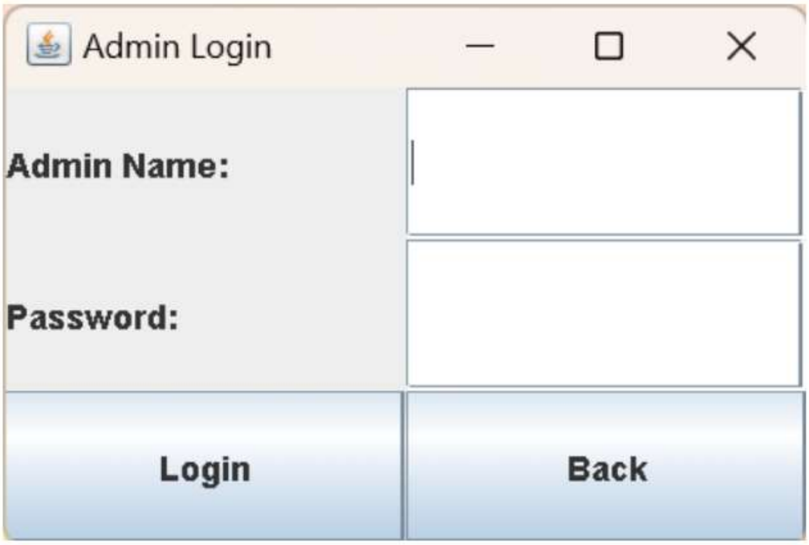
```
        showMainMenu();
    });

    userLogin.setVisible(true);
}
private static void showUserDashboard() {
    JFrame userDashboard = new JFrame("User Dashboard");
    userDashboard.setSize(400, 400);
    userDashboard.setLayout(new GridLayout(4, 1));

    JButton viewShowtimesButton = new JButton("View Showtimes");
    JButton bookTicketButton = new JButton("Book Ticket");
    JButton viewBookingsButton = new JButton("View Bookings");
    JButton logoutButton = new JButton("Logout");

    userDashboard.add(viewShowtimesButton);
    userDashboard.add(bookTicketButton);
    userDashboard.add(viewBookingsButton);
    userDashboard.add(logoutButton);

    viewShowtimesButton.addActionListener(e -> {
        userDashboard.dispose();
        viewUserShowtimes();
    });

    bookTicketButton.addActionListener(e -> {
        userDashboard.dispose();
        bookTicket();
    });
    viewBookingsButton.addActionListener(e -> {
        userDashboard.dispose();
        viewUserBookings();
    });

    logoutButton.addActionListener(e -> {
        loggedInUserID = -1;
        userDashboard.dispose();
        showMainMenu();
    });
    userDashboard.setVisible(true);
}

private static void viewUserShowtimes() {
    JFrame viewShowtimesFrame = new JFrame("Available Showtimes");
    viewShowtimesFrame.setSize(600, 600);
    viewShowtimesFrame.setLayout(new GridLayout(0, 1));

    List<Map<String, Object>> showtimes = db.getRecords("SELECT * FROM showtimes");
    for (Map<String, Object> showtime : showtimes) {
        String showtimeDetails = "Showtime ID: " + showtime.get("showtime_id") +
                                ", Movie ID: " + showtime.get("movie_ID") +
                                ", Theater ID: " + showtime.get("theatre_id") +
                                ", Screen ID: " + showtime.get("screen_id") +
                                ", Day: " + showtime.get("day") +
                                ", Date: " + showtime.get("date");
        JLabel showtimeLabel = new JLabel(showtimeDetails);
        viewShowtimesFrame.add(showtimeLabel);
    }
    JButton backButton = new JButton("Back");
    backButton.addActionListener(e -> {
        viewShowtimesFrame.dispose();
        showUserDashboard();
    });
    viewShowtimesFrame.add(backButton);
    viewShowtimesFrame.setVisible(true);
}
private static void bookTicket() {
    JFrame bookTicketFrame = new JFrame("Book Ticket");
    bookTicketFrame.setSize(300, 300);
    bookTicketFrame.setLayout(new GridLayout(4, 2));
    JLabel showtimeIDLabel = new JLabel("Showtime ID:");
    JTextField showtimeIDField = new JTextField();
    JLabel seatsLabel = new JLabel("Seats (e.g., A1,A2):");
    JTextField seatsField = new JTextField();
    JButton bookButton = new JButton("Book");
    JButton cancelButton = new JButton("Cancel");

    bookTicketFrame.add(showtimeIDLabel);
    bookTicketFrame.add(showtimeIDField);
    bookTicketFrame.add(seatsLabel);
    bookTicketFrame.add(seatsField);
    bookTicketFrame.add(bookButton);
    bookTicketFrame.add(cancelButton);
    bookButton.addActionListener(e -> {
        int showtimeID;
        String selectedSeats = seatsField.getText();
```

