

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-01-Java Architecture, Language Basics](#) / [Lab-01-Logic Building](#)

Status	Finished
Started	Saturday, 5 October 2024, 9:18 PM
Completed	Saturday, 5 October 2024, 9:48 PM
Duration	30 mins 7 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 %)

```

1 import java.util.Scanner;
2 public class Main{
3     public static int checkoddeven(int input){
4         if(input % 2==0){
5             return 1;
6         }
7     } else {
8         return 2;
9     }
10 }
11 public static void main(String[] args){
12     Scanner sc = new Scanner(System.in);
13     int input=sc.nextInt();
14     int result= checkoddeven(input);
15     System.out.println(result);
16 }
17 }
```

	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 %)

```

1 import java.util.Scanner;
2 public class Main{
3     public static int getLastDigit(int input){
4         return Math.abs(input % 10);
5     }
6     public static void main(String[] args){
7         Scanner sc=new Scanner(System.in);
8         int input=sc.nextInt();
9         int result=getLastDigit(input);
10        System.out.println(result);
11    }
12 }
```

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

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

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 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

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

For example:

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

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 class usermaincode{
3     public int addlastdigits(int num1,int num2){
4         return Math.abs(num1)%10 + Math.abs(num2)%10;
5     }
6 }
7 public class Main{
8     public static void main(String[] args){
9         Scanner sc=new Scanner(System.in);
10        int num1=sc.nextInt();
11        int num2=sc.nextInt();
12        int sum=new usermaincode().addlastdigits(num1,num2);
13        System.out.println(sum);
14    }
15 }
```

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

Passed all tests! ✓



◀ Lab-01-MCQ

Jump to...

Is Even? ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-02-Flow Control Statements](#) / [Lab-02-Logic Building](#)

Status	Finished
Started	Sunday, 6 October 2024, 12:27 PM
Completed	Monday, 7 October 2024, 8:15 PM
Duration	1 day 7 hours

Question 1

Correct

Marked out of 5.00

Consider a sequence of the form 0, 1, 1, 2, 4, 7, 13, 24, 44, 81, 149...

Write a method program which takes as parameter an integer n and prints the nth term of the above sequence. The nth term will fit in an integer value.

Example Input:

5

Output:

4

Example Input:

8

Output:

24

Example Input:

11

Output:

149

For example:

Input	Result
5	4
8	24
11	149

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class Main{
3     public static void main(String args[]){
4         Scanner sc=new Scanner(System.in);
5         int b=sc.nextInt();
6         int s[]={0,1,1,2,4};
7         if(b>0)s[0]=0;
8         if(b>1)s[1]=1;
9         if(b>2)s[2]=1;
10        if(b>3)s[3]=2;
11        if(b>4)s[4]=4;
12        for(int i=5;i<b;i++){
13            s[i]=s[i-1]+s[i-2]+s[i-3];
14        }
15        System.out.print(s[b-1]);
16    }
17 }
```

	Input	Expected	Got	
✓	5	4	4	✓
✓	8	24	24	✓
✓	11	149	149	✓

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

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 %)

```

1 import java.util.Scanner;
2 public class MovieSuccessPredictor {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         int actingPower = sc.nextInt();
6         int criticsRating = sc.nextInt();
7         if (actingPower > 8 || criticsRating > 8) {
8             if (actingPower < 2 || criticsRating < 2) {
9                 System.out.println("No");
10            } else {
11                System.out.println("Yes");
12            }
13        }
14    }
15 }
```

```
16 } else {  
17     System.out.println("Maybe");  
18 }  
19 sc.close();  
20 }  
21 }  
22 }  
23 }
```

	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

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 %)

Reset answer

```

1 import java.util.Scanner;
2 public class FactorialTrailingZeros {
3     public static void main(String[] args) {
4         Scanner sc = new Scanner(System.in);
5         int n = sc.nextInt();
6         int count = 0;
7         for (int i = 5; n / i >= 1; i *= 5) {
8             count += n / i;
9         }
10        System.out.println(count);
11        sc.close();
12    }
13 }
```

--	--	--	--	--

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

Passed all tests! ✓



◀ Lab-02-MCQ

Jump to...

Lab-03-MCQ ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-03-Arrays](#) / [Lab-03-Logic Building](#)

Status	Finished
Started	Sunday, 6 October 2024, 10:33 PM
Completed	Sunday, 6 October 2024, 10:40 PM
Duration	7 mins 12 secs

Question 1

Correct

Marked out of 5.00

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 %)

```
1 ↴import java.util.Scanner;
2 ↴public class l{
```

```

3 public static int s(int n3,int[] n2){
4     int m=0;
5     int ms=0;
6     int c=0;
7     int cs=0;
8     for(int n:n2){
9         if(n>=0){
10             c++;
11             cs+=n;
12         }
13     else{
14         if(c>m){
15             m=c;
16             ms=cs;
17         }
18     else if(c==m){
19         ms+=cs;
20     }
21     c=0;
22     cs=0;
23 }
24 }
25 if(c>m){
26     m=c;
27     ms=cs;
28 }
29 }
30 else if(c==m){
31     ms+=cs;
32 }
33 return m>0 ? ms:-1;
34 }
35 }
36
37 public static void main(String args[]){
38     Scanner a=new Scanner(System.in);
39     int n1=a.nextInt();
40     int b[]={};
41     for(int i=0;i<n1;i++){
42         b[i]=a.nextInt();
43     }
44     int r=s(n1,b);
45     System.out.print(r);
46 }
47 }
48 }
```

	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 2

Correct

Marked out of 5.00

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:

$$\{(-8 \times 9), (-4 \times 9), (3 \times 9), (0 \times 9)\} = \{-72, -36, -27, 0\}$$

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:

$$\{(-77 \times 87), (0 \times 87), (-24 \times 87), (-45 \times 87), (-85 \times 87)\} = \{-6699, 0, -2088, -3915, -7395\}$$

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:

$$\{(-18 \times 9), (0 \times 9)\} = \{-162, 0\}$$

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 %)

```

1 import java.util.Scanner;
2 public class Main{
3     public static int[] p(int n,int[] n2){
4         int m=Integer.MIN_VALUE;
5         for(int num:n2){
6             if(num>m){
7                 m=num;
8             }
9         }
10        int[] s=new int[n];
11        for(int i=0;i<n;i++){
12            s[i]=n2[i]-m;
13        }
14        int[] r=new int[n];
15        for(int i=0;i<n;i++){
16            r[i]=s[i]*m;
17        }
18        return r;
19    }
20    public static void main(String args[]){
21        Scanner a=new Scanner(System.in);
22        int n=a.nextInt();
23        int b[]={};
24        for(int i=0;i<n;i++){
25            b[i]=a.nextInt();
26        }
27        int[] r=p(n,b);
28        for(int v:r){
29            System.out.print(v+" ");
30        }
31    }
32}
33

```

	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 3

Correct

Marked out of 5.00

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 %)

```

1 import java.util.Scanner;
2 public class s{
3     public static int c(int n1,int n2[]){
4         int[] p=new int[n1];
5         for(int i=0;i<n1;i++){
6             int m=n2[i];
7             p[i]=g(m,i);
8         }
9         int[] s=new int[n1];
10        for(int i=0;i<n1;i++){
11            s[i]=p[i]*p[i];
12        }
13        int sum=0;
14        for(int v:s){
15            sum+=v;
16        }
17        return sum;
18    }
19    public static int g(int num,int pos){
20        int d=(int) ((num/Math.pow(10,pos))%10);
21        return d;
22    }
23
24    public static void main(String args[]){
25        Scanner a=new Scanner(System.in);
26        int n=a.nextInt();
27        int b[]={};
28        for(int i=0;i<n;i++){
29            b[i]=a.nextInt();
30        }
31        int r=c(n,b);
32        System.out.print(r);
33    }
34 }
```

	Input	Expected	Got	
✓	5 1 51 436 7860 41236	107	107	✓
✓	5 1 5 423 310 61540	53	53	✓

Passed all tests! ✓

◀ Lab-03-MCQ

Jump to...

Simple Encoded Array ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-04-Classes and Objects](#) / [Lab-04-Logic Building](#)

Status	Finished
Started	Sunday, 6 October 2024, 10:41 PM
Completed	Sunday, 6 October 2024, 10:44 PM
Duration	2 mins 52 secs

Question 1

Correct

Marked out of 5.00

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 %)

```

1 public class Student{
2     private String name;
3     private int rollno;
4     public Student(){
5         System.out.println("No-arg constructor is invoked");
6         this.name=null;
7         this.rollno=0;
8     }
9     public Student(String name){
10        System.out.println("1 arg constructor is invoked");
11        this.name=name;
12        this.rollno=0;
13    }
14    public Student(String name,int rollno){
15        System.out.println("2 arg constructor is invoked");
16        this.name=name;
17        this.rollno=rollno;
18    }
19    public void display(){
20        System.out.println("Name =" +(name !=null ?name: "null")+" , Roll no = "+rollno);
21    }
22    public static void main(String args[]){
23        Student s1=new Student();
24        Student s2=new Student("Rajalakshmi");
25        Student s3=new Student("Lakshmi",101);
26        s1.display();
27        s2.display();
28        s3.display();
29    }
30 }
31

```

	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! ✓

Question 2

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 %)

```
1 import java.util.Scanner;
2 public class Main{
3     private String manufacter;
4     private String operating_system;
5     public String color;
6     private int cost;
7     Main(String manufacter,String operating_system,String color,int cost){
8         this.manufacter=manufacter;
9         this.operating_system=operating_system;
10        this.color=color;
11        this.cost=cost;
12    }
13    public void display(){
14        System.out.printf("manufacturer = %s\noperating_system = %s\ncolor = %s\ncost = %d",
15    }
16    public static void main(String args[]){
17        Scanner a=new Scanner(System.in);
18        Main b=new Main("Redmi","Andriod","Blue",34000);
19        b.display();
20    }
21
22 }
```

	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 3

Correct

Marked out of 5.00

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

```

1 import java.util.Scanner;
2 class Circle
3 {
4     private double radius;
5     public Circle(double radius){
6         // set the instance variable radius
7         this.radius=radius;
8     }
9     public void setRadius(double radius){
10        // set the radius
11        this.radius=radius;
12    }
13
14    }
15    public double getRadius()    {
16        // return the radius
17        return this.radius;
18    }
19
20    }
21    public double calculateArea() { // complete the below statement
22        return Math.PI*Math.pow((this.radius),2);
23    }
24    }
25    public double calculateCircumference()    {
26        // complete the statement
27        return 2*Math.PI*(this.radius);
28    }
29 }
30 class prog{
31     public static void main(String[] args)  {
32         int r;
33         Scanner sc= new Scanner(System.in);
34         r=sc.nextInt();
35         Circle c= new Circle(r);
36         System.out.println("Area = "+String.format("%.2f", c.calculateArea()));
37         // invoke the calculatecircumference method
38         System.out.printf("Circumference = %.2f",c.calculateCircumference());
39     }
40 }
```

```
40  
41 }  
42 }  
43 }
```

	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! ✓

◀ Lab-04-MCQ

Jump to...

