

# **RAJALAKSHMI ENGINEERING COLLEGE**

**[AUTONOMOUS]**

**RAJALAKSHMI NAGAR, THANDALAM – 602 105**



**RAJALAKSHMI  
ENGINEERING COLLEGE**

**CS23333 OBJECT ORIENTED PROGRAMMING USING JAVA**

**Laboratory Record Note Book**

Name : . SRI SANJANA B .....

Year / Branch / Section : . . II/IT/D.....

College Roll No. : ..... 2116231001214.....

Semester : ..... III.....

Academic Year : ..... 2024-2025 .....

**RAJALAKSHMI ENGINEERING COLLEGE**  
**[AUTONOMOUS]**

**RAJALAKSHMI NAGAR, THANDALAM – 602 105**

**BONAFIDE CERTIFICATE**

Name : . . . . . SRI SANJANA B . . . . .

Academic Year : 2024-2025 Semester: III Branch : IT-D

2116231001214

**Register No.**

Certified that this is the bonafide record of work done by the above student in the CS23333 –Object Oriented Programming using JAVA during the year 2024 - 2025.

**Signature of Faculty in-charge**

Submitted for the Practical Examination held on . . . . . 27/11/2024. . . . .

**Internal Examiner**

**External Examiner**

## INDEX

<b>Lab Week</b>	<b>Date</b>	<b>Name of the Experiment</b>	<b>Page No</b>	<b>Signature</b>
1	20.9.24	Java Architecture, Language Basics	1	
2	20.9.24	Flow Control Statements	5	
3	21.9.24	Arrays	11	
4	1.10.24	Classes And Objects	17	
5	1.10.24	Inheritance	23	
6	3.10.24	String, StringBuffer	29	
7	3.10.24	Interfaces	35	
8	6.10.24	Polymorphism, Abstract Classes, Final Keyword	41	
9	9.10.24	Exceptional Handling	47	
10	4.10.24	Collection - List	52	
11	10.11.24	Set, Map	57	
12	10.11.24	Introduction to I/O, I/O Operations, Object Serialization	63	
13	27.11.24	Java Project Report	72	

**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 Odd
3 {
4     public static void main(String[] args)
5     {
6         Scanner sc=new Scanner(System.in);
7         int n=sc.nextInt();
8         if(n%2!=0)
9         {
10             System.out.println(2);
11         }
12         else
13         {
14             System.out.println(1);
15         }
16     }
17 }
```

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

Passed all tests! ✓

**Question 1**

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 lastDigit
3 {
4     public static void main(String[] args)
5     {
6         Scanner sc=new Scanner(System.in);
7         int n=sc.nextInt();
8         int a=n%10;
9         int b=Math.abs(a);
10        System.out.println(b);
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 public class Sum
3 {
4     public static void main(String[] args)
5     {
6         Scanner sc=new Scanner(System.in);
7         int a=sc.nextInt();
8         int b=sc.nextInt();
9         int c=Math.abs(a%10);
10        int d=Math.abs(b%10);
11        System.out.println(c+d);
12    }
13 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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? ►

**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 Pattern
3 {
4     public static void main(String[] args)
5     {
6         Scanner sc=new Scanner(System.in);
7         int n=sc.nextInt();
8         int[] arr=new int[n];
9         arr[0]=0;
10        arr[1]=1;
11        arr[2]=1;
12        for(int i=3;i<n;i++)
13        {
14            arr[i]=arr[i-1]+arr[i-2]+arr[i-3];
15        }
16        System.out.println(arr[n-1]);
17    }
18 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	5	4	4	✓
✓	8	24	24	✓
✓	11	149	149	✓

Passed all tests! ✓

**Question 2**

Correct

Marked out of 5.00

You have recently seen a motivational sports movie and want to start exercising regularly. Your coach tells you that it is important to get up early in the morning to exercise. She sets up a schedule for you:

On weekdays (Monday - Friday), you have to get up at 5:00. On weekends (Saturday & Sunday), you can wake up at 6:00. However, if you are on vacation, then you can get up at 7:00 on weekdays and 9:00 on weekends.

Write a program to print the time you should get up.

**Input Format**

Input containing an integer and a boolean value.

The integer tells you the day it is (1-Sunday, 2-Monday, 3-Tuesday, 4-Wednesday, 5-Thursday, 6-Friday, 7-Saturday). The boolean is true if you are on vacation and false if you're not on vacation.

You have to print the time you should get up.

**Example Input:**

1 false

**Output:**

6:00

**Example Input:**

5 false

**Output:**

5:00

**Example Input:**

1 true

**Output:**

9:00

**For example:**

Input	Result
1 false	6:00
5 false	5:00
1 true	9:00

**Answer:** (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class Sports
3 {
4     public static void main(String[] main)
5     {
6         Scanner sc=new Scanner(System.in);
7         int n=sc.nextInt();
8         boolean b=sc.nextBoolean();
9         if( (n==6 || n==2 || n==3 || n==4|| n==5) && b==false )
10        {
11            System.out.println("5:00");
12        }
13        else if( (n==1 || n==7) && b==false)
14        {
15            System.out.println("6:00");
16        }
    }
}

```

```
17     else if( (n==6 || n==2 || n==3 || n==4 || n==5) && b==true)
18     {
19         System.out.println("7:00");
20     }
21     else if( (n==1 || n==7) && b==true )
22     {
23         System.out.println("9:00");
24     }
25 }
26 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	1 false	6:00	6:00	✓
✓	5 false	5:00	5:00	✓
✓	1 true	9:00	9:00	✓

Passed all tests! ✓



**Question 3**

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 Movie
3 {
4     public static void main(String[] args)
5     {
6         Scanner sc=new Scanner(System.in);
7         int acting=sc.nextInt();
8         int rating=sc.nextInt();
9         if(acting<2 || rating<2)
10        {
11            System.out.println("No");
12        }
13        else if(acting>8 || rating>8)
14        {
15            System.out.println("Yes");
}

```

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

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	9 5	Yes	Yes	✓
✓	1 9	No	No	✓
✓	6 4	Maybe	Maybe	✓

Passed all tests! ✓

◀ Lab-02-MCQ

Jump to...

Lab-03-MCQ ►

**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 Main
3 {
```

```

4   public static void main(String[] args)
5   {
6       Scanner sc=new Scanner(System.in);
7       int n=sc.nextInt();
8       int[] arr=new int[n];
9       for(int i=0;i<n;i++)
10      {
11          arr[i]=sc.nextInt();
12
13      }
14      int maxLen=0,currentLen=0,maxSum=0;
15      int currentSum=0;
16      boolean hasPositive=false;
17      for(int i=0;i<n;i++)
18      {
19          if(arr[i]>=0)
20          {
21              hasPositive=true;
22              currentLen++;
23              currentSum+=arr[i];
24
25          }
26          else
27          {
28              if(currentLen>maxLen)
29              {
30                  maxLen=currentLen;
31                  maxSum=currentSum;
32              }
33              else if(currentLen==maxLen)
34              {
35                  maxSum+=currentSum;
36              }
37              currentLen=0;
38              currentSum=0;
39
40          }
41      }
42      if(currentLen>maxLen)
43      {
44          maxSum=currentSum;
45      }
46      else if(currentLen==maxLen)
47      {
48          maxSum+=currentSum;
49
50      }
51      if(!hasPositive)
52      {

```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 {
4     public static void main(String[] args)
5     {
6         Scanner sc=new Scanner(System.in);
7         int input1=sc.nextInt();
8         int[] input2=new int[input1];
9         for(int i=0;i<input1;i++)
10        {
11            input2[i]=sc.nextInt();
12        }
13        int max=Integer.MIN_VALUE;
14        for(int i=0;i<input1;i++)
15        {
16            if(input2[i]>max)
17            {
18                max=input2[i];
19            }
20        }
21        for(int i=0;i<input1;i++)
22        {
23            input2[i]=(input2[i]-max)*max;
24        }
25        for(int i=0;i<input1;i++)
26        {
27            System.out.print(input2[i]+" ");
28        }
29    }
30 }
```

	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 0<sup>th</sup> index of the array pick up digits as per below:

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

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

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

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

4<sup>th</sup> 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 Main
3 {
4     public static void main(String[] args)
5     {
6         Scanner sc=new Scanner(System.in);
7         int input1=sc.nextInt();
8         int[] input2=new int[input1];
9         for(int i=0;i<input1;i++)
10        {
11            input2[i]=sc.nextInt();
12        }
13    }
14    int[] resultArray=new int[input1];
15    for(int i=0;i<input1;i++)
16    {
17        String numStr=String.valueOf(input2[i]);
18        int position=numStr.length()-1-i;
19        if(position>=0)
20        {
21            resultArray[i]=Character.getNumericValue(numStr.charAt(position));
22        }
23        else
24        {
25            resultArray[i]=0;
26        }
27    }
28    int sum=0;
29    for(int i=0;i<input1;i++)
30    {
31        resultArray[i]*=resultArray[i];
32        sum+=resultArray[i];
33    }
34    System.out.println(sum);
35 }
36 }
```

	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 ►

**Question 1**

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 =  $\pi r^2$**

**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.io.*;
2 import java.util.Scanner;
3 class Circle
4 {
5     private double radius;
6     public Circle(double radius){
7         // set the instance variable radius
8         this.radius=radius;
9     }
10    }
11    public void setRadius(double radius){
12        // set the radius
13        this.radius=radius;
14    }
15    }
16    public double getRadius()    {
17        // return the radius
18        return radius;
19    }
20    }
21    public double calculateArea() { // complete the below statement
22        return Math.PI*radius*radius;
23    }
24    }
25    public double calculateCircumference()    {
26        // complete the statement
27        return 2*Math.PI*radius;
28    }
29    }
30    }
31 }
32 }
33 class prog{
34     public static void main(String[] args)  {
35         int r;
36         Scanner sc= new Scanner(System.in);
37         r=sc.nextInt();
38         Circle c= new Circle(r);
39         System.out.println("Area = "+String.format("%.2f", c.calculateArea()));
40         System.out.println("Circumference = "+String.format("%.2f",c.calculateCircumference()));
41     }
42 }
```

```
41  
42  
43 }  
44 }  
45 }
```