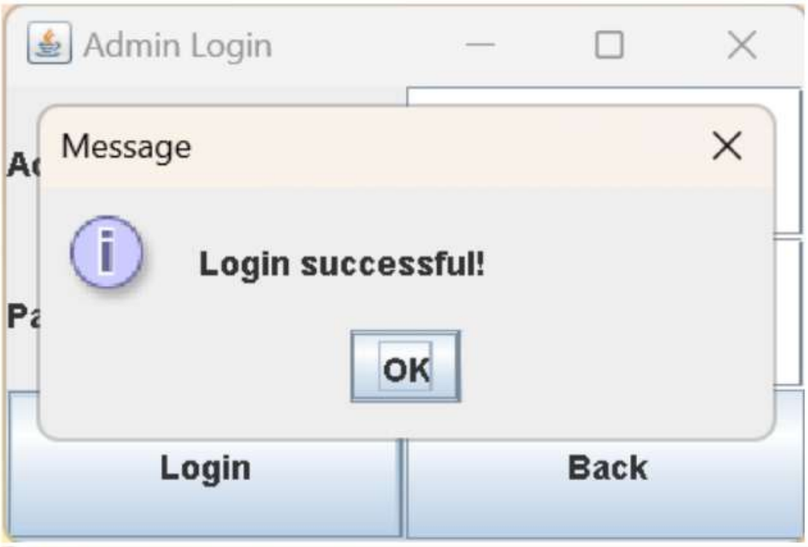
```
      try {
        showtimeID = Integer.parseInt(showtimeIDField.getText());
      } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(bookTicketFrame, "Invalid Showtime ID.");
        return;
      }
      String sql = "INSERT INTO tickets (UserID, showtime_id, selected_seats,
payment_status, availability_status) VALUES (?, ?, ?, 'Paid', 'Confirmed')";
      Object[] values = {loggedInUserID, showtimeID, selectedSeats};
      int rowsAffected = db.executeUpdate(sql, values);
      if (rowsAffected > 0) {
        JOptionPane.showMessageDialog(bookTicketFrame, "Ticket booked successfully!");
      } else {
        JOptionPane.showMessageDialog(bookTicketFrame, "Failed to book ticket.");
      }
      bookTicketFrame.dispose();
      showUserDashboard();
    });

    cancelButton.addActionListener(e -> {
      bookTicketFrame.dispose();
      showUserDashboard();
    });
    bookTicketFrame.setVisible(true);
  }

  private static void viewUserBookings() {
    JFrame viewBookingsFrame = new JFrame("My Bookings");
    viewBookingsFrame.setSize(600, 600);
    viewBookingsFrame.setLayout(new GridLayout(0, 1));
    List<Map<String, Object>> bookings = db.getRecords("SELECT * FROM tickets WHERE
UserID = " + loggedInUserID);
    for (Map<String, Object> booking : bookings) {
      String bookingDetails =  " Showtime ID: " + booking.get("showtime_id") +
                  ", Seats: " + booking.get("selected_seats") +
                  ", Status: " + booking.get("availability_status");
      JLabel bookingLabel = new JLabel(bookingDetails);
      viewBookingsFrame.add(bookingLabel);
    }
    JButton backButton = new JButton("Back");
    backButton.addActionListener(e -> {
      viewBookingsFrame.dispose();
      showUserDashboard();
    });

    viewBookingsFrame.add(backButton);
    viewBookingsFrame.setVisible(true);
  }
}

package movieticketbooking;
import java.sql.*;
import java.util.*;
import javax.swing.*;

public class DatabaseOperation {
  static final String DB_URL = "jdbc:mysql://localhost/moviedb";
  static final String USER = "root";
  static final String PASS = "GsJm$2408";
  public Connection connectToDatabase() {
    Connection conn = null;
    try {
      conn = DriverManager.getConnection(DB_URL, USER, PASS);
    } catch (SQLException e) {
      JOptionPane.showMessageDialog(null, "Database connection failed: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
    return conn;
  }

  public int executeUpdate(String sql, Object[] values) {
    int rowsAffected = 0;
    try (Connection conn = connectToDatabase();
       PreparedStatement ps = conn.prepareStatement(sql)) {
      for (int i = 0; i < values.length; i++) {
        ps.setObject(i + 1, values[i]);
      }
      rowsAffected = ps.executeUpdate();
    } catch (SQLException e) {
      JOptionPane.showMessageDialog(null, "SQL Update Failed: " + e.getMessage(),
"Error", JOptionPane.ERROR_MESSAGE);
    }
    return rowsAffected;
  }

  public List<Map<String, Object>> getRecords(String sql) {
    List<Map<String, Object>> records = new ArrayList<>();
    try (Connection conn = connectToDatabase();
       PreparedStatement pstmt = conn.prepareStatement(sql);
       ResultSet rs = pstmt.executeQuery()) {
      ResultSetMetaData rsmd = rs.getMetaData();
      int columnCount = rsmd.getColumnCount();
      while (rs.next()) {
        Map<String, Object> row = new HashMap<>();
        for (int i = 1; i <= columnCount; i++) {
          row.put(rsmd.getColumnName(i), rs.getObject(i));
        }
        records.add(row);
      }
    } catch (SQLException e) {
      JOptionPane.showMessageDialog(null, "SQL Query Failed: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);
    }
    return records;
  }

  public int getSeatingCapacity(String sql, int parameter) {
    int seatingCapacity = 0;
    try (Connection conn = connectToDatabase();
       PreparedStatement ps = conn.prepareStatement(sql)) {
      ps.setInt(1, parameter);
      ResultSet rs = ps.executeQuery();
      if (rs.next()) {
        seatingCapacity = rs.getInt("SeatingCapacity");
      }
    } catch (SQLException e) {
      JOptionPane.showMessageDialog(null, "Error fetching seating capacity: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
```

```
    }
    return seatingCapacity;
  }

  public ArrayList<Integer> getBookedSeats(int showtimeID) {
    String sql = "SELECT SelectedSeats FROM bookings WHERE ShowtimeID = ?";
    ArrayList<Integer> bookedSeats = new ArrayList<>();
    try (Connection conn = connectToDatabase();
       PreparedStatement ps = conn.prepareStatement(sql)) {
      ps.setInt(1, showtimeID);
      ResultSet rs = ps.executeQuery();
      while (rs.next()) {
        bookedSeats.add(rs.getInt("SelectedSeats"));
      }
    } catch (SQLException e) {
      JOptionPane.showMessageDialog(null, "Error fetching booked seats: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);
    }
    return bookedSeats;
  }

  public int removeBooking(int bookingID) {
    String sql = "DELETE FROM bookings WHERE BookingID = ?";
    int rowsAffected = 0;
    try (Connection conn = connectToDatabase();
       PreparedStatement ps = conn.prepareStatement(sql)) {
      ps.setInt(1, bookingID);
      rowsAffected = ps.executeUpdate();
    } catch (SQLException e) {
      JOptionPane.showMessageDialog(null, "Error removing booking: " + e.getMessage(),
"Error", JOptionPane.ERROR_MESSAGE);
    }
    return rowsAffected;
  }

  public Map<String, Object> validatePass(String sql, Object... params) {
    Map<String, Object> result = new HashMap<>();
    try (Connection conn = connectToDatabase();
       PreparedStatement ps = conn.prepareStatement(sql)) {
      for (int i = 0; i < params.length; i++) {
        ps.setObject(i + 1, params[i]);
      }
      ResultSet rs = ps.executeQuery();
      if (rs.next()) {
        ResultSetMetaData metaData = rs.getMetaData();
        int columnCount = metaData.getColumnCount();
        for (int i = 1; i <= columnCount; i++) {
          result.put(metaData.getColumnName(i), rs.getObject(i));
        }
      }
    } catch (SQLException e) {
      e.printStackTrace();
    }
    return result.isEmpty() ? null : result; // Return null if no record found
  }
  public int fetchUserID(String sql, String username) {
    int userID = 0;
    try (Connection conn = connectToDatabase();
       PreparedStatement ps = conn.prepareStatement(sql)) {
            ps.setString(1, username);

            ResultSet rs = ps.executeQuery();

            if (rs.next()) {

                userID = rs.getInt("UserID");

            }

        } catch (SQLException e) {

            JOptionPane.showMessageDialog(null, "Error fetching UserID: " + e.getMessage(),
"Error", JOptionPane.ERROR_MESSAGE);

        }

        return userID;

    }

}
```