Number of Primes in a specified range ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-05-Inheritance](#) / [Lab-05-Logic Building](#)

Status	Finished
Started	Sunday, 6 October 2024, 10:45 PM
Completed	Sunday, 6 October 2024, 10:51 PM
Duration	6 mins 8 secs

Question 1

Correct

Marked out of 5.00

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 %)**Reset answer**

```
1 class College
2 {
3     protected String collegeName;
4
5     public College(String c) {
6         // initialize the instance variables
7         collegeName=c;
8     }
9
10    public void admitted() {
11        System.out.println("A student admitted in "+collegeName);
12    }
13 }
14 class Student extends College{
15
16     String stu;
17     String dep;
18
19     public Student(String collegeNameo, String studentNameo, String deparo) {
20         // initialize the instance variables
21         super(collegeNameo);
22         stu=studentNameo;
23         dep=deparo;
24     }
25 }
26
27 
```

```
27 // printing the details\n28     // return the details of the stu\n29     return "CollegeName : "+collegeName+"\nStudentName : "+stu+"\nDepartment : "+dep;\n30 }\n31 }\n32 }\n33 class prog {\n34 public static void main (String[] args) {\n35     Student s1 = new Student("REC","Venkatesh","CSE");\n36             // invoke the admitted() method\n37     s1.admitted();\n38     System.out.println(s1.toString());\n39 }\n40 }
```

	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! ✓

Question 2

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 %)

[Reset answer](#)

```
1 class BankAccount {  
2     // Private field to store the account number  
3     private String accountNumber;  
4  
5     // Private field to store the balance  
6     private double balance;  
7  
8     // Constructor to initialize account number and balance  
9     public BankAccount(String accountNumber,double balance){  
10         this.accountNumber=accountNumber;  
11         this.balance=balance;  
12     }  
13  
14  
15  
16  
17     // Method to deposit an amount into the account  
18     public void deposit(double amount) {  
19         // Increase the balance by the deposit amount  
20         balance+=amount;  
21     }  
22  
23  
24     // Method to withdraw an amount from the account  
25     public void withdraw(double amount) {  
26         // Check if the balance is sufficient for the withdrawal  
27         if (balance >= amount) {  
28             // Decrease the balance by the withdrawal amount  
29             balance -= amount;  
30         } else {  
31             // Print a message if the balance is insufficient  
32             System.out.println("Insufficient balance");  
33         }  
34     }  
35  
36     // Method to get the current balance  
37     public double getBalance() {  
38         // Return the current balance  
39         return balance;  
40     }  
41 }
```

```

42 }
43
44 class SavingsAccount extends BankAccount {
45     // Constructor to initialize account number and balance
46     public SavingsAccount(String accountNumber, double balance) {
47         // Call the parent class constructor
48         super(accountNumber,balance);
49     }
50 }
51
52

```

	Expected	Got	
✓	<p>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</p>	<p>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</p>	✓

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

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 %)

```
1 v class Mobile{
2 v     public Mobile(){
3         System.out.println("Basic Mobile is Manufactured");
4     }
5 }
6 v class CameraMobile extends Mobile{
7 v     public CameraMobile(){
8         System.out.println("Camera Mobile is Manufactured");
9     }
10 v    public void newFeature(){
11         System.out.println("Camera Mobile with 5MG px");
12     }
13 }
14 v class AndroidMobile extends CameraMobile{
15 v     public AndroidMobile(){
16         System.out.println("Android Mobile is Manufactured");
17     }
18     }
19 v    public void androidMobile(){
20         System.out.println("Touch Screen Mobile is Manufactured");
21     }
22 }
23 v class prog{
24 v     public static void main(String[] args){
25         AndroidMobile o=new AndroidMobile();
26         o.newFeature();
27         o.androidMobile();
```

```
28 }  
29 }
```

	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! ✓

[◀ Lab-05-MCQ](#)

Jump to...

[Is Palindrome Number? ►](#)

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-06-String, StringBuffer](#) / [Lab-06-Logic Building](#)

Status	Finished
Started	Sunday, 6 October 2024, 10:51 PM
Completed	Sunday, 6 October 2024, 10:54 PM
Duration	2 mins 19 secs

Question 1

Correct

Marked out of 5.00

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 %)

```

1 import java.util.Scanner;
2 import java.util.Arrays;
3 import java.util.HashSet;
4 import java.util.Set;
5 public class P{
6     public static String cal(String a1,String b){
7         if((a1==null||a1.trim().isEmpty())&&(b==null||b.trim().isEmpty())) return "null";
8         String c=a1+b;
9         Set<Character> u=new HashSet<>();
10        for(char c1:c.toCharArray()){
11            if(Character.isAlphabetic(c1)){
12                u.add(c1);
13            }
14        }
15        char[] ca=new char[u.size()];
16        int i=0;
17        for(char c2:u{
18            ca[i++]=c2;
19        }
    }
}

```

```
20
21     Arrays.sort(ca);
22     return new StringBuilder(new String(ca)).reverse().toString();
23 }
24 public static void main(String args[]){
25     Scanner a=new Scanner(System.in);
26     String i1=a.nextLine();
27     String i2=a.nextLine();
28     System.out.println(cal(i1,i2));
29 }
```

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

Passed all tests! ✓

Question 2

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 %)

```
1 import java.util.Scanner;
2 import java.util.Arrays;
```

```

3 import java.lang.String;
4 public class Prog{
5     public static String pw(String input,int index){
6         String[] words=input.split(" ");
7         int f=(index/10)-1;
8         int ss=(index%10)-1;
9         String fp=pws(words[f]);
10        String sp=pws(words[ss]);
11        return fp+ " " +sp;
12    }
13    public static String pws(String o){
14        int l=o.length();
15        int mid=l/2;
16        String l1,f1;
17        if(l%2==0){
18            f1=o.substring(0,mid);
19            f1=new StringBuilder(f1).reverse().toString();
20            l1=o.substring(mid);
21            return f1+l1;
22        }
23        else{
24            f1=o.substring(0,mid+1);
25            f1=new StringBuilder(f1).reverse().toString();
26            l1=o.substring(mid);
27            return f1+l1;
28        }
29    }
30    public static void main(String[] args){
31        Scanner a=new Scanner(System.in);
32        String s=a.nextLine();
33        int n=a.nextInt();
34        String r=pw(s,n);
35        System.out.println(r);
36    }
37}
38

```

	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 3

Correct

Marked out of 5.00

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 %)

```

1 import java.util.Scanner;
2 public class Main{
3     public static void main(String args[]){
4         Scanner a=new Scanner(System.in);
5         String input=a.nextLine();
6         String[] w=input.split(":");
7         StringBuilder r=new StringBuilder();
8         for(String word:w){
9             char c1=word.charAt(0);
10            char c2=word.charAt(1);
11            if(c1==c2){
12                r.append(Character.toUpperCase(c1));
13            }
14            else{
15                int p=c1-'a'+1;
16                int p1=c2-'a'+1;
17                int d=Math.abs(p-p1);
18                char newc=(char)('a'+d-1);
19                r.append(Character.toUpperCase(newc));
20            }
21        }
22        System.out.print(r.toString());
23    }
24 }
```

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

Passed all tests! ✓

◀ Lab-06-MCQ

Jump to...

Return second word in Uppercase ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-07-Interfaces](#) / [Lab-07-Logic Building](#)

Status	Finished
Started	Sunday, 6 October 2024, 10:54 PM
Completed	Sunday, 6 October 2024, 10:56 PM
Duration	1 min 46 secs

Question 1

Correct

Marked out of 5.00

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.

```
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:

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

For example:

Test	Input	Result
1	Sadvin Sanjay Sruthi	Sadvin 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 %)

```
1 import java.util.Scanner;
2 interface Playable{
3     void play();
4 }
5 class Football implements Playable{
6     private String name;
7     public Football(String name){
8         this.name=name;
9     }
10    public void play(){
11        System.out.println(name+ " is Playing football");
12    }
13 }
14 class Volleyball implements Playable{
15     private String name;
16     public Volleyball(String name){
17         this.name=name;
18     }
19    public void play(){
20        System.out.println(name + " is Playing volleyball");
21    }
22 }
23 class Basketball implements Playable{
24     private String name;
25 }
```

```

25
26     public BasketBall(String name){
27         this.name=name;
28     }
29     public void play(){
30         System.out.println(name + " is Playing basketball");
31     }
32 }
33 public class Main{
34     public static void main(String args[]){
35         Scanner a=new Scanner(System.in);
36         Playable f=new Football(a.nextLine());
37         Playable v=new Volleyball(a.nextLine());
38         Playable b=new Basketball(a.nextLine());
39         f.play();
40         v.play();
41         b.play();
42     }
43 }

```

	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 2

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 %)

```
1 interface RBI{
2     String parentBank="RBI";
3     double rateofInterest();
4     default void policyNote(){
5         System.out.println("RBI has a new Policy issued in 2023");
6     }
7     static void regulations(){
8         System.out.println("RBI has updated new regulations in 2024.");
9     }
10 }
11 class SBI implements RBI{
12     public double rateofInterest(){
13         return 7.6;
14     }
15 }
16 class Karur implements RBI{
17     public double rateofInterest(){
18         return 7.4;
19     }
20 }
21 public class Main{
22     public static void main(String args[]){
23         SBI sbi =new SBI();
24         Karur k=new Karur();
25         sbi.policyNote();
26         RBI.regulations();
27         System.out.println("SBI rate of interest: " +sbi.rateofInterest() + " per annum.");
28         System.out.println("Karur rate of interest: " + k.rateofInterest() + " per annum.");
29     }
30 }
```

29
30 }

	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 3

Correct

Marked out of 5.00

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);}
```

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 %)

[Reset answer](#)

```
1 import java.util.Scanner;
2 import java.util.Scanner;
3 interface Sports {
4     public void setHomeTeam(String name);
5     public void setVisitingTeam(String name);
6 }
7 interface Football extends Sports {
8     public void homeTeamScored(int points);
9     public void v(int points);
10 }
11 class College implements Football {
12     private String homeTeam;
13     private String visitingTeam;
14     private int homeScore;
15     private int visitingScore;
16     public void setHomeTeam(String name){
17         this.homeTeam=name;
18     }
19     public void setVisitingTeam(String name){
20         this.visitingTeam=name;
21     }
22     public void homeTeamScored(int points){
23         this.homeScore+=points;
24         System.out.println(homeTeam+" "+points+" scored");
25     }
26     public void v(int points){
27         this.visitingScore+=points;
28     }
29 }
```

```

27     visitingScore=points,
28     System.out.println(visitingTeam+" "+points+" scored");
29   }
30   public void winningTeam(){
31     if(homeScore>visitingScore)
32       System.out.println(homeTeam + " is the winner!");
33     else if(homeScore<visitingScore){
34       System.out.println(visitingTeam + " is the winner!");
35     }
36   }
37   else{
38     System.out.println("It's a tie match.");
39   }
40 }
41 }
42 }
43 class prog{
44   public static void main(String[] args){
45     String hname;
46     Scanner sc= new Scanner(System.in);
47     hname=sc.nextLine();
48     String vteam=sc.next();
49     int hpoints=sc.nextInt();
50     int vpoints=sc.nextInt();
51     College G= new College();
52     G.setHomeTeam(hname);

```

	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! ✓

◀ Lab-07-MCQ

Jump to...