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

Passed all tests! ✓

**Question 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 class prog
2 {
3     private String manufacturer;
4     private String operating_system;
5     private String color;
6     private int cost;
7     public prog(String manufacturer, String operating_system, String color, int cost)
8     {
9         this.manufacturer= manufacturer;
10        this.operating_system= operating_system;
11        this.color= color;
12        this.cost= cost;
13    }
14    public void setManufacturer(String manufacturer)
15    {
16        this.manufacturer= manufacturer;
17    }
18    public void setOperating_system(String operating_system)
19    {
20        this.operating_system= operating_system;
21    }
22    public void setColor(String color)
23    {
24        this.color= color;
25    }
26    public void setCost(int cost)
27    {
28        this.cost= cost;
29    }
30    public String getManufacturer()
31    {
32        return manufacturer;
33    }
34}
```

```
34     }
35     public String getOperating_system()
36     {
37         return operating_system;
38     }
39     public String getColor()
40     {
41         return color;
42     }
43     public int getCost()
44     {
45         return cost;
46     }
47     public String toString()
48     {
49         return "manufacturer = "+manufacturer+"\n"+
50             "operating_system = "+operating_system+"\n"+
51             "color = "+color+"\n"+
52             "cost = "+cost;
53 }
```

	<b>Test</b>	<b>Expected</b>	<b>Got</b>	
✓	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 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 class prog
2 {
3     private String name;
4     private int rollno;
5     public prog()
6     {
7         this.name=null;
8         this.rollno=0;
9         System.out.println("No-arg constructor is invoked");
10    }
11    public prog(String name)
12    {
13        this.name=name;
14        this.rollno=0;
15        System.out.println("1 arg constructor is invoked");
16    }
17    public prog(String name,int rollno)
18    {
19        this.name=name;
20        this.rollno=rollno;
21        System.out.println("2 arg constructor is invoked");
22    }
23    public void display()
24    {
25        System.out.println("Name =" +name+ " , Roll no = "+rollno);
26    }
27    public static void main(String[] args)
28    {
29        prog stu1=new prog();
30        prog stu2=new prog("Rajalakshmi");
31        prog stu3=new prog("Lakshmi",101);
32        stu1.display();

```

```
-- 33     stu2.display();  
34     stu3.display();  
35 }  
36 }
```

	<b>Test</b>	<b>Expected</b>	<b>Got</b>	
✓	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! ✓

◀ Lab-04-MCQ

Jump to...

Number of Primes in a specified range ►

**Question 1**

Correct

Marked out of 5.00

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

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

**For example:**

**Result**

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

**Answer:** (penalty regime: 0 %)

[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     BankAccount(String ac, double bal){  
10         accountNumber = ac;  
11         balance = bal;  
12     }  
13     // Method to deposit an amount into the account  
14     public void deposit(double amount) {  
15         // Increase the balance by the deposit amount  
16         balance += amount;  
17     }  
18  
19     // Method to withdraw an amount from the account  
20     public void withdraw(double amount) {  
21         // Check if the balance is sufficient for the withdrawal  
22         if (balance >= amount) {  
23             // Decrease the balance by the withdrawal amount  
24             balance -= amount;  
25         } else {  
26             // Print a message if the balance is insufficient  
27             System.out.println("Insufficient balance");  
28         }  
29     }  
30  
31     // Method to get the current balance  
32     public double getBalance() {  
33         // Return the current balance  
34         return balance;  
35     }  
36  
37 }  
38  
39 class SavingsAccount extends BankAccount {  
40     // Constructor to initialize account number and balance  
41     public SavingsAccount(String accountNumber, double balance) {  
42 }
```

```

43     // Call the parent class constructor
44     super(accountNumber,balance);
45 }
46 // Override the withdraw method from the parent class
47 @Override
48 public void withdraw(double amount) {
49     // Check if the withdrawal would cause the balance to drop below $100
50     if (getBalance() - amount < 100) {
51         // Print a message if the minimum balance requirement is not met
52         System.out.println("Minimum balance of $100 required!");

```

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

Passed all tests! ✓

**Question 2**

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 %)

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

```

28  public String toString(){
29      // return the details of the student
30      return "CollegeName : "+collegeName+"\nStudentName : "+studentName+"\nDepartment : "+department;
31
32 }
33 }
34 // class cse extends Student{
35 //     private String department;
36 // void course(){
37
38 // }
39 // }
40 class prog {
41 public static void main (String[] args) {
42     Student s1 = new Student("REC","Venkatesh","CSE");
43     s1.admitted();                                // invoke the admitted() method
44     System.out.println(s1.toString());
45 }
46 }
47 }
```

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

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 class Moblie{
2     Moblie(){
3         System.out.println("Basic Mobile is Manufactured");
4     }
5     // void basicMoblie(){
6     // }
7 }
8 class CamaraMoblie extends Moblie{
9     CamaraMoblie(){
10        super();
11        System.out.println("Camera Mobile is Manufactured");
12    }
13    void newFeature(){
14        System.out.println("Camera Mobile with 5MG px");
15    }
16 }
17 class AndroidMoblie extends CamaraMoblie{
18     AndroidMoblie(){
19        super();
20        System.out.println("Android Mobile is Manufactured");
21    }
22    void androidMoblie(){
23        System.out.println("Touch Screen Mobile is Manufactured");
24    }
25 }
26 }
27 }
28 public class prog{
```

```
29 public static void main(String A[]){
30     AndroidMobile a = new AndroidMobile();
31     a.newFeature();
32     a.androidMobile();
33 }
34 }
```

	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? ►

**Question 1**

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 2<sup>nd</sup> 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 25<sup>th</sup> 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
3 public class WordProcessor {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         String input = scanner.nextLine();
7         String[] words = input.split(":");
8
9         StringBuilder output = new StringBuilder();
10
11        for (String word : words) {
12            if (word.length() == 2) {
13                char firstChar = word.charAt(0);
14                char secondChar = word.charAt(1);
15
16                if (firstChar == secondChar) {
17                    output.append(firstChar);
18                } else {
19                    int position1 = firstChar - 'a' + 1;
20                    int position2 = secondChar - 'a' + 1;
21                    int maxPos = Math.max(position1, position2);
22                    int minPos = Math.min(position1, position2);
23                    int newPos = maxPos - minPos;
24                    char newChar = (char) ('a' + newPos - 1);
25                    output.append(newChar);
26                }
27            }
28        }
29        System.out.println(output.toString().toUpperCase());
30
31        scanner.close();
32    }
33 }
```

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

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
3 public class WordProcessor {
    ...
}
```

```

4  public static void main(String[] args) {
5      Scanner scanner = new Scanner(System.in);
6      String inputString = scanner.nextLine();
7      int inputNumber = scanner.nextInt();
8
9      String processedString = processWords(inputString, inputNumber);
10     System.out.println(processedString);
11 }
12
13 public static String processWords(String inputString, int inputNumber) {
14     String[] words = inputString.split(" ");
15
16     if (words.length < 2 || inputNumber < 11 || inputNumber > 99) {
17         return "Invalid input";
18     }
19
20     int word1Index = inputNumber / 10 - 1;
21     int word2Index = inputNumber % 10 - 1;
22
23     if (word1Index >= words.length || word2Index >= words.length) {
24         return "Invalid word indices";
25     }
26
27     String word1 = words[word1Index];
28     String word2 = words[word2Index];
29
30     String processedWord1 = processWord(word1);
31     String processedWord2 = processWord(word2);
32
33     return processedWord1 + " " + processedWord2;
34 }
35
36 public static String processWord(String word) {
37     int middleIndex = word.length() / 2;
38     StringBuilder input = new StringBuilder();
39     StringBuilder input1 = new StringBuilder();
40     input.append(word.substring(0,middleIndex+1));
41     input1.append(word.substring(0,middleIndex));
42     if(word.length()%2==0){
43         return input1.reverse()+word.substring(middleIndex);
44     }
45     else{
46         return input.reverse()+word.substring(middleIndex);
47     }
48 }
49 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 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.HashSet;
2 import java.util.Scanner;
3 import java.util.Set;
4 import java.util.stream.Collectors;
5
6 public class StringProcessor {
7     public static void main(String[] args) {
8         Scanner scanner = new Scanner(System.in);
9         String input1 = scanner.nextLine();
10        String input2 = scanner.nextLine();
11        String result = processStrings(input1, input2);
12        System.out.println(result);
13    }
14    public static String processStrings(String input1, String input2) {
15        if ((input1 == null || input1.trim().isEmpty()) && (input2 == null || input2.trim().isEmpty())) {
16            return "null";
17        }
18        String concatenated = input1 + input2;
19        Set<Character> uniqueChars = new HashSet<>();
20        for (char c : concatenated.toCharArray()) {

```

```
21     if (c != ' ') {  
22         uniqueChars.add(c);  
23     }  
24     String sortedChars = uniqueChars.stream().map(String::valueOf).sorted((a, b) -> b.compareTo(a)).collect(Co  
25     return sortedChars;  
26 }  
27  
28 }  
29
```

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

Passed all tests! ✓

◀ Lab-06-MCQ

Jump to...