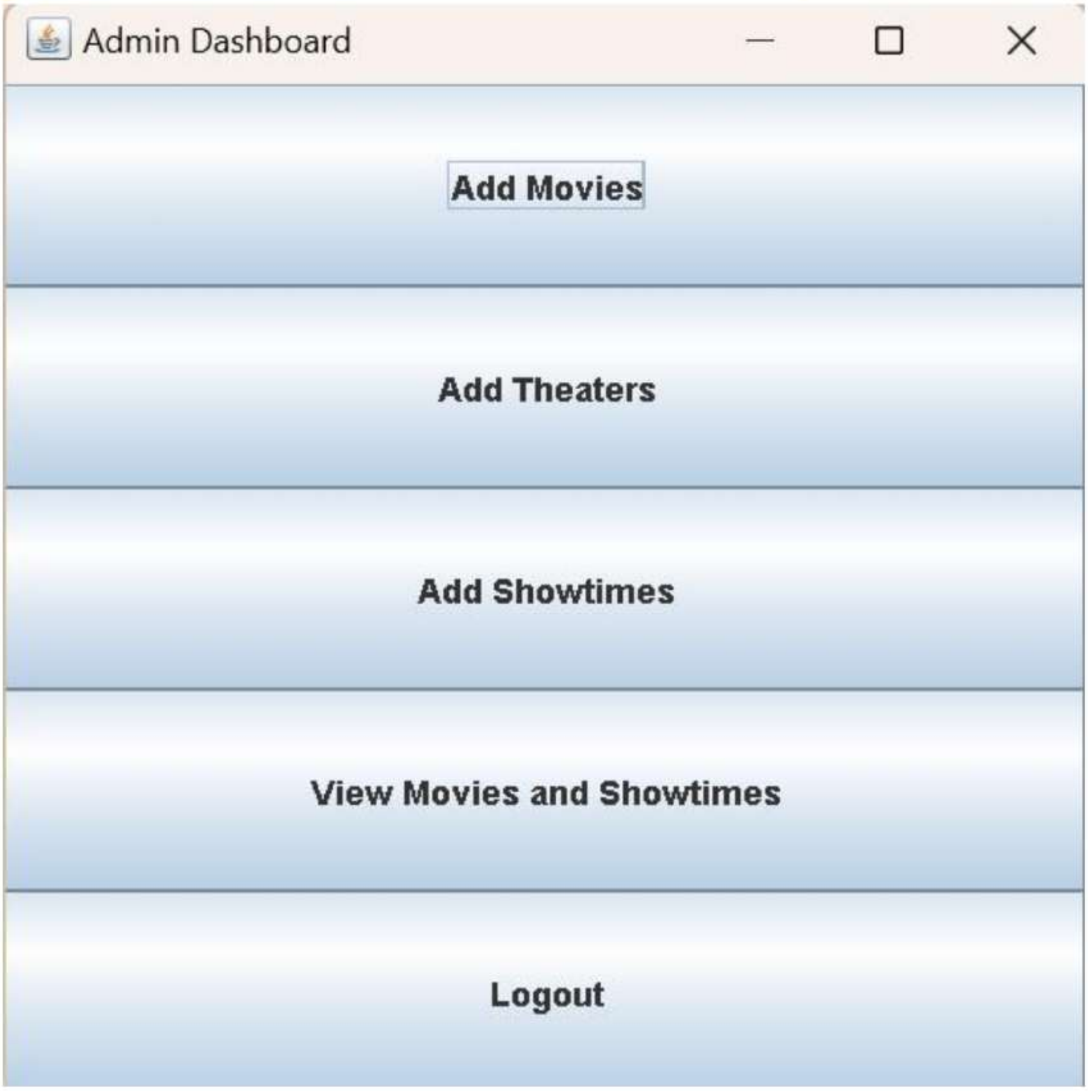
SNAPSHOTS:

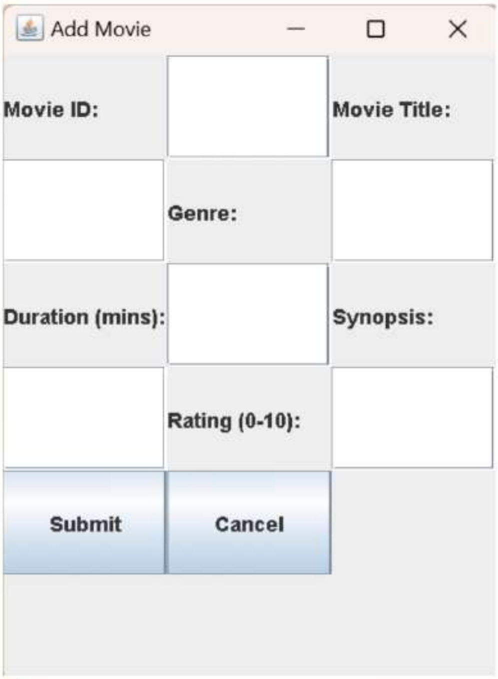

Movie Ticket Booking System
Admin Login
User Login
Exit

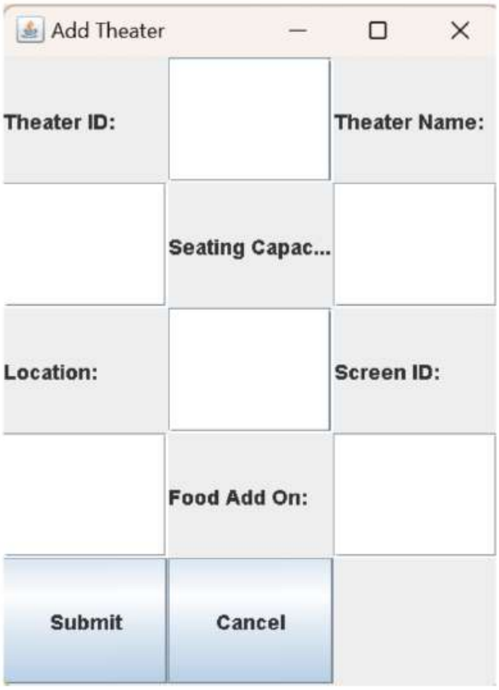

Admin Login
Admin Name:
Password:
Login
Back

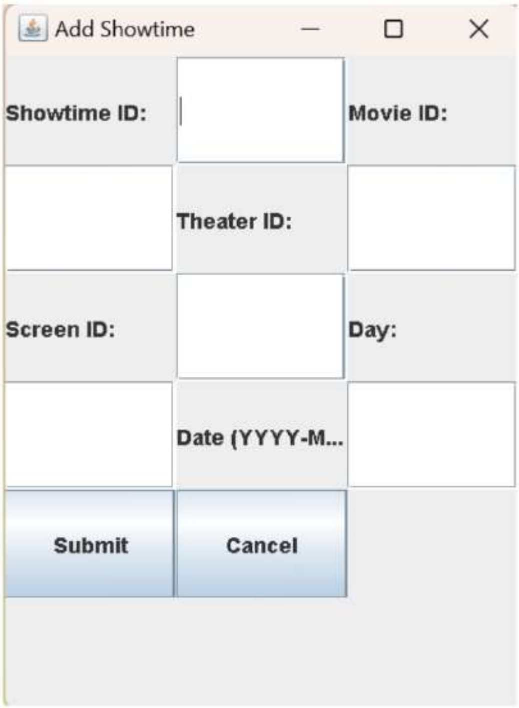


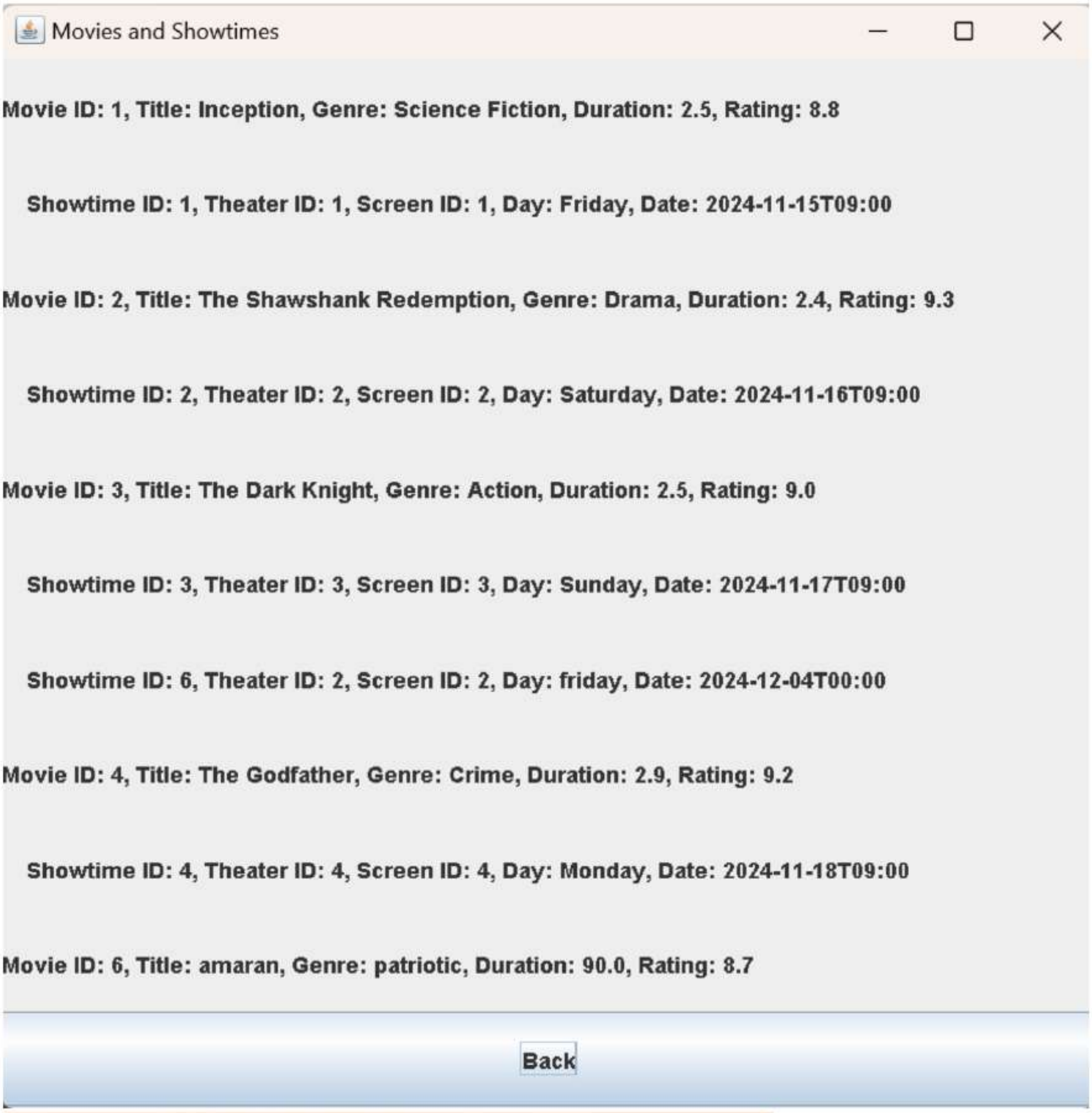
Login successful!



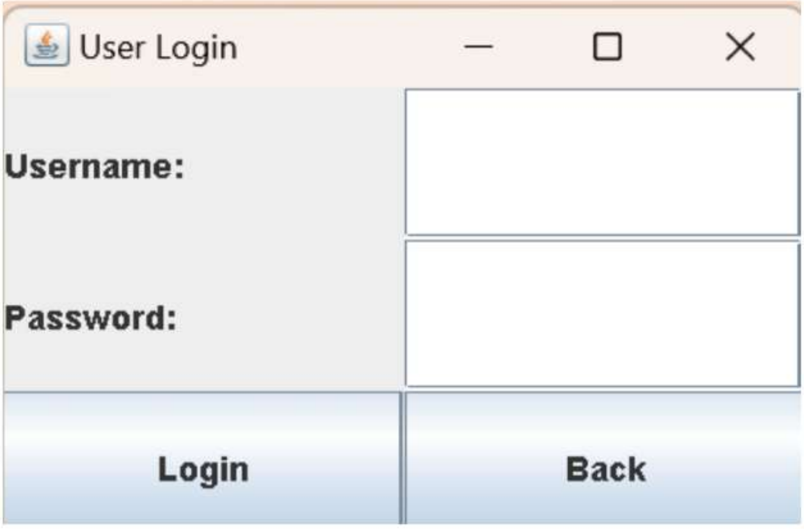

Add Movie
Movie ID:
Movie Title:
Genre:
Duration (mins):
Synopsis:
Rating (0-10):
Submit
Cancel