Generate series and find Nth element ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-08 - Polymorphism, Abstract Classes, final Keyword](#) / [Lab-08-Logic Building](#)

Status	Finished
Started	Sunday, 6 October 2024, 10:56 PM
Completed	Sunday, 6 October 2024, 10:58 PM
Duration	1 min 29 secs

Question 1

Correct

Marked out of 5.00

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 %)

```
1 class FinalExample {
2
3
4     int maxSpeed = 120;
5
6     // Final method
7     public final void displayMaxSpeed() {
8         System.out.println("The maximum speed is: " + maxSpeed + " km/h");
9     }
10 }
11
12 class SubClass extends FinalExample {
13
14     /* public void displayMaxSpeed() {
15         System.out.println("Cannot override a final method");
16     }*/
17
18     // You can create new methods here
19     public void showDetails() {
20         System.out.println("This is a subclass of FinalExample.");
21     }
22 }
23 }
```

```
24 v class prog {  
25 v     public static void main(String[] args) {  
26         FinalExample obj = new FinalExample();  
27         obj.displayMaxSpeed();  
28  
29         SubClass subObj = new SubClass();  
30         subObj.showDetails();  
31     }  
32 }  
33  
34
```

	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! ✓



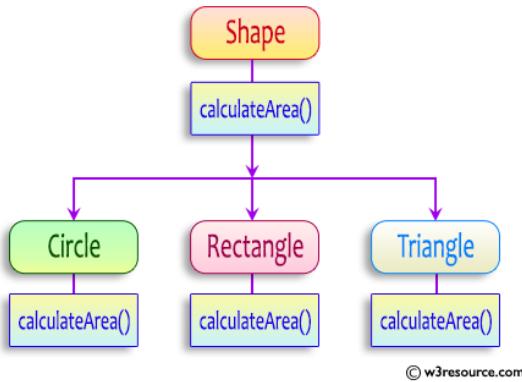
Question 2

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 %)

```

1 import java.util.Scanner;
2 interface Playable {
3     void play();
4 }
  
```

```

5  class Football implements Playable {
6      String name;
7  public Football(String name) {
8      this.name = name;
9  }
10 @Override
11 public void play() {
12     System.out.println(name + " is playing football");
13 }
14 }
15 class Volleyball implements Playable {
16     String name;
17 public Volleyball(String name) {
18     this.name = name;
19 }
20 @Override
21 public void play() {
22     System.out.println(name + " is playing volleyball");
23 }
24 }
25 abstract class Shape {
26     public abstract double calculateArea();
27 }
28 class Rectangle extends Shape {
29     double length;
30     double breadth;
31 public Rectangle(double length, double breadth) {
32     this.length = length;
33     this.breadth = breadth;
34 }
35 @Override
36 public double calculateArea() {
37     return length * breadth;
38 }
39 }
40 class Triangle extends Shape {
41     double base;
42     double height;
43 public Triangle(double base, double height) {
44     this.base = base;
45     this.height = height;
46 }
47 @Override
48 public double calculateArea() {
49     return 0.5 * base * height;
50 }
51 }
52 class Circle extends Shape {

```

	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 3

Correct

Marked out of 5.00

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 %)

```

1 import java.util.Scanner;
2
3 abstract class Operations {
4     public boolean isVowel(char c) {
5         return c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u';
6     }
7
8     public boolean hasVowelEnds(String s) {
9         return (isVowel(s.charAt(0)) && isVowel(s.charAt(s.length() - 1)));
10    }

```

```

11 }
12
13 class Concatenate extends Operations {
14     public String ans = "";
15
16     Concatenate(String[] words) {
17         for (String s : words) {
18             if (hasVowelEnds(s.toLowerCase())) {
19                 ans += s.toLowerCase();
20             }
21         }
22     }
23
24     public String getAnswer() {
25         if (ans.isEmpty()) return "no matches found";
26         return ans.trim();
27     }
28 }
29
30 class prog {
31     public static void main(String args[]) {
32         Scanner scan = new Scanner(System.in);
33         int n = scan.nextInt();
34         scan.nextLine();
35         String[] words = scan.nextLine().split(" ");
36         Concatenate cn = new Concatenate(words);
37         System.out.println(cn.getAnswer());
38         scan.close();
39     }
40 }
41

```

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

Passed all tests! ✓

◀ Lab-08-MCQ

Jump to...

FindStringCode ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-09-Exception Handling](#) / [Lab-09-Logic Building](#)

Status	Finished
Started	Friday, 18 October 2024, 7:58 PM
Completed	Friday, 18 October 2024, 9:15 PM
Duration	1 hour 16 mins

Question 1

Correct

Marked out of 5.00

Write a Java program to handle `ArithmaticException` 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.ArithmaticException: / 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.ArithmaticException: / by zero I am always executed

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class ExceptionHandlingExample {
3     public static void main(String[] args) {
4         Scanner scanner = new Scanner(System.in);
5         //System.out.println("Enter the size of the array:");
6         int size = scanner.nextInt();
7
8         int[] array = new int[size];
9
10        //System.out.println("Enter " + size + " elements:");
11        for (int i = 0; i < size; i++) {
12            array[i] = scanner.nextInt();
13        }
14
15        try {
16            // Divide the 0th element by the 1st element
17            int result = array[0] / array[1];
18            //System.out.println("Result of division: " + result);
19
20            // Access an element out of bounds
21            System.out.println("Accessing 3rd index: " + array[3]);
22        } catch (ArithmaticException e) {
23            System.out.println("java.lang.ArithmaticException: " + e.getMessage());
24        } catch (ArrayIndexOutOfBoundsException e) {
25            System.out.println("java.lang.ArrayIndexOutOfBoundsException: " + e.getMessage());
26        } finally {
27            System.out.println("I am always executed");
28        }
29    }
}

```

```
30     scanner.close();  
31 }  
32 }
```

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

Passed all tests! ✓



Question 2

Correct

Marked out of 5.00

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](#)

```

1 class prog {
2
3     public static void main(String[] args) {
4         int n = 82;
5
6         trynumber(n);
7
8         n = 37;
9
10    trynumber(n);
11 }
12
13
14
15
16     public static void trynumber(int n) {
17         try {
18
19             checkEvenNumber(n);
20
21
22             System.out.println(n + " is even.");
23         } catch (OddNumberException e) {
24
25             System.out.println(e.getMessage());
26         }
27     }
28
29
30     public static void checkEvenNumber(int number) throws OddNumberException {
31         if (number % 2 != 0) {
32
33             throw new OddNumberException("Error: " + number + " is odd.");
34         }
35     }
36 }
37
38
39     class OddNumberException extends Exception {
40         public OddNumberException(String message) {
41             super(message);
42         }
43     }
44 }
```

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

Passed all tests! ✓

Question 3

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 %)

```
1 import java.util.Scanner;
2 import java.util.InputMismatchException;
3
4 class prog {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         int length = sc.nextInt();
9
10
11         int[] name = new int[length];
12         int sum = 0;
13
14     try {
15
16         for (int i = 0; i < length; i++) {
17             name[i] = sc.nextInt();
18             sum += name[i];
19         }
20
21         System.out.println(sum);
22     } catch (InputMismatchException e) {
23
24         System.out.println("You entered bad data.");
25     }
26 }
```

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

Passed all tests! ✓

◀ Lab-09-MCQ

Jump to...

The "Nambiar Number" Generator ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-10- Collection- List](#) / [Lab-10-Logic Building](#)

Status	Finished
Started	Monday, 4 November 2024, 6:10 PM
Completed	Monday, 4 November 2024, 6:47 PM
Duration	37 mins 30 secs

Question 1

Correct

Marked out of 1.00

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 %)

```

1 import java.util.ArrayList;
2 import java.util.Scanner;
3 public class Main {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         ArrayList<Integer> arrayList = new ArrayList<>();
7         int n = scanner.nextInt();
8         for (int i = 0; i < n; i++) {
9             arrayList.add(scanner.nextInt());
10    }
11    if (!arrayList.isEmpty()) {
12        System.out.println("ArrayList: " + arrayList);
13        int firstElement = arrayList.get(0);
14        int lastElement = arrayList.get(arrayList.size() - 1);
15        System.out.println("First : " + firstElement + ", Last : " + lastElement);
16    } else {
17        System.out.println("The ArrayList is empty.");
18    }
19    scanner.close();
20 }
21 }
22
23

```

	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

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](#)