Return second word in Uppercase ►

**Question 1**

Correct

Marked out of 5.00

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

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

RBI interface has two more methods default and static method.

```
default void policyNote() {
    System.out.println("RBI has a new Policy issued in 2023.");
}

static void regulations(){
    System.out.println("RBI has updated new regulations on 2024.");
}
```

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

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

**Sample Input/Output:**

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

**For example:**

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

**Answer:** (penalty regime: 0 %)

```
1 interface RBI {
2     String parentBank = "RBI";
3     double rateOfInterest();
4     default void policyNote() {
5         System.out.println(parentBank + " has a new Policy issued in 2023");
6     }
7     static void regulations() {
8         System.out.println(parentBank + " has updated new regulations in 2024.");
9     }
10 }
11 class SBI implements RBI {
12     @Override
13     public double rateOfInterest() {
14         return 7.6;
15     }
16 }
17 class Karur implements RBI {
18     @Override
19     public double rateOfInterest() {
20         return 7.4;
21     }
22 }
23 public class BankPolicies {
24     public static void main(String[] args) {
25         RBI sbi = new SBI();
26         RBI karur = new Karur();
27         sbi.policyNote();
28         RBI.regulations();
29         System.out.println("SBI rate of interest: " + sbi.rateOfInterest() + " per annum.");
30     }
31 }
```

```
30     System.out.println("Karur rate of interest: " + karur.rateOfInterest() + " per annum.");
31 }
32 }
```

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

Passed all tests! ✓



**Question 2**

Correct

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 Playing volleyball 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     String name;
7
8 ↓     public Football(String name) {
9         this.name = name;
10    }
11
12    @Override
13 ↓     public void play() {
14         System.out.println(name + " is Playing football");
15    }
16 }
17
18 ↓ class Volleyball implements Playable {
19     String name;
20
21 ↓     public Volleyball(String name) {
22         this.name = name;
23    }
24    @Override
25 ↓     public void play() {
26         System.out.println(name + " is Playing volleyball");
27    }
28 }
```

```

26         System.out.println(name + " is Playing volleyball");
27     }
28 }
29 class Basketball implements Playable {
30     String name;
31
32     public Basketball(String name) {
33         this.name = name;
34     }
35
36     @Override
37     public void play() {
38         System.out.println(name + " is Playing basketball");
39     }
40 }
41 public class SportsGame {
42     public static void main(String[] args) {
43         Scanner scanner = new Scanner(System.in);
44         Playable[] players = new Playable[3];
45         String[] sports = {"Football", "Volleyball", "Basketball"};
46         for (int i = 0; i < players.length; i++) {
47             String name = scanner.nextLine();
48             switch (sports[i]) {
49                 case "Football":
50                     players[i] = new Football(name);
51                     break;
52                 case "Volleyball":

```

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

Passed all tests! ✓

**Question 3**

Correct

Marked out of 5.00

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 interface Sports {
3     public void setHomeTeam(String name);
4     public void setVisitingTeam(String name);
5 }
6 interface Football extends Sports {
7     public void homeTeamScored(int points);
8     public void visitingTeamScored(int points);
9 }
10
11
12 class College implements Football {
13     String homeTeam;
14     String visitingTeam;
15
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         System.out.println(homeTeam+" "+points+" scored");
24     }
25     public void visitingTeamScored(int points){
26         System.out.println(visitingTeam+" "+points+" scored");
27     }
28 }
```

```

28  public int winningteam(int p1, int p2){
29      if(p1>p2)
30          return 1;
31      else if(p1<p2)
32          return 0;
33      else
34          return -1;
35  }
36 }
37 public class Main{
38     public static void main(String[] args){
39         Scanner sc= new Scanner(System.in);
40         String hname=sc.next();
41         String vteam=sc.next();
42         int hpoints=sc.nextInt();
43         int vpoints=sc.nextInt();
44         College s= new College();
45         s.setHomeTeam(hname);
46         s.setVisitingTeam(vteam);
47         s.homeTeamScored(hpoints);
48         s.visitingTeamScored(vpoints);
49         if((s.winningTeam(htpoints,vpoints))==-1){
50             System.out.println(hname+" is the winner!");
51         }
52         else if((s.winningTeam(htpoints,vpoints))==-1){

```

	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 ►

**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 %)

[Reset answer](#)

```
1 class FinalExample {
2
3     // Final variable
4     final 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 class prog {
```

```
25 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 }
```

	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! ✓

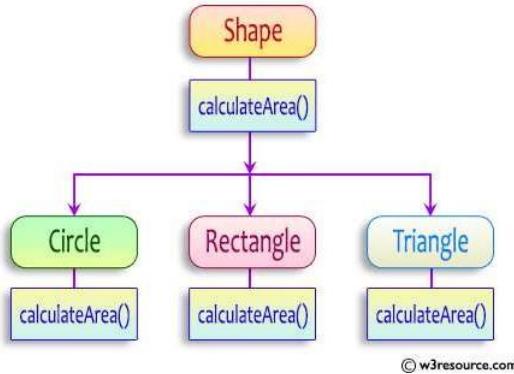
**Question 2**

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.*;
2 abstract class Shape{
3     abstract void calculatearea();
4 }
5 class Circle extends Shape{
```

```

6   float rad;
7   Circle(float rad){
8       this.rad = rad;
9   }
10  void calculatearea(){
11      System.out.format("Area of a circle: %.2f\n",3.14159*rad*rad);
12  }
13 }
14 class Rectangle extends Shape{
15     float l;
16     float br;
17     Rectangle(float l,float br){
18         this.l = l;
19         this.br = br;
20     }
21     void calculatearea(){
22         System.out.format("Area of a Rectangle: %.2f\n", (l*br));
23     }
24 }
25 }
26 class Triangle extends Shape{
27     float ba;
28     float h;
29     Triangle(float ba ,float h){
30         this.ba = ba;
31         this.h = h;
32     }
33     void calculatearea(){
34         System.out.format("Area of a Triangle: %.2f",0.5*ba*h);
35     }
36 }
37 }
38 class prog{
39     public static void main (String are[]){
40         Scanner scan = new Scanner(System.in);
41         float rad = scan.nextFloat();
42         float l = scan.nextFloat();
43         float br = scan.nextFloat();
44         float ba = scan.nextFloat();
45         float h = scan.nextFloat();
46         Circle c = new Circle(rad);
47         Rectangle r = new Rectangle(l,br);
48         Triangle t = new Triangle(ba,h);
49         c.calculatearea();
50         r.calculatearea();
51         t.calculatearea();
52     }

```

	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 andlast 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 andlast 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.*;
2 class prog{
3     public static void main(String ae[]){
4         Scanner scan = new Scanner(System.in);
5         int n = scan.nextInt();
6         String arr[] = new String[n];
7         scan.nextLine();
8         String str = scan.nextLine();
9         String temp = "";
10        int j=0;
11        int l=str.length();

```

```

12   for(int i = 0;i<1;i++){
13     if(str.charAt(i)==' '){
14       arr[j] = temp;
15       temp ="";
16       j++;
17     }
18   } else{
19     temp +=str.charAt(i);
20   }
21 }
22 arr[j] = temp;
23 String s = "";
24 char [] cha ={'a','A','e','E','i','I','o','O','u','U','u'};
25 for(int i=0;i<n;i++){
26   int c=0;
27   char [] ar = arr[i].toCharArray();
28   char ch1 = ar[0];
29   char ch2 = ar[ar.length -1];
30   for(char k : cha){
31     if(k==ch1){
32       c++;
33     }
34     if(k==ch2){
35       c++;
36     }
37   }
38   if(c==2){
39     s+=arr[i];
40   }
41 }
42 if(s==""){
43   System.out.print("no matches found");
44 }
45 else{
46   System.out.print(s.toLowerCase());
47 }
48
49
50
51
52

```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 ►

**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 Main
3 {
4     public static void main(String[] args)
5     {
6         Scanner sc=new Scanner(System.in);
7         int n=sc.nextInt();
8         int[] arr=new int[n];
9         for(int i=0;i<n;i++)
10        {
11            arr[i]=sc.nextInt();
12        }
13        try
14        {
15            int a=arr[0]/arr[1];
16            int b=arr[n];
17        }
18        catch(ArithmaticException ae)
19        {
20            System.out.println(ae.toString());
21        }
22        catch(ArrayIndexOutOfBoundsException e)
23        {
24            System.out.println(e.toString());
25        }
26        finally
27        {
28            System.out.println("I am always executed");
29        }
}

```

```
30  
31 }  
32 }
```

	<b>Test</b>	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 v class prog {  
2 v   public static void main(String[] args) {  
3 v     int n = 82;  
4 v     trynumber(n);  
5 v     n = 37;  
6 v     // call the trynumber(n);  
7 v     trynumber(n);  
8 v   }  
9 v  
10 v }  
11 v  
12 v   public static void trynumber(int n) {  
13 v     try {  
14 v       //call the checkEvenNumber()  
15 v       checkEvenNumber(n);  
16 v       System.out.println(n + " is even.");  
17 v     } catch (Exception e) {  
18 v       System.out.println("Error: " + e.getMessage());  
19 v     }  
20 v   }  
21 v  
22 v   public static void checkEvenNumber(int number) throws Exception {  
23 v     if (number % 2 != 0) {  
24 v       throw new Exception(number+" is odd.");  
25 v     }  
26 v   }  
27 v }  
28 }
```

	<b>Expected</b>	<b>Got</b>	
✓	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 class prog {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         int length = sc.nextInt();
7         // create an array to save user input
8         int[] name = new int[length];
9         int sum=0;//save the total sum of the array.
10
11
12     /* Define try-catch block to save user input in the array "name"
13     If there is an exception then catch the exception otherwise print
14     the total sum of the array. */
15     try
16     {
17         for(int i=0;i<length;i++)
18         {
19             name[i]=sc.nextInt();
20             sum+=name[i];
21         }
22         System.out.println(sum);
23     }
24
25     catch(InputMismatchException ae )
26 }
```

```
27  {
28      System.out.println("You entered bad data.");
29  }
30
31
32
33
34
35
36
37  }
38 }
```

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 ►

**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.*;
2 public class Solution
3 {
4     public static void main(String[] args)
5     {
6         Scanner sc=new Scanner(System.in);
7         int n=sc.nextInt();
8         ArrayList<Integer> arr=new ArrayList<Integer>(n);
9         while(sc.hasNextInt())
10        {
11             int a=sc.nextInt();
12             arr.add(a);
13         }
14
15         int b=arr.size()-1;
16         System.out.println("ArrayList: "+arr);
17         System.out.println("First : "+arr.get(0)+" , Last : "+arr.get(b));
18     }
19 }
```