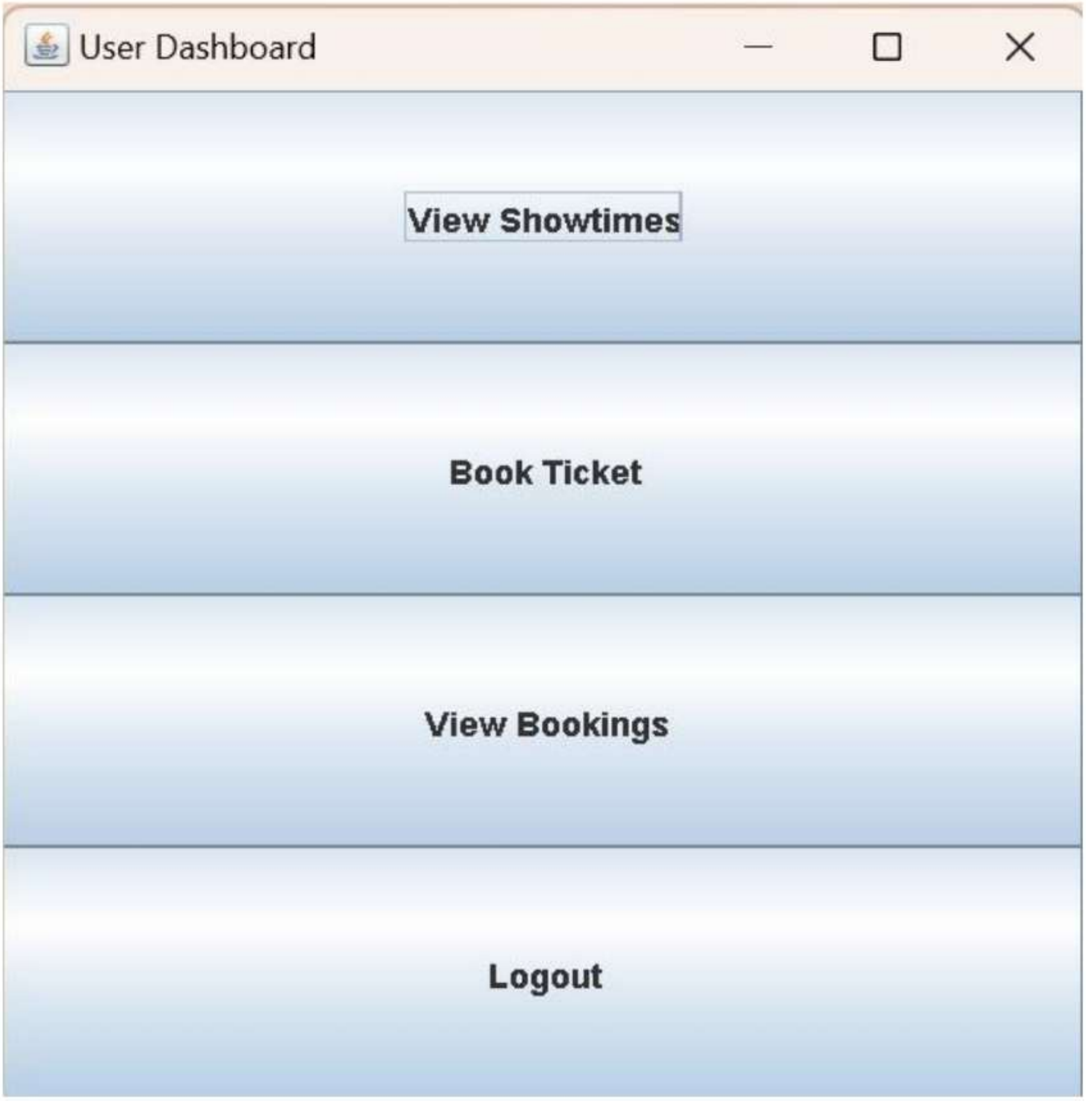

Add Theater
Theater ID:
Theater Name:
Seating Capac...
Location:
Screen ID:
Food Add On:
Submit
Cancel