```
1 import java.util.ArrayList;
2 import java.util.Scanner;
3
4 class prog {
5
6     public static void main(String[] args)
7     {
8         Scanner sc= new Scanner(System.in);
9         int n = sc.nextInt();
10
11     ArrayList<Integer> list = new ArrayList<Integer>();
12
13     for(int i = 0; i<n;i++)
14         list.add(sc.nextInt());
15
16     System.out.println("ArrayList: " + list);
17
18
19     list.set(1, 100);
20
21     // Getting the index of the first occurrence of 100
22     System.out.println("Index of 100 = " + list.indexOf(100));
23
24     // Getting the index of the last occurrence of 100
25     System.out.println("LastIndex of 100 = " + list.lastIndexOf(100));
26
27     // Check whether 200 is in the list or not
28     System.out.println(list.contains(200)); // Output: false
29
30     // Print ArrayList size
31     System.out.println("Size Of ArrayList = " + list.size());
32
33     // Inserting 500 at index 1
34     list.add(1, 500);
35
36     // Removing an element from position 3
37     list.remove(3);
38
39     // Print the final ArrayList
40     System.out.print("ArrayList: " + list);
41
42     sc.close();
43 }
44 }
```

	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]	✓

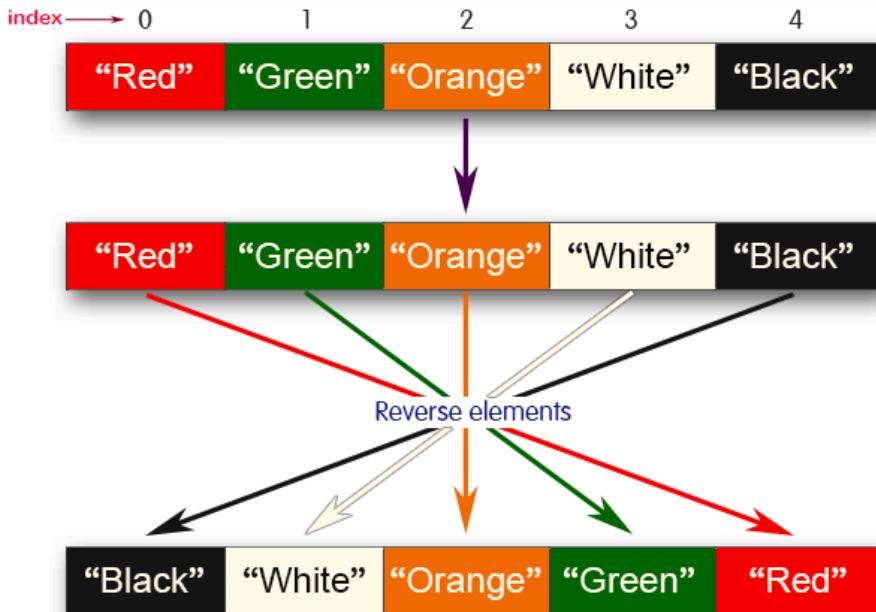
Passed all tests! ✓

Question 3

Correct

Marked out of 1.00

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 %)

```

1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.Scanner;
4 public class ReverseArrayList {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         int n = sc.nextInt();
8         sc.nextLine();
9         ArrayList<String> list = new ArrayList<>();
10        for (int i = 0; i < n; i++) {
11            list.add(sc.nextLine().trim());
12        }
13        System.out.println("List before reversing : ");
14        System.out.println(list);
15        Collections.reverse(list);
16        System.out.println("List after reversing : ");
17        System.out.println(list);
18        sc.close();
19    }
20 }
```

	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! ✓

◀ Lab-10-MCQ

Jump to...

Lab-11-MCQ ▶

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-11-Set, Map](#) / [Lab-11-Logic Building](#)

Status	Finished
Started	Saturday, 16 November 2024, 4:52 PM
Completed	Saturday, 16 November 2024, 4:59 PM
Duration	7 mins 8 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 %)

[Reset answer](#)

```

1 import java.util.HashSet;
2 import java.util.Scanner;
3
4 class prog {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         // Read the number of elements
9         int n = sc.nextInt();
10
11        // Create a HashSet object called numbers
12        HashSet<Integer> numbers = new HashSet<>();
13
14        // Add values to the set
15        for (int i = 0; i < n; i++) {
16            numbers.add(sc.nextInt());
17        }
18    }

```

```

19     // Read the search key
20     int skey = sc.nextInt();
21
22     // Check if the key exists in the set
23     if (numbers.contains(skey)) {
24         System.out.println(skey + " was found in the set.");
25     } else {
26         System.out.println(skey + " was not found in the set.");
27     }
28
29     sc.close();
30 }
31
32 }
```

	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 %)

```
1 import java.util.HashSet;
2 import java.util.Scanner;
3 public class CompareSets {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         HashSet<String> set1 = new HashSet<>();
7         int n1 = sc.nextInt();
8         sc.nextLine();
9         for (int i = 0; i < n1; i++) {
10             set1.add(sc.nextLine());
11         }
12         HashSet<String> set2 = new HashSet<>();
13         int n2 = sc.nextInt();
14         sc.nextLine();
15         for (int i = 0; i < n2; i++) {
16             set2.add(sc.nextLine());
17         }
18         set1.retainAll(set2);
19         for (String element : set1) {
20             System.out.println(element);
21         }
22     }
23 }
```

	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[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 %)[Reset answer](#)

```

1 import java.util.HashMap;
2 import java.util.Map.Entry;
3 import java.util.Set;
4 import java.util.Scanner;
5
6 class prog {
7     public static void main(String[] args) {
8         // Creating HashMap with default initial capacity and load factor
9         HashMap<String, Integer> map = new HashMap<String, Integer>();
10
11     String name;
12     int num;
13     Scanner sc = new Scanner(System.in);
14
15     // Reading number of entries
16     int n = sc.nextInt();
17     for (int i = 0; i < n; i++) {
18         name = sc.next();
19         num = sc.nextInt();
20         map.put(name, num);
21     }
22
23     // Printing key-value pairs
24     Set<Entry<String, Integer>> entrySet = map.entrySet();
25     for (Entry<String, Integer> entry : entrySet) {
26         System.out.println(entry.getKey() + " : " + entry.getValue());
27     }
28
29     System.out.println("-----");
30
31     // Creating another HashMap
32     HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();
33
34     // Inserting key-value pairs to anotherMap using put() method
35     anotherMap.put("SIX", 6);
36     anotherMap.put("SEVEN", 7);
37
38     // Inserting key-value pairs of map to anotherMap using putAll() method
39     anotherMap.putAll(map);
40
41     // Printing key-value pairs of anotherMap
42     entrySet = anotherMap.entrySet();
43     for (Entry<String, Integer> entry : entrySet) {
44         System.out.println(entry.getKey() + " : " + entry.getValue());
45     }
46
47     // Adds key-value pair 'FIVE-5' only if it is not present in map
48     map.putIfAbsent("FIVE", 5);

```

```

49
50     // Retrieving a value associated with key 'TWO'
51     Integer value = map.get("TWO");
52     System.out.println(value != null ? value : "Key 'TWO' not found");

```

	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! ✓

◀ Lab-11-MCQ

Jump to...

TreeSet example ►

[Dashboard](#) / [My courses](#) / [CS23333-OOPUJ-2023](#) / [Lab-12-Introduction to I/O, I/O Operations, Object Serialization](#) / [Lab-12-Logic Building](#)

Status	Finished
Started	Saturday, 16 November 2024, 5:00 PM
Completed	Saturday, 16 November 2024, 6:08 PM
Duration	1 hour 8 mins

Question 1

Incorrect

Marked out of 5.00

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:

- 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.

- 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".

- 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 %)

```

1 import java.util.Scanner;
2 public class WordReversal {
3     public static String reverseWordsWithCaseOption(String sentence, int case_option) {
4         String[] words = sentence.split(" ");
5         StringBuilder modifiedSentence = new StringBuilder();
6         for (String word : words) {
7             String reversedWord = reverseWordWithCaseOption(word, case_option);
8             modifiedSentence.append(reversedWord).append(" ");
9         }
}

```

```

10     return modifiedSentence.toString().trim();
11 }
12 private static String reverseWordWithCaseOption(String word, int case_option) {
13     StringBuilder reversed = new StringBuilder(word.length());
14     for (int i = word.length() - 1; i >= 0; i--) {
15         reversed.append(word.charAt(i));
16     }
17     if (case_option == 0) {
18         for (int i = 0; i < reversed.length(); i++) {
19             char ch = reversed.charAt(i);
20             if (Character.isUpperCase(ch)) {
21                 reversed.setCharAt(i, Character.toLowerCase(ch));
22             } else if (Character.isLowerCase(ch)) {
23                 reversed.setCharAt(i, Character.toUpperCase(ch));
24             }
25         }
26     } else if (case_option == 1) {
27         return reversed.toString();
28     }
29     return reversed.toString();
30 }
31 public static void main(String[] args) {
32     Scanner scanner = new Scanner(System.in);
33     String input = scanner.nextLine();
34     int caseOption = scanner.nextInt();
35     if (caseOption != 0 && caseOption != 1) {
36         scanner.close();
37         return;
38     }
39     String result = reverseWordsWithCaseOption(input, caseOption);
40     System.out.println(result);
41     scanner.close();
42 }
43 }
44
45
46
47
48

```

	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	✗

Your code must pass all tests to earn any marks. Try again.

Show differences

Question 2

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 0's.

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 (0000000000000000000000000000).

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: 00001000000000000000000010000000000001000000000010000000000001

The decoded string (original word) will be: WIPRO

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

For example:

Input	Result
010010001	ZYX
00001000000000000000000010000000000001000000000010000000000001	WIPRO

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class BinaryDecoder {
3     public static String decode(String binaryString) {
4         String[] sequences = binaryString.split("1");
5         StringBuilder decodedWord = new StringBuilder();
6         for (String sequence : sequences) {
7             if (!sequence.isEmpty()) {
8                 int length = sequence.length();
9                 char letter = (char) ('Z' - (length - 1));
10                decodedWord.append(letter);
11            }
12        }
13        return decodedWord.toString();
14    }
15    public static void main(String[] args) {
16        Scanner scanner = new Scanner(System.in);
17        String input = scanner.nextLine();
18        if (input.matches("[01]+")) {
19            String decodedWord = decode(input);

```

```
20 }  
21 }  
22 }  
23 }  
24 }
```

	Input	Expected	Got	
✓	010010001	ZYX	ZYX	✓
✓	00001000000000000000000010000000000010000000000010000000000001	WIPRO	WIPRO	✓

Passed all tests! ✓

/

Question 3

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 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	8
b c	

Answer: (penalty regime: 0 %)

```

1 import java.util.HashSet;
2 import java.util.Scanner;
3 public class CommonCharSum {
4     public static int findSingleDigitSum(char[] input1, char[] input2) {
5         HashSet<Character> set1 = new HashSet<>();
6         HashSet<Character> set2 = new HashSet<>();
7         for (char c : input1) {
8             set1.add(c);
9         }
10        for (char c : input2) {
11            set2.add(c);
12        }
13        set1.retainAll(set2);
14        int sum1 = 0;
15        for (char c : set1) {
16            sum1 += (int) c;
17        }
18        while (sum1 >= 10) {
19            int tempSum = 0;
20            while (sum1 > 0) {
21                tempSum += sum1 % 10;
22            }
23        }
24    }
25 }
```

```
22         sum1 /= 10;
23     }
24     sum1 = tempSum;
25 }
26 return sum1;
27 }
28 public static void main(String[] args) {
29     Scanner sc = new Scanner(System.in);
30     String input1String = sc.nextLine();
31     char[] input1 = input1String.replaceAll("\\s", "").toCharArray();
32     String input2String = sc.nextLine();
33     char[] input2 = input2String.replaceAll("\\s", "").toCharArray();
34     int result = findSingleDigitSum(input1, input2);
35     System.out.println(result);
36 }
37 }
38 }
39 }
```

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

Passed all tests! ✓

◀ Lab-12-MCQ

Jump to...