	<b>Test</b>	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 public 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        // printing initial value ArrayList
17        System.out.println("ArrayList: " + list);
18
19        //Replacing the element at index 1 with 100
20        list.set(1,100);
21
22
23        //Getting the index of first occurrence of 100
24        System.out.println("Index of 100 = "+list.indexOf(100));
25
26        //Getting the index of last occurrence of 100
27        System.out.println("LastIndex of 100 = "+ list.lastIndexOf(100));
28        // Check whether 200 is in the list or not
29        System.out.println(list.contains(200));
30        // Print ArrayList size
31        System.out.println("Size Of ArrayList = "+ list.size());
32        //Inserting 500 at index 1
33        list.add(1,500);
34        //Removing an element from position 3
35        list.remove(3);
36        System.out.print("ArrayList: " + list);
37    }
38 }
```

	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]	✓

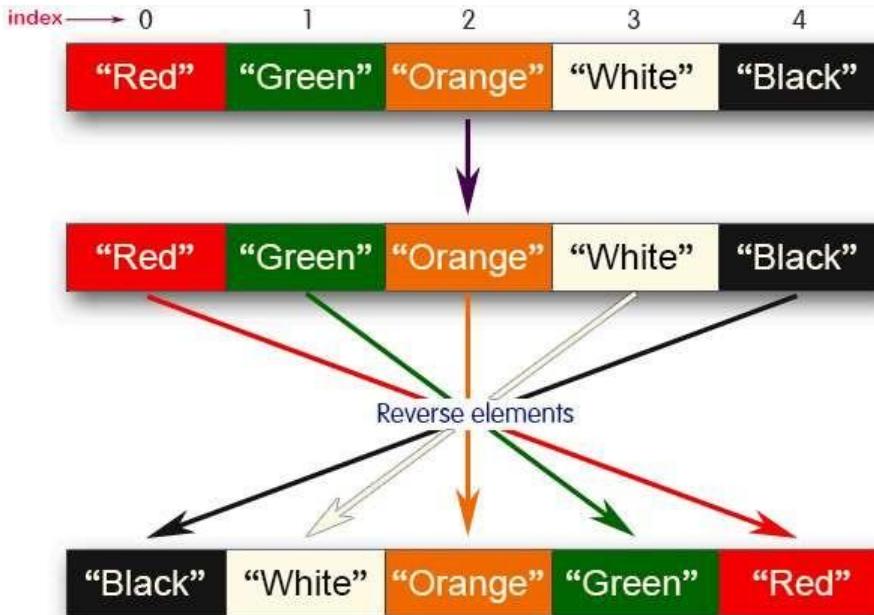
Passed all tests! ✓

**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.*;
2 public class Reverse
3 {
4     public static void main(String[] args)
5     {
6         Scanner sc=new Scanner(System.in);
7         int n=sc.nextInt();
8         ArrayList<String> list=new ArrayList<String>(n);
9         for(int i=0;i<n;i++)
10        {
11            String str=sc.next();
12            list.add(str);
13        }
14        ArrayList<String> list1=new ArrayList<String>(n);
15        for(int i=n-1;i>=0;i--)
16        {
17            String str1=list.get(i);
18            list1.add(str1);
19        }
20        System.out.println("List before reversing :");
21        System.out.println(list);
22        System.out.println("List after reversing :");
23        System.out.println(list1);
24    }
25 }
```

	<b>Test</b>	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	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 ►

**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 class prog
4 {
5     public static void main(String[] args)
6     {
7         Scanner sc= new Scanner(System.in);
8         int n = sc.nextInt();
9         // Create a HashSet object called numbers
10        HashSet<Integer> numbers= new HashSet<Integer>(n);
11        // Add values to the set
12        for(int i=0;i<n;i++)
13        {
14            numbers.add(sc.nextInt());
15        }
16        int skey=sc.nextInt();
17        // Show which numbers between 1 and 10 are in the set
18        if(numbers.contains(skey))
19        {
20            System.out.println(skey+ " was found in the set.");
21        }
22    }
23 }
```

```
21
22     }
23     else
24     {
25         System.out.println(skey + " was not found in the set.");
26     }
27 }
28 }
```

	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 import java.util.ArrayList;
4 public class Main
5 {
6     public static void main(String[] args)
7     {
8         Scanner scanner = new Scanner(System.in);
9         HashSet<String> set1 = new HashSet<>();
10        HashSet<String> set2 = new HashSet<>();
11        int numElementsSet1 = scanner.nextInt();
12        scanner.nextLine();
13        for (int i = 0; i < numElementsSet1; i++) {
14            String num = scanner.nextLine();
15            set1.add(num);
16        }
17        int numElementsSet2 = scanner.nextInt();
18        scanner.nextLine();
19        for (int i = 0; i < numElementsSet2; i++)
20        {
21            String num = scanner.nextLine();
22            set2.add(num);
23        }
24        set1.retainAll(set2);
25        ArrayList<String> list = new ArrayList<>(set1);
26        for (int i = 0; i < list.size(); i++)
27        {
28            System.out.println(list.get(i));
29        }
30    }
31}
```

```
29 }  
30 }  
31 }  
32 }
```

	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 class prog
6 {
7     public static void main(String[] args)
8     {
9         //Creating HashMap with default initial capacity and load factor
10        HashMap<String, Integer> map = new HashMap<String, Integer>();
11
12        String name;
13        int num;
14        Scanner sc= new Scanner(System.in);
15        int n=sc.nextInt();
16        for(int i =0;i<n;i++)
17        {
18            name=sc.next();
19            num= sc.nextInt();
20            map.put(name,num);
21        }
22
23        //Printing key-value pairs
24
25
26        Set<Entry<String, Integer>> entrySet = map.entrySet();
27
28        for (Entry<String, Integer> entry : entrySet)
29        {
30            System.out.println(entry.getKey()+" : "+entry.getValue());
31        }
32        System.out.println("-----");
33        //Creating another HashMap
34
35        HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();
36
37        //Inserting key-value pairs to anotherMap using put() method
38
39        anotherMap.put("SIX", 6);
40
41        anotherMap.put("SEVEN", 7);
42
43        //Inserting key-value pairs of map to anotherMap using putAll() method
44
45        anotherMap.putAll      ( map    ); // code here
46
47        //Printing key-value pairs of anotherMap
48
49        entrySet = anotherMap.entrySet();

```

```
50
51     for (Entry<String, Integer> entry : entrySet)
52 {
```

	<b>Test</b>	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	1	3 ONE 1 TWO 2 THREE 3	ONE : 1 TWO : 2 THREE : 3 <hr/> SIX : 6 ONE : 1 TWO : 2 SEVEN : 7 THREE : 3 2 true true 4	ONE : 1 TWO : 2 THREE : 3 <hr/> SIX : 6 ONE : 1 TWO : 2 SEVEN : 7 THREE : 3 2 true true 4	✓

Passed all tests! ✓

◀ Lab-11-MCQ

Jump to...

TreeSet example ►

### Question 1

Correct

Marked out of 5.00

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

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

Each alphabet is represented by a sequence of 0s.

This is as mentioned below:

Z: 0

Y:00

x · 000

W : 0000

V:00000

U · 000000

T : 0000000

and so on upto A having 26 0's (000000000000000000000000000000).

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:

The decoded string (original word) will be: WIPRO

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

**For example:**

Input	Result
010010001	ZYX
000010000000000000000000000000001000000000000100000000001000000000010000000000000001	WIPRO

**Answer:** (penalty regime: 0 %)

```
1 import java.util.Scanner;
2 public class BinaryStringDecoder
3 {
4     public static void main(String[] args)
5     {
6         Scanner scanner = new Scanner(System.in);
7         String encodedString = scanner.nextLine();
8         String[] parts = encodedString.split("1");
9         StringBuilder decodedString = new StringBuilder();
10        for (String part : parts) {
11            int zeroCount = part.length();
12            if (zeroCount >= 1 && zeroCount <= 26) {
13                char decodedChar = (char) ('Z' - zeroCount + 1);
14                decodedString.append(decodedChar);
15            }
16        }
17        System.out.println(decodedString.toString());
18    }
19 }
```

20 |

	<b>Input</b>	<b>Expected</b>	<b>Got</b>	
✓	010010001	ZYX	ZYX	✓
✓	000010000000000000000000100000000000100000000001000000000001	WIPRO	WIPRO	✓

Passed all tests! ✓



**Question 2**

Correct

Marked out of 5.00

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

Get the ASCII values of all the extracted alphabets.

Calculate sum of those ASCII values. Lets call it `sum1` and calculate single digit sum of `sum1`, i.e., keep adding the digits of `sum1` until you arrive 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.Scanner;
2 public class ArrayCommonElementSum {
3     public static void main(String[] args) {
4         Scanner scanner = new Scanner(System.in);
5         String[] array1 = scanner.nextLine().split(" ");
6         String[] array2 = scanner.nextLine().split(" ");
7         int asciiSum = 0;
8         for (String element : array2) {
9             for (String elem1 : array1) {
10                 if (element.equals(elem1)) {
11                     asciiSum += element.charAt(0);
12                     break;
13                 }
14             }
15         }
16         int singleDigitSum = asciiSum % 9;
17         if (singleDigitSum == 0 && asciiSum > 0) {
18             singleDigitSum = 9;
19         }
20         System.out.println(singleDigitSum);
21     }
22 }
```

zz | j

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

Passed all tests! ✓

/

**Question 3**

Correct

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 SeigOlhoncet 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:**

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

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

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

Examples:

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

**For example:**

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

**Answer:** (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class WordReversal
3 {
4     public static String reverseWords(String sentence, int caseOption) {
5         String[] words = sentence.split(" ");
6         StringBuilder modifiedSentence = new StringBuilder();
7         for (String word : words) {
8             String reversedWord = reverseWord(word, caseOption);
9             if (modifiedSentence.length() > 0) {
10                 modifiedSentence.append(" ");
11             }
12             modifiedSentence.append(reversedWord);
13         }
14         return modifiedSentence.toString();
15     }
16     private static String reverseWord(String word, int caseOption) {
17         char[] chars = word.toCharArray();
18         int left = 0;
19         int right = chars.length - 1;
20         while (left < right) {
21             char temp = chars[left];
22             chars[left] = chars[right];
23             chars[right] = temp;
24             left++;
25             right--;
26         }
27         if (caseOption == 0) {
28             return new String(chars);
29         } else {
30             return new String(chars).toLowerCase();
31         }
32     }
33 }
```