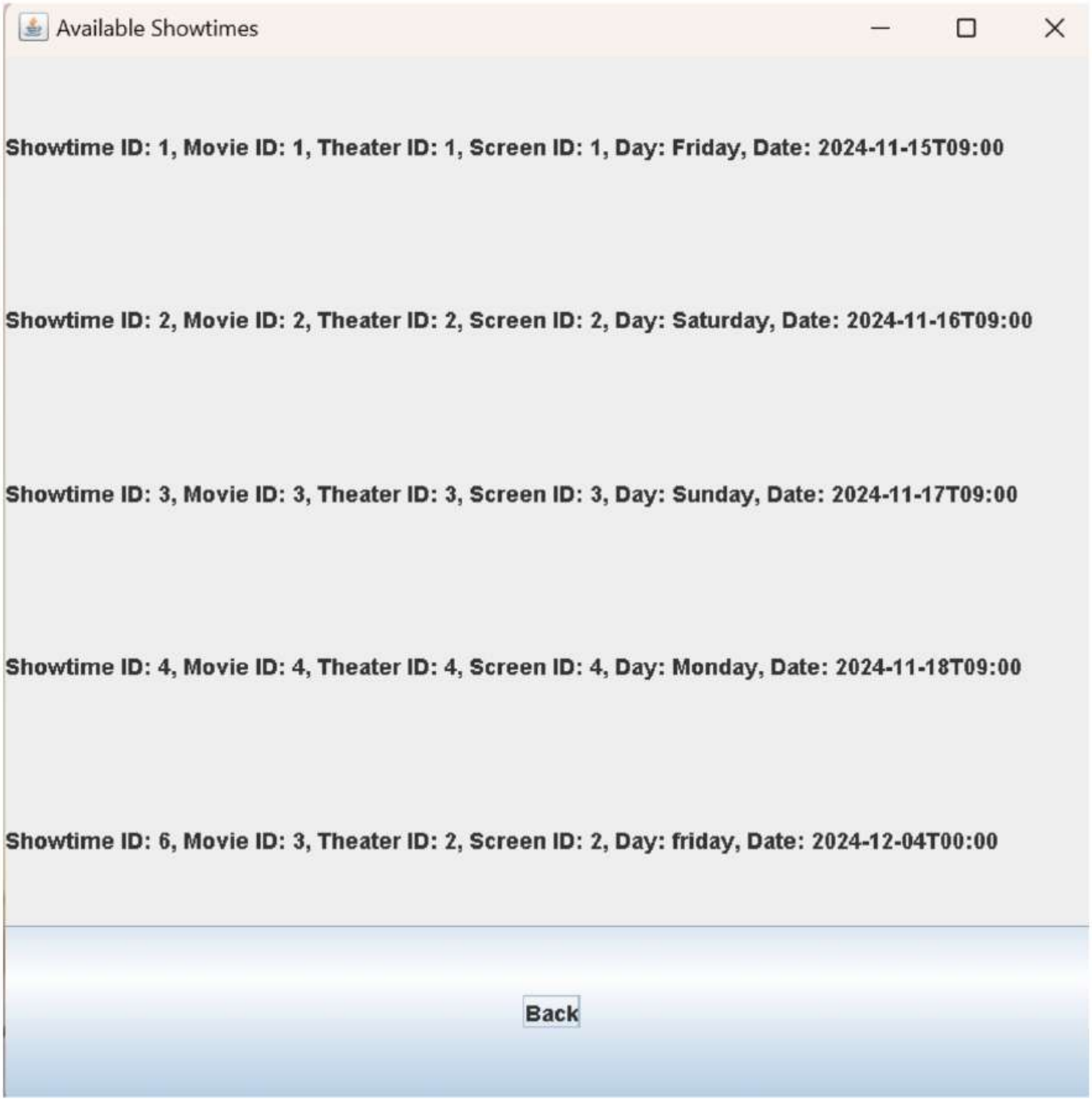

Add Showtime
Showtime ID:
Movie ID:
Theater ID:
Screen ID:
Day:
Date (YYYY-M...
Submit
Cancel



```
Movie ID: 1, Title: Inception, Genre: Science Fiction, Duration: 2.5, Rating: 8.8

  Showtime ID: 1, Theater ID: 1, Screen ID: 1, Day: Friday, Date: 2024-11-15T09:00

Movie ID: 2, Title: The Shawshank Redemption, Genre: Drama, Duration: 2.4, Rating: 9.3

  Showtime ID: 2, Theater ID: 2, Screen ID: 2, Day: Saturday, Date: 2024-11-16T09:00

Movie ID: 3, Title: The Dark Knight, Genre: Action, Duration: 2.5, Rating: 9.0

  Showtime ID: 3, Theater ID: 3, Screen ID: 3, Day: Sunday, Date: 2024-11-17T09:00