Identify possible words ►

Java miniproject

Source code:

```
import javafx.application.Application;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.Background;
import javafx.scene.layout.StackPane;
import javafx.scene.layout.VBox;
import javafx.geometry.Pos;
import javafx.stage.Stage;
import javafx.scene.image.Image;
import javafx.scene.layout.BackgroundImage;

public class MainApp extends Application {

    @Override
    public void start(Stage primaryStage) {
        showMainWindow(primaryStage);
    }

    private void showMainWindow(Stage primaryStage) {
        Button backButton = new Button("Back");
        backButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-
font-weight: bold; -fx-text-fill: white; -fx-background-color: #333;");
        backButton.setOnAction(e -> showLoginPage(primaryStage));

        Button foodTrackingButton = new Button("Food Tracking");
        Button activityTrackingButton = new Button("Activity Tracking");
        Button medicalConditionTrackingButton = new Button("Medical Condition
Tracking");
        Button waterTrackingButton = new Button("Water Tracking");
        Button bmiScaleButton = new Button("BMI Scale");
        Button userProfileButton = new Button("User Profile");

        foodTrackingButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size:
14px;");
        activityTrackingButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size:
14px;");
        medicalConditionTrackingButton.setStyle("-fx-font-family: 'Arial'; -fx-
font-size: 14px;");
        waterTrackingButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size:
14px;");
        bmiScaleButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size:
14px;");
```

```

        userProfileButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px;");

        // Set up button actions to navigate to corresponding pages
        foodTrackingButton.setOnAction(e -> openFoodTrackingPage());
        activityTrackingButton.setOnAction(e -> openActivityTrackingPage());
        medicalConditionTrackingButton.setOnAction(e ->
openMedicalConditionPage());
        waterTrackingButton.setOnAction(e -> openWaterTrackingPage());
        bmiScaleButton.setOnAction(e -> openBmiScalePage());
        userProfileButton.setOnAction(e -> openUserProfilePage());

        VBox layout = new VBox(20);
        layout.getChildren().addAll(
            backButton,
            foodTrackingButton,
            activityTrackingButton,
            medicalConditionTrackingButton,
            waterTrackingButton,
            bmiScaleButton,
            userProfileButton
        );
        layout.setAlignment(Pos.CENTER);

        // Set background and scene without using additional properties
        StackPane root = new StackPane();
        Image image = new Image("file:/C:/XboxGames/image.jpg"); // Update the
file path as needed
        BackgroundImage backgroundImage = new BackgroundImage(image, null, null,
null, null);
        Background background = new Background(backgroundImage);
        root.setBackground(background);
        root.getChildren().add(layout);

        Scene scene = new Scene(root, 600, 400);
        primaryStage.setTitle("Personal Fitness Tracker");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    private void openFoodTrackingPage() {
        // Open Food Tracking page logic here
        FoodTrackingPage foodTrackingPage = new FoodTrackingPage();
        Stage foodTrackingStage = new Stage();
        foodTrackingPage.start(foodTrackingStage);
    }
}

```

```
}

private void openActivityTrackingPage() {
    // Open Activity Tracking page logic here
    ActivityTrackingPage activityTrackingPage = new ActivityTrackingPage();
    Stage activityTrackingStage = new Stage();
    activityTrackingPage.start(activityTrackingStage);
}

private void openMedicalConditionPage() {
    // Open Medical Condition page logic here
    MedicalConditionTrackingPage medicalConditionPage = new
MedicalConditionTrackingPage();
    Stage medicalConditionStage = new Stage();
    medicalConditionPage.start(medicalConditionStage);
}

private void openWaterTrackingPage() {
    // Open Water Tracking page logic here
    WaterTrackingPage waterTrackingPage = new WaterTrackingPage();
    Stage waterTrackingStage = new Stage();
    waterTrackingPage.start(waterTrackingStage);
}

private void openBmiScalePage() {
    // Open BMI Scale page logic here
    BmiScalePage bmiScalePage = new BmiScalePage();
    Stage bmiScaleStage = new Stage();
    bmiScalePage.start(bmiScaleStage);
}

private void openUserProfilePage() {
    // Open User Profile page logic here
    UserProfilePage userProfilePage = new UserProfilePage();
    Stage userProfileStage = new Stage();
    userProfilePage.start(userProfileStage);
}

private void showLoginPage(Stage primaryStage) {
    // Logic to show login page
}

public static void main(String[] args) {
    launch(args);
}
```

```
}

import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.Button;
import javafx.scene.layout.VBox;
import javafx.stage.Stage;
import javafx.scene.control.Alert;
import javafx.scene.control.Alert.AlertType;
import javafx.scene.control.TextField;
import javafx.scene.control.Label;
import java.sql.*;

public class ActivityTrackingPage extends Application {

    @Override
    public void start(Stage stage) {
        // Title or text for the page title
        Button backButton = new Button("Back to Main");
        backButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px; -fx-
font-weight: bold; -fx-text-fill: white; -fx-background-color: #333;");
        backButton.setOnAction(e -> stage.close()); // This closes the current
page, you may want to return to the main window

        // Buttons for activity tracking features
        Button trackActivityButton = new Button("Track New Activity");
        trackActivityButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size:
14px;");
        trackActivityButton.setOnAction(e -> openTrackActivityWindow());

        Button activityHistoryButton = new Button("View Activity History");
        activityHistoryButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size:
14px;");

        // Layout for the page
        VBox layout = new VBox(20);
        layout.setAlignment(Pos.CENTER);
        layout.getChildren().addAll(backButton, trackActivityButton,
activityHistoryButton);

        // Set the scene for the page
        Scene scene = new Scene(layout, 400, 300);
        stage.setTitle("Activity Tracking");
        stage.setScene(scene);
```

```
        stage.show();
    }

private void openTrackActivityWindow() {
    // Open a new window to track a new activity
    Stage stage = new Stage();
    VBox layout = new VBox(15);

    // Inputs for activity type, duration, distance
    Label activityLabel = new Label("Enter activity type (e.g., Running):");
    TextField activityField = new TextField();

    Label durationLabel = new Label("Enter duration (minutes):");
    TextField durationField = new TextField();

    Label distanceLabel = new Label("Enter distance (km):");
    TextField distanceField = new TextField();

    Button saveButton = new Button("Save Activity");
    saveButton.setOnAction(e -> {
        String activityType = activityField.getText();
        String duration = durationField.getText();
        String distance = distanceField.getText();

        if (!activityType.isEmpty() && !duration.isEmpty() &&
!distance.isEmpty()) {
            saveActivity(activityType, Integer.parseInt(duration),
Double.parseDouble(distance));
            Alert alert = new Alert(AlertType.INFORMATION);
            alert.setTitle("Success");
            alert.setHeaderText(null);
            alert.setContentText("Activity saved successfully!");
            alert.showAndWait();
        } else {
            Alert alert = new Alert(AlertType.ERROR);
            alert.setTitle("Error");
            alert.setHeaderText(null);
            alert.setContentText("Please fill in all fields.");
            alert.showAndWait();
        }
    });
}

layout.getChildren().addAll(activityLabel, activityField, durationLabel,
durationField, distanceLabel, distanceField, saveButton);
Scene scene = new Scene(layout, 300, 250);
```

```
        stage.setTitle("Track New Activity");
        stage.setScene(scene);
        stage.show();
    }

    private void saveActivity(String activityType, int duration, double distance)
{
    // Database connection details
    String url = "jdbc:mysql://localhost:3306/fitness_tracker"; // Replace
with your actual database name
    String user = "root"; // Replace with your MySQL username
    String password = "manu"; // Replace with your MySQL password

    String query = "INSERT INTO activity_tracking (user_id, activity_type,
duration, distance, activity_date) VALUES (?, ?, ?, ?, ?, NOW())";

    try (Connection conn = DriverManager.getConnection(url, user, password);
         PreparedStatement stmt = conn.prepareStatement(query)) {

        // Assuming a hardcoded user_id for now (you'll likely get this from
the logged-in user)
        int userId = 1; // You can replace this with the actual user ID
dynamically

        stmt.setInt(1, userId);
        stmt.setString(2, activityType);
        stmt.setInt(3, duration);
        stmt.setDouble(4, distance);

        stmt.executeUpdate();
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

public static void main(String[] args) {
    launch(args);
}
}

import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
```

```
import javafx.scene.text.Font;
import javafx.stage.Stage;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;

public class BmiScalePage extends Application {

    @Override
    public void start(Stage stage) {
        // Title for the page
        Label titleLabel = new Label("BMI Scale");
        titleLabel.setFont(new Font("Arial", 24));

        // Input fields for height and weight
        Label heightLabel = new Label("Enter your height (in meters):");
        TextField heightField = new TextField();

        Label weightLabel = new Label("Enter your weight (in kilograms):");
        TextField weightField = new TextField();

        // Button to calculate BMI
        Button calculateButton = new Button("Calculate BMI");
        calculateButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px");

        // Label to display the result
        Label resultLabel = new Label("Your BMI will be displayed here.");
        resultLabel.setStyle("-fx-font-size: 16px; -fx-font-weight: bold");

        // Calculate BMI when the button is clicked
        calculateButton.setOnAction(e -> {
            try {
                double height = Double.parseDouble(heightField.getText());
                double weight = Double.parseDouble(weightField.getText());
                if (height > 0 && weight > 0) {
                    double bmi = weight / (height * height);
                    resultLabel.setText(String.format("Your BMI is: %.2f", bmi));
                    saveBmiToDatabase(height, weight, bmi); // Save the result
                } else {
                    resultLabel.setText("Please enter valid positive values.");
                }
            } catch (Exception e) {
                resultLabel.setText("Please enter valid positive values.");
            }
        });
    }

    private void saveBmiToDatabase(double height, double weight, double bmi) {
        Connection connection = null;
        PreparedStatement statement = null;
        String query = "INSERT INTO bmi_data (height, weight, bmi) VALUES (?, ?, ?)";

        try {
            connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/bmi_db", "root", "password");
            statement = connection.prepareStatement(query);
            statement.setDouble(1, height);
            statement.setDouble(2, weight);
            statement.setDouble(3, bmi);
            statement.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        } finally {
            try {
                if (statement != null) statement.close();
                if (connection != null) connection.close();
            } catch (SQLException e) {
                e.printStackTrace();
            }
        }
    }
}
```

```

        } catch (NumberFormatException ex) {
            resultLabel.setText("Please enter valid numbers.");
        }
    });

    // Back button to go to the previous page
    Button backButton = new Button("Back");
    backButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px;");
    backButton.setOnAction(e -> stage.close()); // Close the current page,
modify this as per your navigation logic

    // Layout for the page
    VBox layout = new VBox(15);
    layout.setAlignment(Pos.CENTER);
    layout.getChildren().addAll(
        titleLabel,
        heightLabel,
        heightField,
        weightLabel,
        weightField,
        calculateButton,
        resultLabel,
        backButton
    );

    // Set up the scene
    Scene scene = new Scene(layout, 400, 300);
    stage.setTitle("BMI Scale");
    stage.setScene(scene);
    stage.show();
}

private void saveBmiToDatabase(double height, double weight, double bmi) {
    String url = "jdbc:mysql://localhost:3306/fitness_tracker"; // Replace
with your actual database name
    String user = "root"; // Replace with your MySQL username
    String password = "manu"; // Replace with your MySQL password

    String query = "INSERT INTO bmi_tracking (user_id, height, weight, bmi,
bmi_date) VALUES (?, ?, ?, ?, ?, NOW())";

    try (Connection conn = DriverManager.getConnection(url, user, password);
PreparedStatement stmt = conn.prepareStatement(query)) {