```

10         modifiedSentence.append(" ");
11         modifiedSentence.append(reversedWord);
12     }
13
14     return modifiedSentence.toString();
15 }
16
17 private static String reverseWord(String word, int caseOption) {
18     String reversedWord = new StringBuilder(word).reverse().toString();
19     if (caseOption == 0) {
20         return reversedWord;
21     } else if (caseOption == 1) {
22         StringBuilder casePreservedWord = new StringBuilder(reversedWord);
23
24         for (int i = 0; i < word.length(); i++) {
25             char originalChar = word.charAt(i);
26             if (Character.isUpperCase(originalChar)) {
27                 casePreservedWord.setCharAt(i, Character.toUpperCase(casePreservedWord.charAt(i)));
28             } else if (Character.isLowerCase(originalChar)) {
29                 casePreservedWord.setCharAt(i, Character.toLowerCase(casePreservedWord.charAt(i)));
30             }
31         }
32         return casePreservedWord.toString();
33     }
34     return word;
35 }
36
37 public static void main(String[] args) {
38     Scanner scanner = new Scanner(System.in);
39     String sentence = scanner.nextLine();
40     int caseOption = scanner.nextInt();
41     String result = reverseWords(sentence, caseOption);
42     System.out.println(result);
43 }
44 }
```

0

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

Passed all tests! ✓



◀ Lab-12-MCQ

Jump to...

Identify possible words ►

# **EMPLOYEE MANAGEMENT SYSTEM**

CS23333 – Object Oriented Programming using JAVA

Mini Project Report Submitted by

SUVATHI D - 231001225

SRI SANJANA B - 231001214

SHREENITHI A - 231001196

*of*

BACHELOR OF TECHNOLOGY

*in*

INFORMATION TECHNOLOGY

RAJALAKSHMI ENGINEERING COLLEGE,

THANDALAM

(An Autonomous Institution)



**2024**

## **RAJALAKSHMI ENGINEERING COLLEGE**

### **BONA FIDE CERTIFICATE**

Certified that this project report titled “**EMPLOYEE MANAGEMENT SYSTEM**” is the bona fide work of “**SUVATHI D(231001225), SRI SANJANA B(231001214),SHREENITHI A(231001196)**” who Carried out the project work under my supervision.

Submitted for the Practical Examination held on \_\_\_\_\_

#### **HEAD/IT**

##### **Dr.P.Valarmathie**

Professor and Head,  
Information Technology,  
Rajalakshmi Engineering College  
Thandalam, Chennai – 602 105

#### **SUPERVISOR**

##### **Mr.K E Narayana,**

Assistant Professor,  
Information Technology  
Rajalakshmi Engineering College  
Thandalam, Chennai – 602 105

## **ABSTRACT**

The main objective of this project is to build an Employee management system that will store and retrieve records of Employees. The System is intended to accept process, generate Employee details. The System was developed using basic technologies such as MYSQL database and Java (JDBC, SWING, AWT).The system is free of errors and very efficient and less time consuming due to care taken to develop it. All the phases of Development cycle are employed and it is worthwhile to state that the system is user friendly and strong

---

# TABLE OF CONTENTS

<b>1</b>	<b>INTRODUCTION</b>	<b>EMPLOYEE MANAGEMENT SYSTEM</b>	<b>6</b>
	<b>1.1 Purpose</b>		
	<b>1.2 Scope of Project</b>		
	<b>1.3 Software Requirement Specification</b>		
<b>2</b>	<b>SYSTEM FLOW DIAGRAMS</b>		<b>13</b>
	<b>2.1 Use Case Diagram</b>		
	<b>2.2 Entity-relationship Diagrams</b>		
	<b>2.3 Data Flow Diagram</b>		
<b>3</b>	<b>MODULE DESCRIPTION</b>		<b>15</b>
	<b>3.2 Add Employee</b>		
	<b>3.3 View Employee</b>		
	<b>3.4 Delete Employee</b>		
	<b>3.5 Add Salary</b>		
	<b>3.6 Update Salary</b>		
<b>4</b>	<b>IMPLEMENTATION</b>		<b>16</b>
	<b>4.1 Design</b>		
	<b>4.2 Database Design</b>		
	<b>4.3 Code</b>		
<b>5</b>	<b>CONCLUSION</b>		<b>32</b>
<b>6</b>	<b>REFERENCE</b>		<b>32</b>

# **1 INTRODUCTION:**

Employee Management System is a Software which is helpful for the Authorities. In the current system all the activities are done manually. It is very time consuming and costly. Our Employee Management System deals with the various activities related to the Employees.

By using this software, we can get Name and Salary details of the Employees in a single window page

## **1.1 PURPOSE:**

The objective of *Employee management system* is to allow the administrator of any organization to edit and find out the personal details of a Employee and allows them to keep up to date their profile. Overall, it will make Employee information management an easier job for the administration and the Employee of any organization.

## **1.2 SCOPE OF THE PROJECT:**

The proposed system will work with the administrator. The system works and fulfils all the functionalities as per the proposed system. It will provide reduced response time against the queries made by different users. This project is based on Java (JDBC) language with MYSQL database which manage the details of the Employee because it is a tedious job for any organization. Employee Information system will store all the details of the Employees including their background information.

## **1.3 Software requirement Specification:**

### **Introduction:**

The Employee management system can handle all the details about a Employee. The details include Employee personal details, Salary details etc., the Employee management system is an automated version of manual Employee management system.

---

## **Document Purpose:**

This SRS Document contains the complete software requirements for the Employee Management System and describes the design decisions, architectural design and the detailed design needed to implement the system. It provides the visibility in the design and provides information needed for software support.

## **Product Scope:**

Employee Management System is developing for general purpose and used to replace old paper work system. SMS is to build to efficiently provide Employee information to school/college administration. It provides a mechanism to edit the Employee information form which makes the system flexible.

**Definitions,                  Acronyms                  and**

**Abbreviations:** EMS - Online Employee Information  
Management        System SRS    -   Software   Requirements  
Specification

## **References and Acknowledgement:**

- [1] <https://www.javatpoint.com/java.awt>
- [2] <https://www.javatpoint.com/java-swing>

## **Overall Description:**

The Employee management system allows authorized members to access the records of academically registered Employees. It can be used in various educational institutes across the globe and simplifies working of institutes.

## **Product Perspective:**

The proposed system shall be developed using client/server architecture and be compatible with Microsoft Windows Operating System. The front end of the system will be developed using Java AWT and SWING and backend will be developed using My SQL server.