  Showtime ID: 6, Theater ID: 2, Screen ID: 2, Day: friday, Date: 2024-12-04T00:00

Movie ID: 4, Title: The Godfather, Genre: Crime, Duration: 2.9, Rating: 9.2

  Showtime ID: 4, Theater ID: 4, Screen ID: 4, Day: Monday, Date: 2024-11-18T09:00

Movie ID: 6, Title: amaran, Genre: patriotic, Duration: 90.0, Rating: 8.7
```




User Dashboard
View Showtimes
Book Ticket
View Bookings
Logout



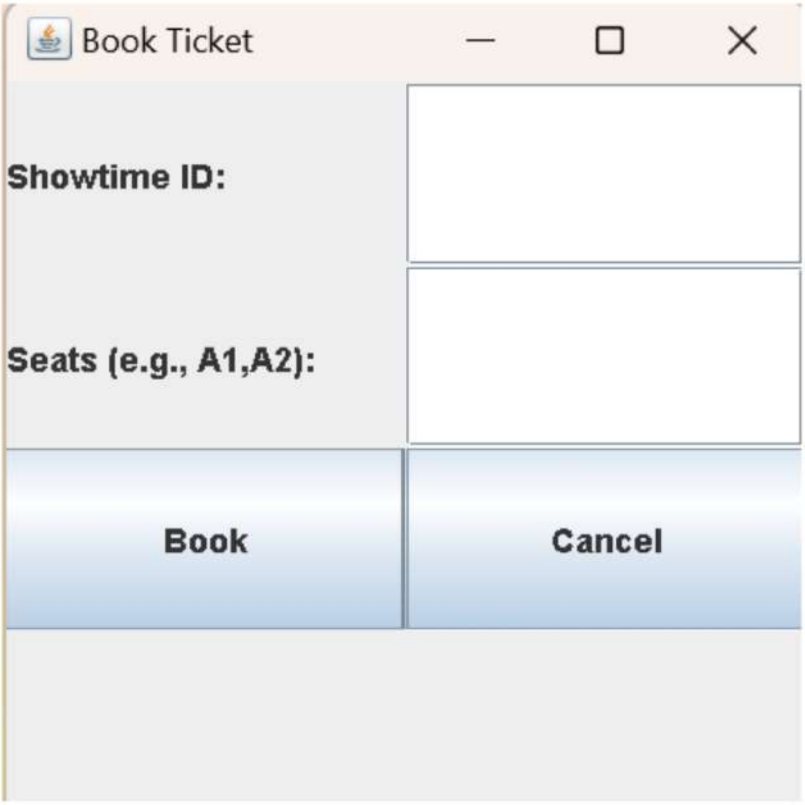
Showtime ID: 1, Movie ID: 1, Theater ID: 1, Screen ID: 1, Day: Friday, Date: 2024-11-15T09:00

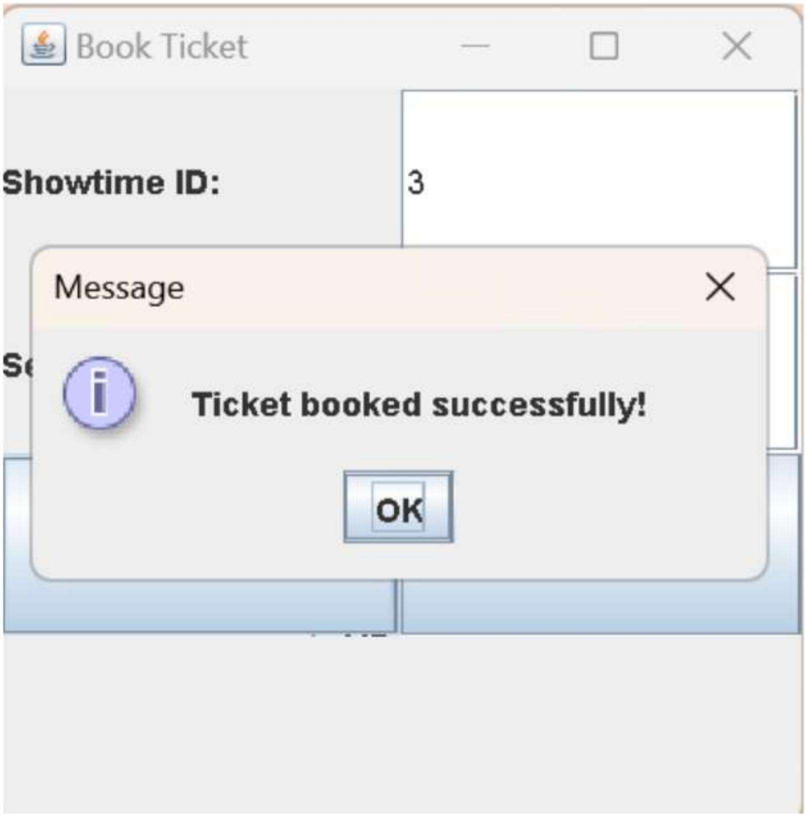
Showtime ID: 2, Movie ID: 2, Theater ID: 2, Screen ID: 2, Day: Saturday, Date: 2024-11-16T09:00