```

```
// Assuming a hardcoded user_id for now (you'll likely get this from
the logged-in user)
    int userId = 1; // Replace with actual user ID dynamically

    stmt.setInt(1, userId);
    stmt.setDouble(2, height);
    stmt.setDouble(3, weight);
    stmt.setDouble(4, bmi);

    stmt.executeUpdate();
} catch (SQLException ex) {
    ex.printStackTrace();
}
}

public static void main(String[] args) {
    launch(args);
}
}

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;

public class DatabaseConnection {
    private static final String URL =
"jdbc:mysql://localhost:3306/fitness_tracker"; // Database URL
    private static final String USER = "root"; // MySQL username
    private static final String PASSWORD = "manu"; // MySQL password

    // Method to establish a connection to the database
    public static Connection connect() {
        try {
            return DriverManager.getConnection(URL, USER, PASSWORD);
        } catch (SQLException e) {
            System.out.println("Database connection error: " + e.getMessage());
            return null;
        }
    }

    // Main method - Entry point of the application
    public static void main(String[] args) {
        // Attempt to establish a connection
        Connection connection = connect();
```

```
// Check if the connection was successful
if (connection != null) {
    System.out.println("Connection to the database established
successfully!");
    // Close the connection after use (optional for this demo, but always
good practice)
    try {
        connection.close();
    } catch (SQLException e) {
        System.out.println("Error closing the connection: " +
e.getMessage());
    }
} else {
    System.out.println("Failed to connect to the database.");
}
}

import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.scene.text.Font;
import javafx.stage.Stage;
import java.sql.*;

public class FoodTrackingPage extends Application {

    @Override
    public void start(Stage stage) {
        // Title for the page
        Label titleLabel = new Label("Food Tracking");
        titleLabel.setFont(new Font("Arial", 24));

        // Input fields for food type, quantity, and time
        Label foodLabel = new Label("Enter food item:");
        TextField foodField = new TextField();

        Label quantityLabel = new Label("Enter quantity (grams):");
        TextField quantityField = new TextField();

        Label timeLabel = new Label("Enter time of consumption:");
        TextField timeField = new TextField();
    }
}
```

```

// Button to log food
Button logFoodButton = new Button("Log Food");
logFoodButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px;");

// Label to display the result
Label resultLabel = new Label("Food entry will be displayed here.");
resultLabel.setStyle("-fx-font-size: 16px; -fx-font-weight: bold;");

// Action for the "Log Food" button
logFoodButton.setOnAction(e -> {
    String foodItem = foodField.getText();
    String quantity = quantityField.getText();
    String time = timeField.getText();

    if (!foodItem.isEmpty() && !quantity.isEmpty() && !time.isEmpty()) {
        // Save food entry to MySQL database
        saveFoodEntry(foodItem, Integer.parseInt(quantity), time);
        resultLabel.setText("Logged: " + foodItem + " (" + quantity + "g")
at " + time);
    } else {
        resultLabel.setText("Please fill in all fields.");
    }
});

// Back button to go to the previous page
Button backButton = new Button("Back");
backButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px;");
backButton.setOnAction(e -> stage.close()); // Close the current page,
modify this as per your navigation logic

// Layout for the page
VBox layout = new VBox(15);
layout.setAlignment(Pos.CENTER);
layout.getChildren().addAll(
    titleLabel,
    foodLabel,
    foodField,
    quantityLabel,
    quantityField,
    timeLabel,
    timeField,
    logFoodButton,
    resultLabel,
    backButton
);

```

```
// Set up the scene
Scene scene = new Scene(layout, 400, 300);
stage.setTitle("Food Tracking");
stage.setScene(scene);
stage.show();
}

private void saveFoodEntry(String foodItem, int quantity, String time) {
    // Database connection details
    String url = "jdbc:mysql://localhost:3306/fitness_tracker"; // Replace
with your actual database name
    String user = "root"; // Replace with your MySQL username
    String password = "manu"; // Replace with your MySQL password

    String query = "INSERT INTO food_tracking (user_id, food_name, quantity,
consumption_time, date) VALUES (?, ?, ?, ?, CURDATE())";

    try (Connection conn = DriverManager.getConnection(url, user, password);
        PreparedStatement stmt = conn.prepareStatement(query)) {

        // Assuming a hardcoded user_id for now (you'll likely get this from
the logged-in user)
        int userId = 1;

        stmt.setInt(1, userId);
        stmt.setString(2, foodItem);
        stmt.setInt(3, quantity);
        stmt.setString(4, time); // Assuming time is in a valid format
(HH:mm)

        stmt.executeUpdate();
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

public static void main(String[] args) {
    launch(args);
}
}

import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
```

```
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.scene.text.Font;
import javafx.stage.Stage;
import java.sql.*;

public class MedicalConditionTrackingPage extends Application {

    @Override
    public void start(Stage stage) {
        // Title for the page
        Label titleLabel = new Label("Medical Condition Tracking");
        titleLabel.setFont(new Font("Arial", 24));

        // Input fields for condition name, severity, and medications
        Label conditionLabel = new Label("Enter medical condition:");
        TextField conditionField = new TextField();

        Label severityLabel = new Label("Enter severity (1-10):");
        TextField severityField = new TextField();

        Label medicationLabel = new Label("Enter medication (optional):");
        TextField medicationField = new TextField();

        // Button to log condition
        Button logConditionButton = new Button("Log Condition");
        logConditionButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px;");

        // Label to display the result
        Label resultLabel = new Label("Condition entry will be displayed here.");
        resultLabel.setStyle("-fx-font-size: 16px; -fx-font-weight: bold;");

        // Action for the "Log Condition" button
        logConditionButton.setOnAction(e -> {
            String condition = conditionField.getText();
            String severity = severityField.getText();
            String medication = medicationField.getText();

            if (!condition.isEmpty() && !severity.isEmpty()) {
                // Save the condition to the database
                logConditionToDatabase(condition, severity, medication);
                resultLabel.setText("Logged: " + condition + " (Severity: " +
severity + ")");
                if (!medication.isEmpty()) {

```

```

        resultLabel.setText(resultLabel.getText() + " | Medication: "
+ medication);
    }
} else {
    resultLabel.setText("Please fill in all fields.");
}
});

// Back button to go to the previous page
Button backButton = new Button("Back");
backButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px;");
backButton.setOnAction(e -> stage.close()); // Close the current page

// Layout for the page
VBox layout = new VBox(15);
layout.setAlignment(Pos.CENTER);
layout.getChildren().addAll(
    titleLabel,
    conditionLabel,
    conditionField,
    severityLabel,
    severityField,
    medicationLabel,
    medicationField,
    logConditionButton,
    resultLabel,
    backButton
);

// Set up the scene
Scene scene = new Scene(layout, 400, 300);
stage.setTitle("Medical Condition Tracking");
stage.setScene(scene);
stage.show();
}

// Method to log the medical condition into the MySQL database
private void logConditionToDatabase(String condition, String severity, String
medication) {
    String url = "jdbc:mysql://localhost:3306/fitness_tracker";
    String user = "root";
    String pass = "manu";

    try (Connection conn = DriverManager.getConnection(url, user, pass)) {