---

## **Product Functionality:**

- a) Add Employee: to store a new Employee.
- b) View Employee: to view and update an existing Employee
- c) Delete Employee: to delete an existing Employee.
- d) Add Salary: to store a particular Employee Salary.
- e) Update Salary: to view and update an existing Employee Salary.

## **User and Characteristics:**

Qualification: At least matriculation and comfortable with English.

Experience: Should be well versed/informed about the registration process of the university.

Technical Experience: Elementary knowledge of computers.

## **Operating Environment:**

Hardware requirements:

- Any Processor Over i3.
- Any version of Windows 8,10,11.
- Processor speed: 2.0 GHz
- RAM: 4GB
- Hard disk: 500GB Software requirements:
- Database: MySQL
- Frontend: JAVA (SWING, AWT)
- Technology: JAVA (JDBC)

## **Constraints:**

It will only accessed by the administrator.

The delete operation is available only to the administrator.

To reduce the complexity of the system, there is no check on delete operation.

Hence, administrator should be very careful before deletion of any record and he/she will be responsible for data consistency.

---

## Specific Requirements:

### User Interface:

- The EMS will have following user-friendly single page interface.
- a) Add Employee: to store a new Employee.
  - b) View Employee: to view and update a existing Employee
  - c) Delete Employee: to delete a existing Employee.
  - d) Add Salary: to store a particular Employee Salary.
  - e) Update Salary: to view and update a existing Employee Salary

### Hardware Interface:

- a) Screen resolution of at least 640 x 480 or above.
- b) **Any version of Windows 8,10,11.**

### Software Interface:

- a) MS-Windows Operating System
- b) Java AWT and SWING for designing front-end
- c) MS SQL Server for backend
- d) PLATFORM : JAVA LANGUAGE
- e) INTEGRATED DEVELOPMENT ENVIRONMENT(IDE): VS CODE

### Functional Requirements:

#### Log in Module (LM):

User (admin) shall be able to load the Login Module. The LM shall support the user to log into the system. The login panel shall contain fields to contain a user name and a field for password. The password field shall be masked with symbols when the user types. It shall also contain a button labeled as Login. When the user clicks on Login button the username and password will be verified by database administrator and then only the user will able to use the system functions.

#### Registered Users Module (RUM):

---

After successful login, user shall be able to continue navigating through the Application and view Employee detailed information. After successful login, user (admin) shall be able to update and maintain their Employee profile, such as changing Salary and personal details.

### **Administrator Module (AM):**

After successful login, system shall display administrative functions. Administrative functions shown shall be add and update. When administrator clicks on the add button, system shall display a section where administrator can add new Employee details, remove unused Employee details and many more. When administrator clicks on update button, system shall display a section where administrator can update Employee details which are currently stored in the database. When administrator adds, updates or delete and entry, the AM module will send the request to the Server Module which will do the necessary changes to the DB.

### **Server Module (SM):**

SM shall be between the various modules and the DB. SM shall receive all requests and format the pages accordingly to be displayed. SM shall validate and execute all requests from the other modules

## **Non-functional Requirements:**

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, etc.).

### **Performance:**

The system and the server must be capable of handling the real-time error functionality occurs by the defined users. In addition, the system must be safety critical. All failures reported by the server side must be handled instantaneously to allow for user and system safety.

---

## **Reliability:**

The system is safety critical. If it moves out of normal operation mode, the requirement to drop or down the server and fix it as soon as possible and open it again. This emergency behaviour shall not occur without reason.

## **Availability:**

When in normal operating conditions, request by a user for system shall be handled within 1 second. Immediate feedback of the systems activities shall be communicated to the user. **Security:**

There shall be a strong security mechanism should be place in the server side of the system to keep unwanted users to hack or damage the system. However, all users of the system give and store the details of privacy related to personal information and many other.

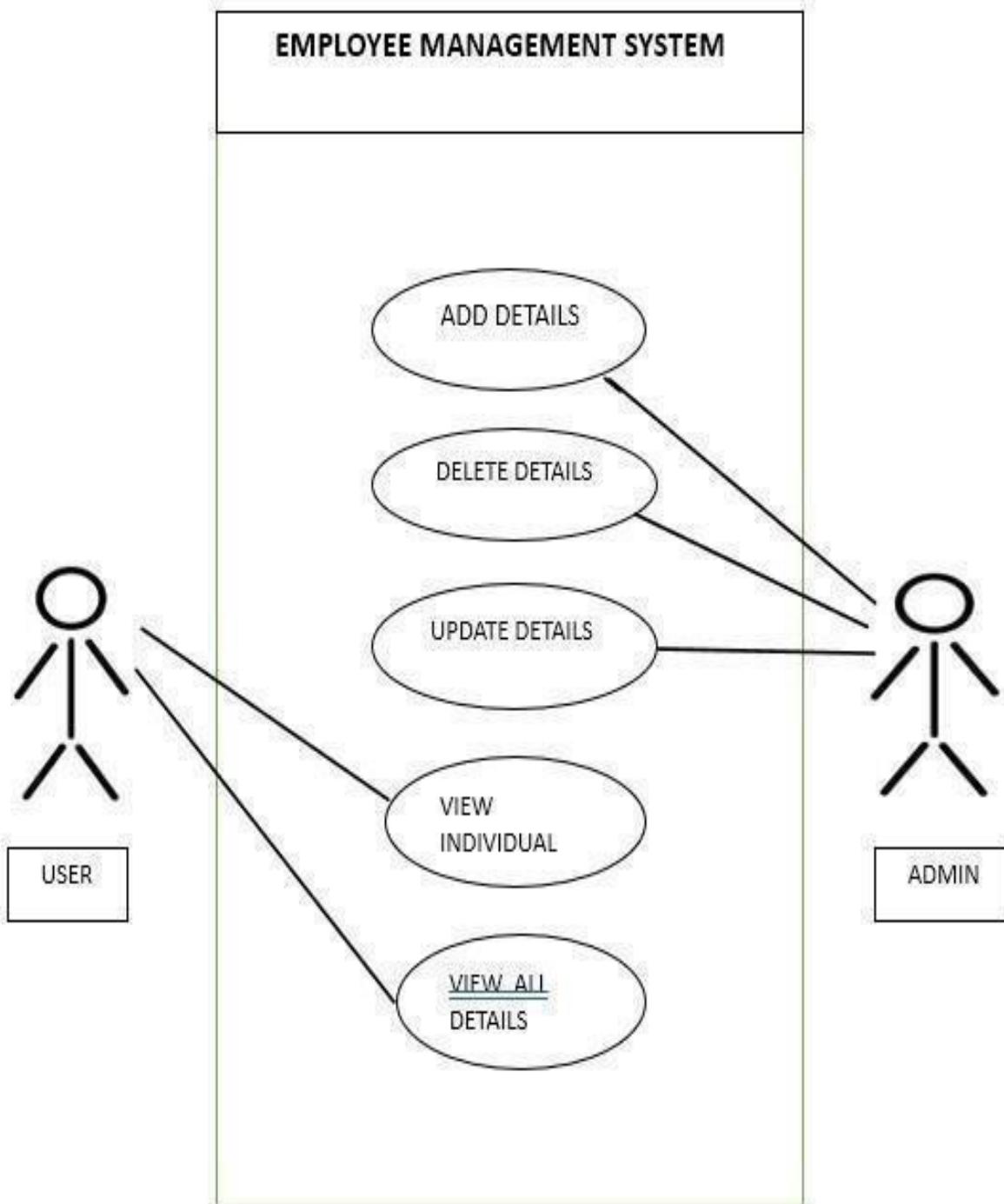
## **Maintainability:**

There shall be design documents describing maintenance of the software and database used to save the user details as well as the daily updated and modification done in system. There shall be an access on the control system by the admin to be maintained it properly at the front end as well as at back end.

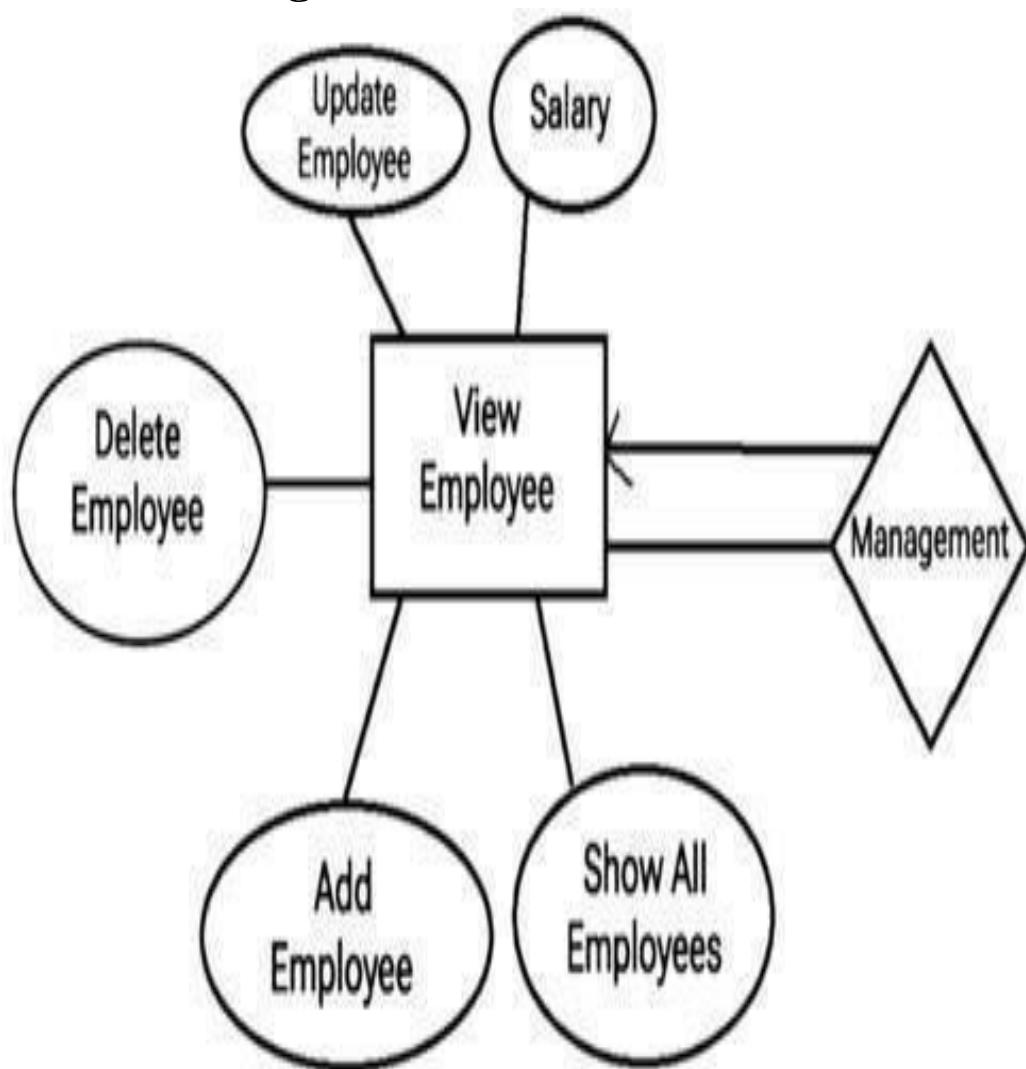
---

## SYSTEM FLOW DIAGRAMS:

### 2.1 Use case Diagram:



## 2.2 ER Model Diagram:



## 2.3 Data Flow Diagram:



Employee id	Name	Salary
1	Mani	10000
2	Kamal	9999
3	Kailash	10000

## 3. MODULE DESCRIPTION:

The different types of modules present in this project are:

### Details:

#### Employee id, Name, Salary

**3.1 Add Employee:** In this section admin can add Employee details

**3.2 View Employee:** In this section admin can view and update Employee details.

**3.3 Delete Employee:** In this section admin can delete Employees details.

**3.4 Update Employee:** In this section admin can update Employee Salary

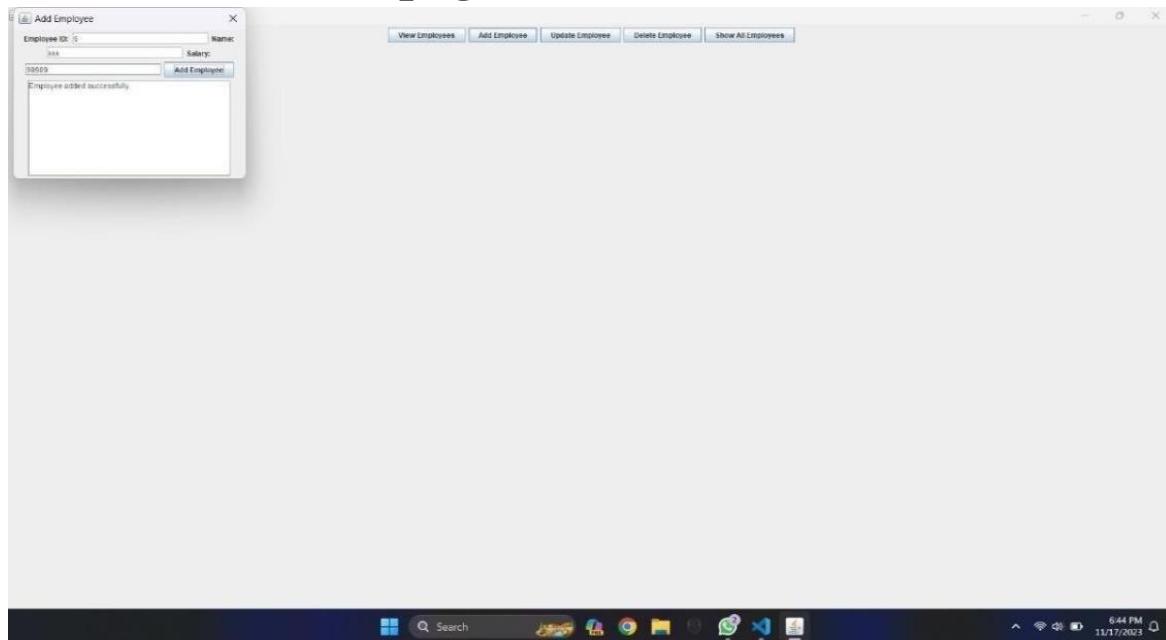
**3.5 View Employee:** In this section user can view details of individual Employee

**3.6 Show all:** In this section user view details of all the employees.

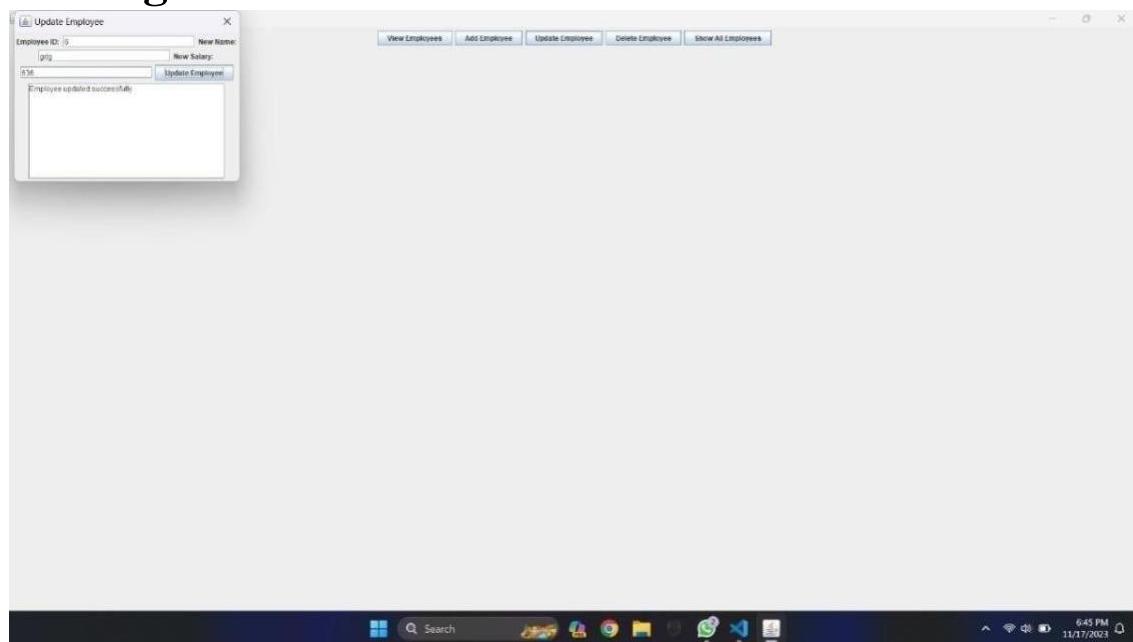
## 4 IMPLEMENTATION :

### 4.1 DESIGN :

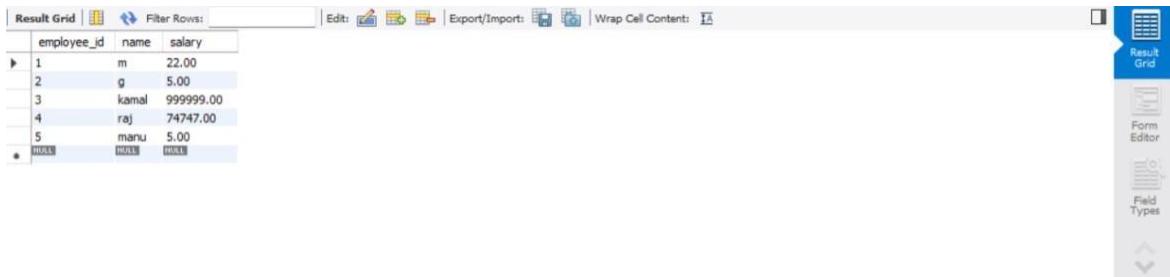
#### SCREENS: Homepage



#### Adding Details



## 4.2 DATABASE DESIGN:



A screenshot of a database management system interface showing a result grid. The grid displays five rows of data with columns for employee\_id, name, and salary. The interface includes various toolbar icons and a sidebar on the right.

employee_id	name	salary
1	m	22.00
2	g	5.00
3	kamal	999999.00
4	raj	74747.00
5	manu	5.00
*	HULL	HULL

### **4.3 CODE:**