Showtime ID: 3, Movie ID: 3, Theater ID: 3, Screen ID: 3, Day: Sunday, Date: 2024-11-17T09:00

Showtime ID: 4, Movie ID: 4, Theater ID: 4, Screen ID: 4, Day: Monday, Date: 2024-11-18T09:00

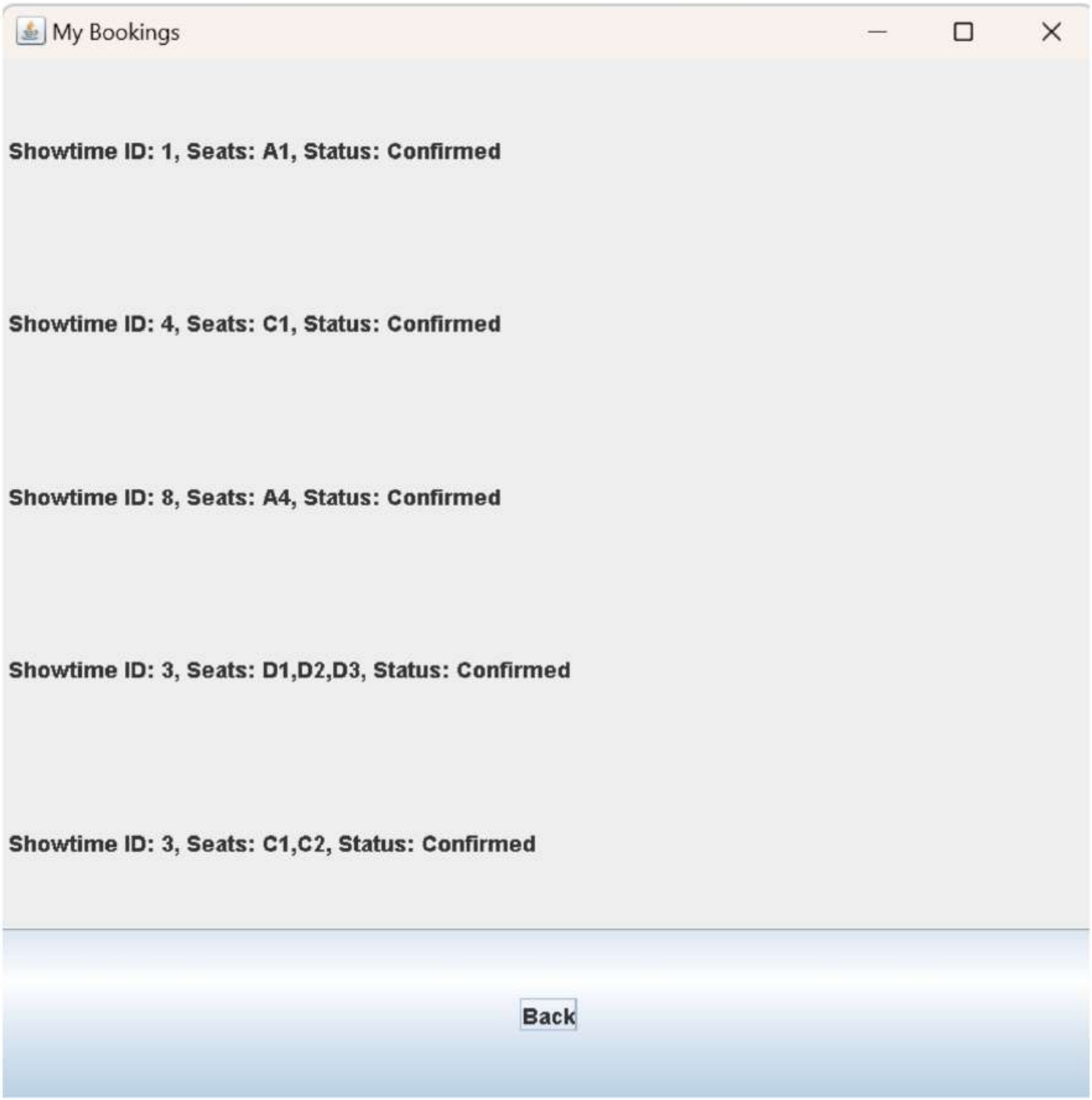
Showtime ID: 6, Movie ID: 3, Theater ID: 2, Screen ID: 2, Day: friday, Date: 2024-12-04T00:00





Showtime ID: 3

Ticket booked successfully!



Showtime ID: 1, Seats: A1, Status: Confirmed

Showtime ID: 4, Seats: C1, Status: Confirmed

Showtime ID: 8, Seats: A4, Status: Confirmed

Showtime ID: 3, Seats: D1,D2,D3, Status: Confirmed

Showtime ID: 3, Seats: C1,C2, Status: Confirmed