```

```
        String query = "INSERT INTO medical_conditions (condition_name, severity, medication) VALUES (?, ?, ?)";
        try (PreparedStatement stmt = conn.prepareStatement(query)) {
            stmt.setString(1, condition);
            stmt.setInt(2, Integer.parseInt(severity));
            stmt.setString(3, medication.isEmpty() ? null : medication);
            stmt.executeUpdate();
        }
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

public static void main(String[] args) {
    launch(args);
}
}

import javafx.application.Application;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.scene.text.Font;
import javafx.stage.Stage;
import java.sql.*;

public class UserProfilePage extends Application {

    @Override
    public void start(Stage stage) {
        // Title for the page
        Label titleLabel = new Label("User Profile");
        titleLabel.setFont(new Font("Arial", 24));

        // User information fields
        Label nameLabel = new Label("Full Name:");
        TextField nameField = new TextField();

        Label emailLabel = new Label("Email Address:");
        TextField emailField = new TextField();

        Label passwordLabel = new Label("Password:");
        PasswordField passwordField = new PasswordField();
    }
}
```

```
// Profile picture (for simplicity, use a placeholder)
Label pictureLabel = new Label("Profile Picture:");
Button changePicButton = new Button("Change Picture");

// Button to save updated profile
Button saveButton = new Button("Save Profile");
saveButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px;");

// Label to show save status
Label statusLabel = new Label();
statusLabel.setStyle("-fx-font-size: 16px; -fx-font-weight: bold;");

// Action for the "Save Profile" button
saveButton.setOnAction(e -> {
    String name = nameField.getText();
    String email = emailField.getText();
    String password = passwordField.getText();

    // Insert or update user profile in the database
    if (!name.isEmpty() && !email.isEmpty() && !password.isEmpty()) {
        saveUserProfile(name, email, password);
        statusLabel.setText("Profile updated successfully!");
    } else {
        statusLabel.setText("Please fill in all fields.");
    }
});

// Back button to go to the previous page
Button backButton = new Button("Back");
backButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px;");
backButton.setOnAction(e -> stage.close()); // Close the current page

// Layout for the page
VBox layout = new VBox(15);
layout.setAlignment(Pos.CENTER);
layout.getChildren().addAll(
    titleLabel,
    nameLabel,
    nameField,
    emailLabel,
    emailField,
    passwordLabel,
    passwordField,
    pictureLabel,
    changePicButton,
```

```

        saveButton,
        statusLabel,
        backButton
    );

    // Set up the scene
    Scene scene = new Scene(layout, 400, 350);
    stage.setTitle("User Profile");
    stage.setScene(scene);
    stage.show();
}

private void saveUserProfile(String name, String email, String password) {
    String url = "jdbc:mysql://localhost:3306/fitness_tracker";
    String user = "root";
    String pass = "manu";

    try (Connection conn = DriverManager.getConnection(url, user, pass)) {
        String query = "INSERT INTO user_profiles (full_name, email,
password) VALUES (?, ?, ?)";
        try (PreparedStatement stmt = conn.prepareStatement(query)) {
            stmt.setString(1, name);
            stmt.setString(2, email);
            stmt.setString(3, password); // You should hash the password in
a real app
            stmt.executeUpdate();
        }
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

public static void main(String[] args) {
    launch(args);
}
}

import javafx.application.Application;
import javafx.collections.FXCollections;
import javafx.collections.ObservableList;
import javafx.geometry.Pos;
import javafx.scene.Scene;
import javafx.scene.control.*;
import javafx.scene.layout.*;
import javafx.scene.text.Font;

```

```
import javafx.stage.Stage;
import java.sql.*;

public class WaterTrackingPage extends Application {

    private ObservableList<WaterIntakeItem> data =
FXCollections.observableArrayList();

    @Override
    public void start(Stage stage) {
        // Title for the page
        Label titleLabel = new Label("Water Tracking");
        titleLabel.setFont(new Font("Arial", 24));

        // Water intake entry field
        Label intakeLabel = new Label("Enter Water Intake (ml):");
        TextField intakeField = new TextField();

        // Button to log water intake
        Button logButton = new Button("Log Water");
        logButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px;");

        // Label to show intake status
        Label statusLabel = new Label();
        statusLabel.setStyle("-fx-font-size: 16px; -fx-font-weight: bold;");

        // Table for displaying water tracking history
        TableView<WaterIntakeItem> table = new TableView<>();
        TableColumn<WaterIntakeItem, String> dateColumn = new
TableColumn<>("Date");
        TableColumn<WaterIntakeItem, String> intakeColumn = new
TableColumn<>("Intake (ml)");

        // Adding columns to the table
        table.getColumns().add(dateColumn);
        table.getColumns().add(intakeColumn);

        // Load data from MySQL
        loadWaterData();

        // Setting items to the table
        table.setItems(data);

        // Action for the "Log Water" button
        logButton.setOnAction(e -> {
```

```

        String intake = intakeField.getText();
        if (!intake.isEmpty() && isNumeric(intake)) {
            // Save the water intake to the database
            logWaterIntake(intake);
            statusLabel.setText("Water logged: " + intake + " ml");
            loadWaterData(); // Reload data from the database
        } else {
            statusLabel.setText("Please enter a valid number.");
        }
    });

    // Back button to go to the previous page
    Button backButton = new Button("Back");
    backButton.setStyle("-fx-font-family: 'Arial'; -fx-font-size: 14px;");
    backButton.setOnAction(e -> stage.close()); // Close the current page

    // Layout for the page
    VBox layout = new VBox(15);
    layout.setAlignment(Pos.CENTER);
    layout.getChildren().addAll(
        titleLabel,
        intakeLabel,
        intakeField,
        logButton,
        statusLabel,
        table,
        backButton
    );

    // Set up the scene
    Scene scene = new Scene(layout, 500, 400);
    stage.setTitle("Water Tracking");
    stage.setScene(scene);
    stage.show();
}

// Helper method to check if the input is a number
private boolean isNumeric(String str) {
    try {
        Integer.parseInt(str);
        return true;
    } catch (NumberFormatException e) {
        return false;
    }
}

```

```
// Method to load water intake data from the database
private void loadWaterData() {
    data.clear(); // Clear the existing data
    String url = "jdbc:mysql://localhost:3306/your_database";
    String user = "root";
    String pass = "your_password";

    try (Connection conn = DriverManager.getConnection(url, user, pass)) {
        String query = "SELECT date, intake FROM water_intake ORDER BY date
DESC";
        try (Statement stmt = conn.createStatement(); ResultSet rs =
stmt.executeQuery(query)) {
            while (rs.next()) {
                String date = rs.getString("date");
                String intake = rs.getString("intake");
                data.add(new WaterIntakeItem(date, intake));
            }
        }
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

// Method to log water intake into the database
private void logWaterIntake(String intake) {
    String url = "jdbc:mysql://localhost:3306/fitness_tracker";
    String user = "root";
    String pass = "manu";

    try (Connection conn = DriverManager.getConnection(url, user, pass)) {
        String query = "INSERT INTO water_intake (date, intake) VALUES
(CURDATE(), ?)";
        try (PreparedStatement stmt = conn.prepareStatement(query)) {
            stmt.setInt(1, Integer.parseInt(intake));
            stmt.executeUpdate();
        }
    } catch (SQLException ex) {
        ex.printStackTrace();
    }
}

public static void main(String[] args) {
    launch(args);
}
```