```
import javax.swing.*; import
java.awt.*; import
java.awt.event.ActionEvent; import
java.awt.event.ActionListener; import
java.sql.Connection; import
java.sql.DriverManager; import
java.sql.PreparedStatement; import
java.sql.ResultSet; import
java.sql.SQLException;

public class EmployeeManagementApp extends JFrame implements
ActionListener { private static final String
JDBC_URL =
"jdbc:mysql://localhost:3306/employee_db";
private static final String USERNAME = "root";
private static final String PASSWORD = "root";

private JButton viewButton, addButton, updateButton, deleteButton,
allEmpButton;

public EmployeeManagementApp() {
    setTitle("Employee Management");
    setSize(400, 200);
    setDefaultCloseOperation(EXIT_ON_CLOSE)
    ; setLayout(new FlowLayout());

    viewButton = new JButton("View Employees"); addButton
```

---

```
= new JButton("Add Employee"); updateButton = new
JButton("Update Employee"); deleteButton = new
JButton("Delete Employee"); allEmpButton = new JButton("Show
All Employees");

viewButton.addActionListener(this); addButton.addActionListener(this);
updateButton.addActionListener(this);
deleteButton.addActionListener(this);
allEmpButton.addActionListener(this); add(viewButton); add(addButton);
add(updateButton); add(deleteButton); add(allEmpButton);

}

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == viewButton) {
        openViewEmployeesWindow();
    } else if (e.getSource() == addButton) { openAddEmployeeWindow();
    } else if (e.getSource() == updateButton) { openUpdateEmployeeWindow();
    } else if (e.getSource() == deleteButton) { openDeleteEmployeeWindow();
    } else if (e.getSource() == allEmpButton) {
        openShowAllEmployeesWindow();
    }
}

private void openViewEmployeesWindow() {
```

---

```

JDialog viewDialog = new JDialog(this, "View Employees", true);
viewDialog.setSize(600,      400);      viewDialog.setLayout(new
FlowLayout());

JComboBox<String>  searchCriteriaComboBox  =  new
JComboBox<>(new String[]{"Employee ID", "Name", "Salary"});

JTextField searchText = new JTextField(20);

JButton searchButton = new JButton("Search"); JTextArea
resultTextArea      =      new      JTextArea(10,      50);
resultTextArea.setEditable(false);

searchButton.addActionListener(new ActionListener() {
    @Override public void actionPerformed(ActionEvent
e) {
        try          (Connection          connection          =
DriverManager.getConnection(JDBC_URL, USERNAME, PASSWORD)) {
            String criteria = (String)
searchCriteriaComboBox.getSelectedItem();

            String searchTerm = searchText.getText();

            String query; if
(criteria.equals("Employee ID")) {    query = "SELECT * FROM employees
WHERE employee_id =
?";

        } else if (criteria.equals("Name")) {
            query = "SELECT * FROM employees WHERE name = ?";
        } else if (criteria.equals("Salary")) { try
{

```

---

```

        query = "SELECT * FROM employees WHERE salary = ?";

    } catch (NumberFormatException ex) {
        resultTextArea.setText("Invalid salary value. Please enter a
valid numeric value.");
        return;
    }

} else { resultTextArea.setText("Invalid search criteria.");
return; }

try (PreparedStatement statement =
connection.prepareStatement(query)) {

if (criteria.equals("Employee ID") || criteria.equals("Name")) {
    statement.setString(1, searchTerm);
}

try (ResultSet resultSet = statement.executeQuery())
{
    resultTextArea.setText(""); // Clear the text area
    while (resultSet.next()) {
        int id =
        resultSet.getInt("employee_id"); String name =
        resultSet.getString("name"); double salary =
        resultSet.getDouble("salary");

        resultTextArea.append("Employee ID: " + id + ", Name: " +
name + ", Salary: " + salary + "\n");
    }
}
} catch (SQLException ex) { ex.printStackTrace();

```

---

```
        }

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

}

});

});
```

```
viewDialog.add(searchCriteriaComboBox);

viewDialog.add(searchText);

viewDialog.add(searchButton);           viewDialog.add(new

JScrollPane(resultTextArea)); viewDialog.setVisible(true);

}
```

```
private void openAddEmployeeWindow() {

JDialog addDialog = new JDialog(this, "Add Employee", true);

addDialog.setSize(400,      300); addDialog.setLayout(new

FlowLayout());

JTextField idField = new JTextField(20); // Add employee ID field

JTextField nameField = new JTextField(20);

JTextField salaryField = new JTextField(20);

JButton addButton = new JButton("Add Employee");

JTextArea      resultTextArea      =      new      JTextArea(10,      30);

resultTextArea.setEditable(false);

addButton.addActionListener(new ActionListener() {
```

---

```
@Override           public           void
actionPerformed(ActionEvent e) {
    try (Connection connection =
DriverManager.getConnection(JDBC_URL, USERNAME, PASSWORD)) {
        int id = Integer.parseInt(idField.getText()); // Get employee ID
from the input field

        String name = nameField.getText();
        double salary = Double.parseDouble(salaryField.getText());

        String insertQuery = "INSERT INTO employees (employee_id,
name, salary) VALUES (?, ?, ?)";

        try (PreparedStatement insertStatement =
connection.prepareStatement(insertQuery)) {
            insertStatement.setInt(1, id); // Set the employee ID in the query
            insertStatement.setString(2, name);
            insertStatement.setDouble(3, salary); int rowsAffected =
            insertStatement.executeUpdate(); if (rowsAffected > 0) {
                resultTextArea.setText("Employee added successfully.");
            } else { resultTextArea.setText("Failed to add the
employee."); }
        } catch (SQLException ex) { ex.printStackTrace();
        }
    } catch (SQLException ex) { ex.printStackTrace();
    }
});
```

```
addDialog.add(new JLabel("Employee ID: ")); // Label for employee ID
addDialog.add(idField); // Input field for employee ID
addDialog.add(new JLabel("Name: ")); addDialog.add(nameField);
addDialog.add(new JLabel("Salary: ")); addDialog.add(salaryField);
addDialog.add(addButton);
addDialog.add(new JScrollPane(resultTextArea));

addDialog.setVisible(true);
}

private void openUpdateEmployeeWindow() {
    JDialog updateDialog = new JDialog(this, "Update Employee", true);
    updateDialog.setSize(400, 300); updateDialog.setLayout(new
    FlowLayout());

    JTextField idField = new JTextField(20);
    JTextField nameField = new JTextField(20);
    JTextField salaryField = new JTextField(20);
    JButton updateButton = new JButton("Update Employee"); JTextArea
    resultTextArea = new JTextArea(10, 30);
    resultTextArea.setEditable(false);

    updateButton.addActionListener(new ActionListener() {
        @Override public void
        actionPerformed(ActionEvent e) { try
        (Connection connection =
DriverManager.getConnection(JDBC_URL, USERNAME, PASSWORD)) {
```

```
int id = Integer.parseInt(idField.getText()); String newName
= nameField.getText();
double newSalary = Double.parseDouble(salaryField.getText());

String updateQuery = "UPDATE employees SET name = ?, salary
= ? WHERE employee_id = ?";
try (PreparedStatement updateStatement =
connection.prepareStatement(updateQuery))
{
    updateStatement.setString(1, newName);
    updateStatement.setDouble(2, newSalary);
    updateStatement.setInt(3, id);
    int rowsAffected = updateStatement.executeUpdate(); if
(rowsAffected > 0) { resultTextArea.setText("Employee
updated successfully.");
} else { resultTextArea.setText("Failed to update the employee.");
}
} catch (SQLException ex) { ex.printStackTrace();
}
} catch (SQLException ex) { ex.printStackTrace();
}
});
}

updateDialog.add(new JLabel("Employee ID: "));
updateDialog.add(idField); updateDialog.add(new
JLabel("New Name: "));
updateDialog.add(nameField);
```

---

```

updateDialog.add(new JLabel("New Salary: "));
updateDialog.add(salaryField);
updateDialog.add(updateButton);
updateDialog.add(new JScrollPane(resultTextArea));
updateDialog.setVisible(true);

}

private void openDeleteEmployeeWindow() {
    JDialog deleteDialog = new JDialog(this, "Delete Employee", true);
    deleteDialog.setSize(400, 150); deleteDialog.setLayout(new
    FlowLayout());

    JTextField idField = new JTextField(20);
    JButton deleteButton = new JButton("Delete Employee"); JTextArea
    resultTextArea = new JTextArea(4, 30);
    resultTextArea.setEditable(false);

    deleteButton.addActionListener(new ActionListener() {
        @Override public void
        actionPerformed(ActionEvent e) {
            try (Connection connection =
DriverManager.getConnection(JDBC_URL, USERNAME, PASSWORD)) {
                int id = Integer.parseInt(idField.getText());
                String deleteQuery = "DELETE FROM employees WHERE
employee_id = ?";
                try (PreparedStatement deleteStatement =

```

---

```
connection.prepareStatement(deleteQuery)) {  
    deleteStatement.setInt(1, id); int rowsAffected =  
    deleteStatement.executeUpdate(); if (rowsAffected >  
    0) {  
        resultTextArea.setText("Employee deleted successfully."); }  
        else { resultTextArea.setText("Failed to delete the employee."); }  
    }  
} catch (SQLException ex) { ex.printStackTrace(); }  
} catch (SQLException ex) { ex.printStackTrace(); }  
});  
  
deleteDialog.add(new JLabel("Employee ID: "));  
deleteDialog.add(idField);  
deleteDialog.add(deleteButton); deleteDialog.add(new  
JScrollPane(resultTextArea));  
  
deleteDialog.setVisible(true);  
}  
  
private void openShowAllEmployeesWindow() {
```

---

```
JDialog allEmpDialog = new JDialog(this, "All Employees", true);
allEmpDialog.setSize(600, 400); allEmpDialog.setLayout(new
FlowLayout());
JTextArea resultTextArea = new JTextArea(10, 50);
resultTextArea.setEditable(false);

try (Connection connection = DriverManager.getConnection(JDBC_URL,
USERNAME, PASSWORD)) {
    String query = "SELECT * FROM employees";
    try (PreparedStatement statement =
connection.prepareStatement(query)) {
        try (ResultSet resultSet = statement.executeQuery()) {
            resultTextArea.setText(""); // Clear the text area while
            (resultSet.next()) { int id =
                resultSet.getInt("employee_id"); String name
                = resultSet.getString("name"); double salary =
                resultSet.getDouble("salary");
                resultTextArea.append("Employee ID: " + id + ", Name: " +
name + ", Salary: " + salary + "\n");
            }
        }
    } catch (SQLException ex) { ex.printStackTrace();
    }
} catch (SQLException ex) { ex.printStackTrace();
}
```

```

        allEmpDialog.add(new JScrollPane(resultTextArea));

    }

    public static void main(String[] args)
    { SwingUtilities.invokeLater(() -> {
        EmployeeManagementApp app = new EmployeeManagementApp();
        app.setVisible(true);
    });
}

```

## TABLE CREATION:

```

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

public class CreateTable {

    public static void main(String[] args) {
        // Database connection parameters
        String JDBC_URL = "jdbc:mysql://localhost:3306/employee_db";
        String USERNAME = "root";

```

---

```
String PASSWORD = "root";

try (Connection connection = DriverManager.getConnection(JDBC_URL,
USERNAME, PASSWORD);

Statement statement = connection.createStatement()) {

// SQL query to create the "employees" table

String createTableSQL = "CREATE TABLE employees (" +
"employee_id INT AUTO_INCREMENT PRIMARY KEY," +
"name VARCHAR(255) NOT NULL," +
"salary DECIMAL(10, 2)" +
// "CREATE SEQUENCE employee_id_seq START WITH 1
INCREMENT BY 1;" +
")";

// Execute the SQL query to create the table
statement.execute(createTableSQL);

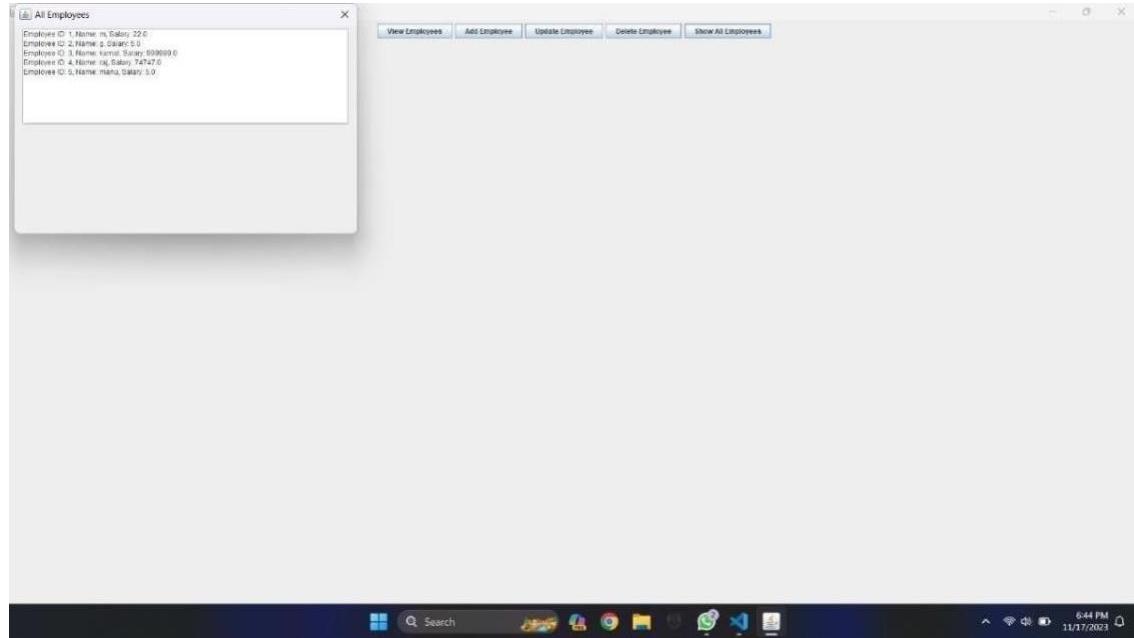
System.out.println("Table 'employees' created successfully.");

} catch (SQLException e) {
e.printStackTrace();
System.err.println("Error creating the table: " + e.getMessage()); }

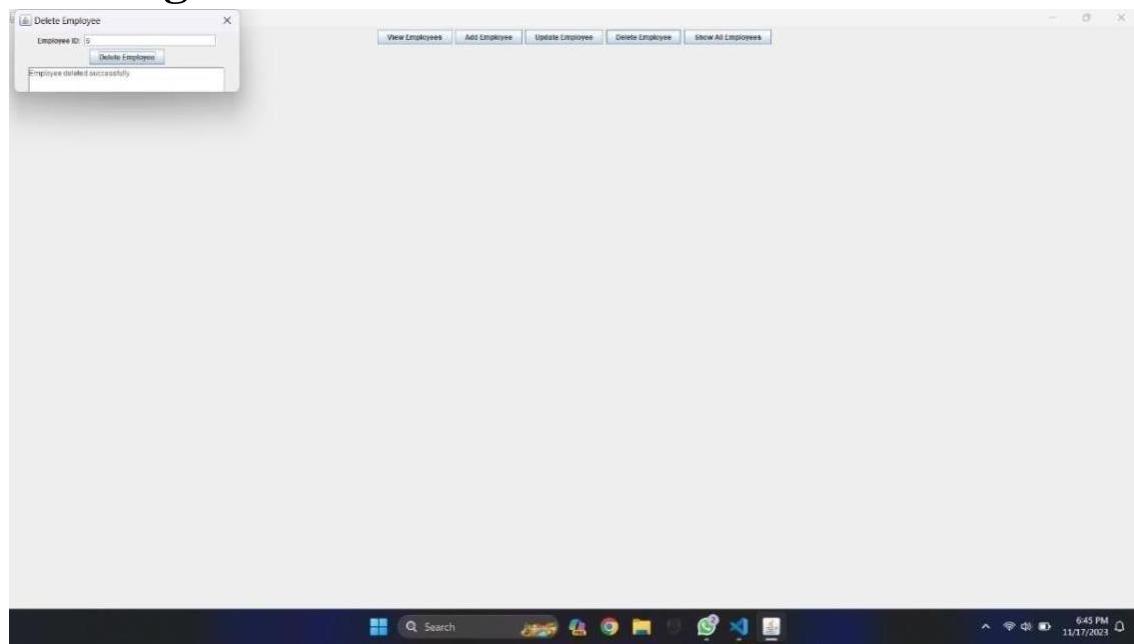
}

}
```

## Viewing details



## Deleting details



## **CONCLUSION:**

The project titled as Employee Management System was deeply studied and analyzed to design the code and implement. It was done under the guidance of the experienced project guide. All the current requirements and possibilities have been taken care during the project time.

## **REFERENCE LINKS:**

- [1] <https://www.javatpoint.com/java.awt>
- [2] <https://www.javatpoint.com/java-swing>