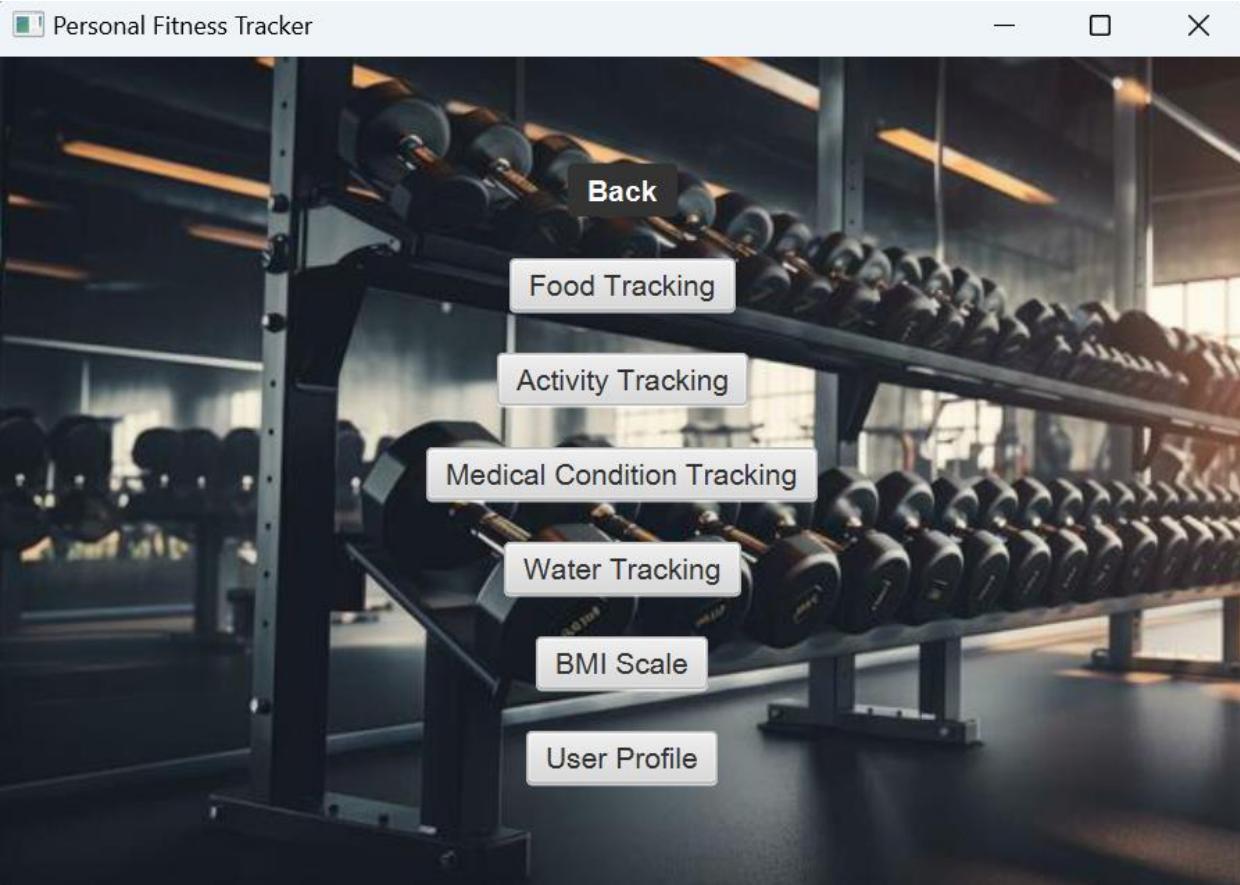
```
// Class for representing each water intake item
public static class WaterIntakeItem {
    private String date;
    private String intake;

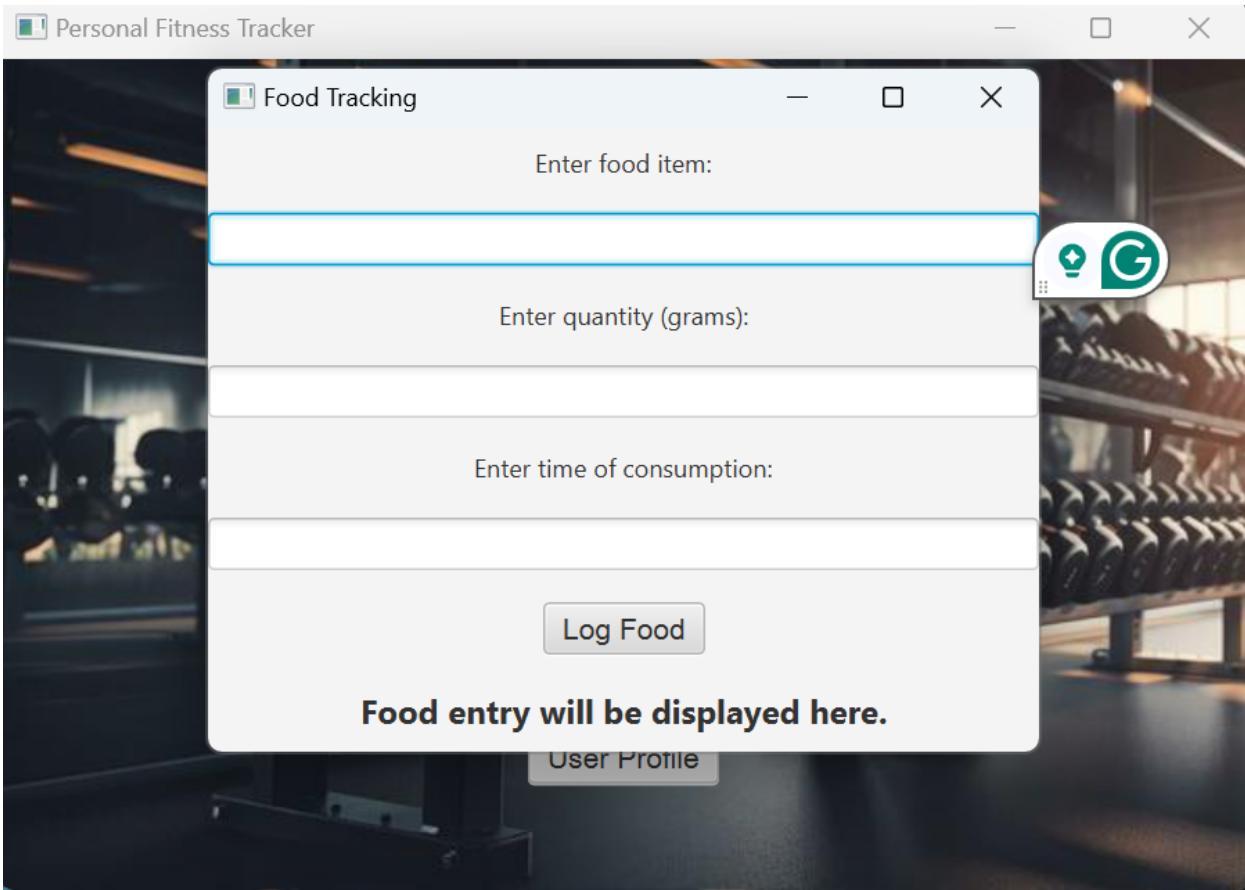
    public WaterIntakeItem(String date, String intake) {
        this.date = date;
        this.intake = intake;
    }

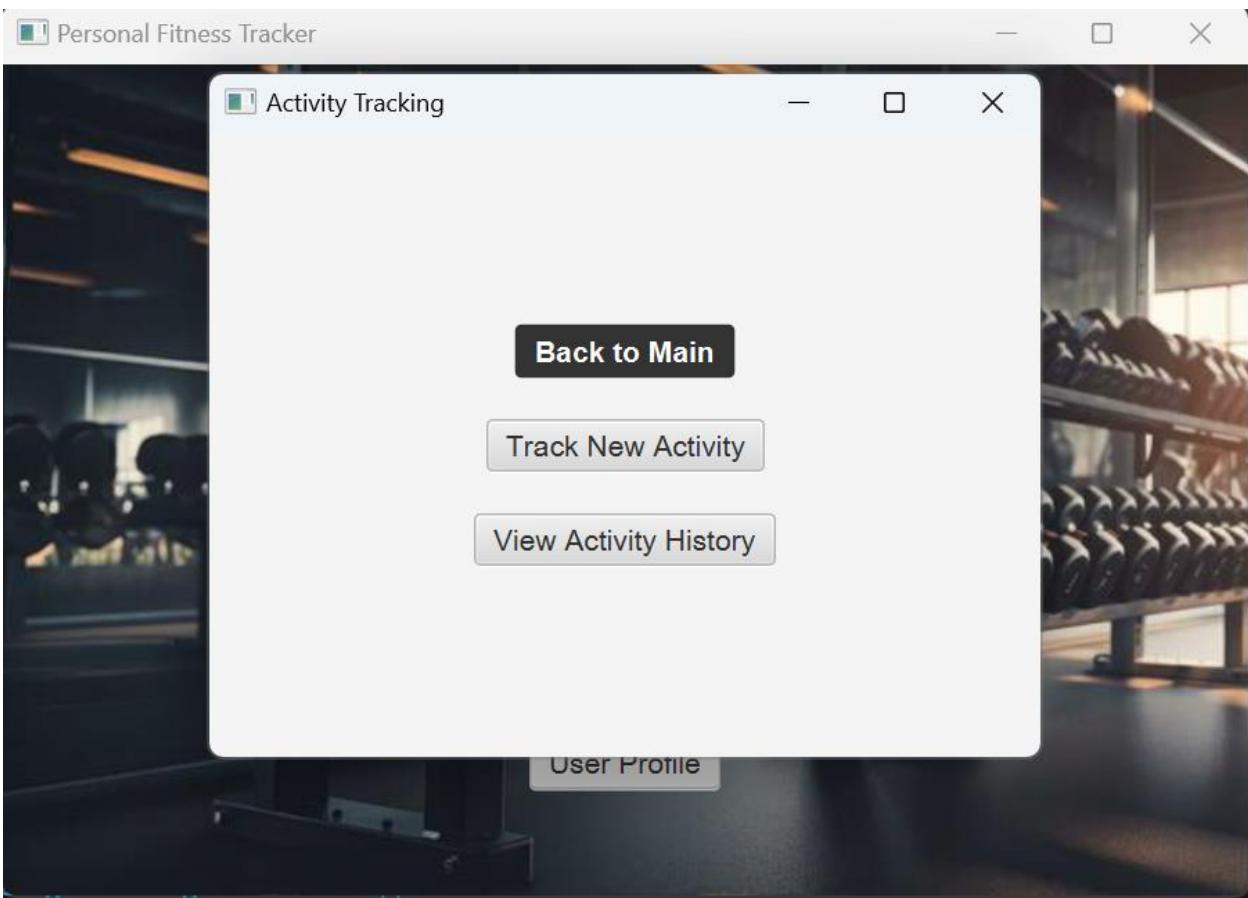
    public String getDate() {
        return date;
    }

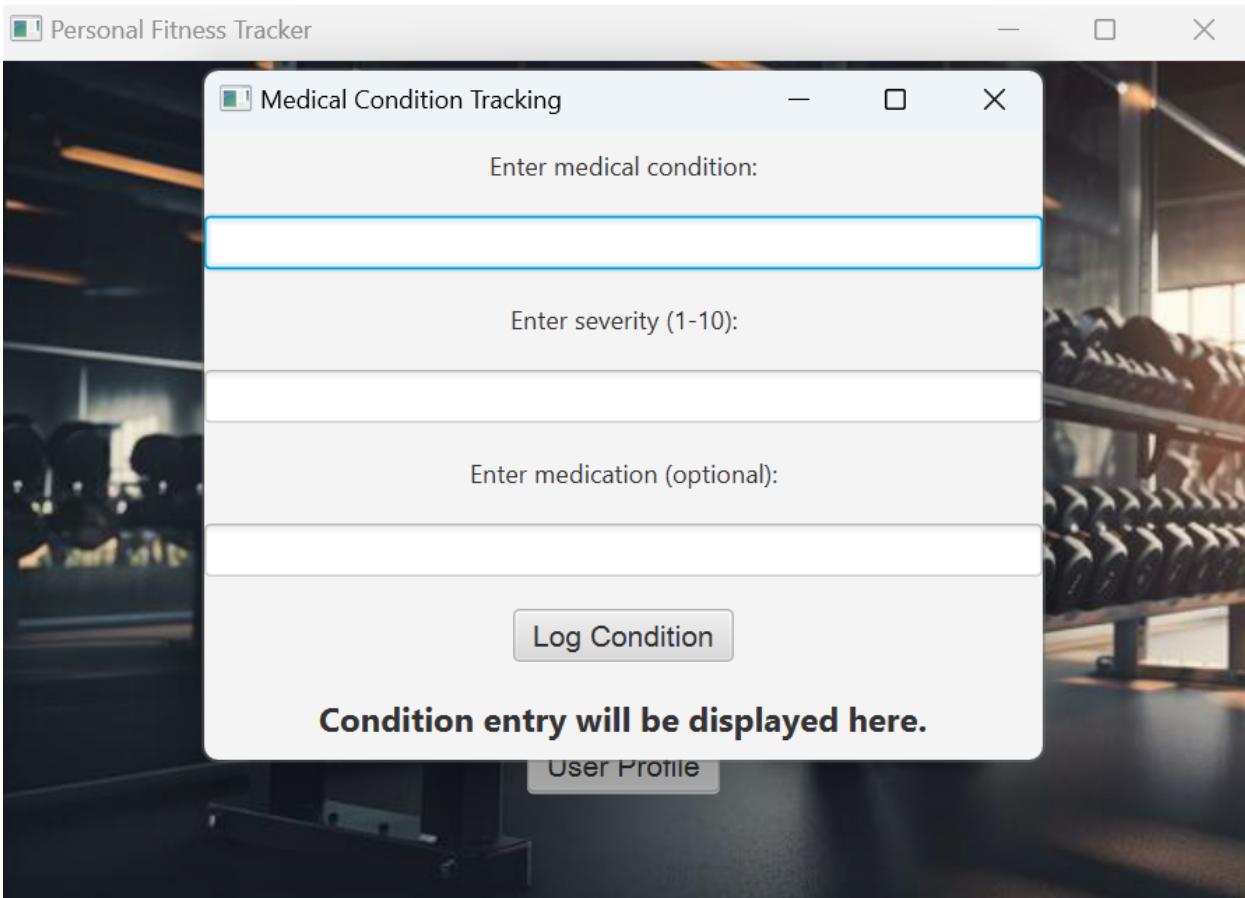
    public String getIntake() {
        return intake;
    }
}
```

Output:









Pers

Water Tracking



Water Tracking

Enter Water Intake (ml):

Log Water

Date	Intake (ml)
No content in table	

Back

