

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 IfOdd {
3     public static void main (String args[]){
4         Scanner obj = new Scanner (System.in);
5         int n = obj.nextInt();
6         int result = (n%2==0)?1:2;
7         System.out.println(result);
8     }
9 }
```

	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     public static void main (String args[]){
4         Scanner obj = new Scanner(System.in);
5         int n = obj.nextInt();
6         int l = Math.abs(n)%10;
7         System.out.println(l);
8     }
9 }
```

	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: The 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
267 154	11
267 -154	11
-267 154	11
-267 -154	11

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class lastdigit {
3     public static void main (String args[]){
4         Scanner obj = new Scanner (System.in);
5         int n = obj.nextInt();
6         int m = obj.nextInt();
7         int k = Math.abs(n)%10 + Math.abs(m)%10;
8         System.out.println(k);
9     }
10 }
```

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

Passed all tests! ✓

//

◀ Lab-01-MCQ

Jump to...

Is Even? ►

Question 1

Correct

Marked out of 5.00

Write a program that takes as parameter an integer n.

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

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

Note: $n! < 10^5$

Example Input:

3

Output:

0

Example Input:

60

Output:

14

Example Input:

100

Output:

24

Example Input:

1024

Output:

253

For example:

Input	Result
3	0
60	14
100	24
1024	253

Answer: (penalty regime: 0 %)

```

1 // Java program to count trailing 0s in n!
2 import java.io.*;
3 import java.util.Scanner;
4 class prog {
5     // Function to return trailing
6     // 0s in factorial of n
7     static int findTrailingZeros(int n)
8     {
9         int count = 0;
10        if (n < 0) // Negative Number Edge Case
11            return -1;
12
13        // Initialize result
14
15        // Keep dividing n by powers

```

```

17     // keep dividing n by powers
18     // of 5 and update count
19     for (int i = 5; n / i >= 1; i*=5)
20         count += n / i;
21
22     return count;
23 }
24
25 // Driver Code
26 public static void main(String[] args)
27 {
28     int n ;
29     Scanner sc= new Scanner(System.in);
30     n = sc.nextInt();
31     int result = findTrailingZeros(n);
32     System.out.println(result);
33 }
34

```

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

Passed all tests! ✓



Question 2

Correct

Marked out of 5.00

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
3 public class Custom{
4     public static int findNth(int n){
5         if(n == 0) return 0;
6         if (n == 1 || n==2) return 1;
7
8         int t0 = 0 , t1 = 1 , t2 =1 ;
9         int term = 0;
10
11        for (int i = 3 ; i<= n ; i++){
12            term = t0 +t1+t2;
13            t0 = t1;
14            t1 = t2;
15            t2 = term;
16        }
17        return t1;
18    }
19    public static void main(String args[]){
20        Scanner sc = new Scanner(System.in);
21        int n = sc.nextInt();
22        int result = findNth(n);
23        System.out.print(result);
24    }
}

```

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

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

Consider the following sequence:

1st term: 1

2nd term: 1 2 1

3rd term: 1 2 1 3 1 2 1

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

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

Example Input:

1

Output:

1

Example Input:

4

Output:

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

For example:

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

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner ;
2
3 public class Seq{
4     public static String generator(int n){
5         if (n == 1){
6             return "1";
7         }
8         String prev = generator(n-1);
9         return prev+" "+n+" "+prev;
10    }
11    public static void main (String args[]){
12        Scanner sc = new Scanner(System.in);
13        int n = sc.nextInt();
14        String r = generator(n);
15        System.out.print(r);
16    }
17 }
```

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

Passed all tests! ✓

◀ Lab-02-MCQ

Jump to...

Lab-03-MCQ ►

Question 1

Correct

Marked out of 5.00

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

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

This is explained below:

Example 1:

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

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

Step 1:

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

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

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

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

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

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

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

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

Step 2:

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

{1, 25, 16, 49, 16}

Step 3:

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

Note:

- 1) While picking up a number in Step1, if you observe that the number is smaller than the required position then use 0.
- 2) In the given function, input1[] is the array of numbers and input2 represents the number of elements in input1.

Example 2:

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

Step 1:

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

{1, 0, 4, 0, 6}

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

Step 2:

{1, 0, 16, 0, 36}

Step 3:

The final result = 53.

For example:

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

Answer: (penalty regime: 0 %)

```

1 import java.util.*;
2
3 public class SumPosition{
4     public static void main (String args[]){
5         Scanner sc = new Scanner(System.in);
6         int n = sc.nextInt();
7         int[] a = new int[n];
8         int[] b = new int[n];
9         for (int i=0 ; i<n ; i++){
10             a[i] = sc.nextInt();
11         }
12         int x=0, y = 0;
13         for (int i=0 ; i<n ; i++){
14             y = (int) Math.pow(10,i+1);
15             x = (a[i]/(y/10))%10 ;
16             b[i] = x*x ;
17         }
18         int sum = 0;
19         for (int i=0 ; i<n ; i++){
20             sum += b[i];
21         }
22         System.out.print(sum);
23     }
24 }
```

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

Passed all tests! ✓

Question 2

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
```

```

3 public class Longest{
4     public static void main(String[] args){
5         Scanner sc = new Scanner(System.in);
6         int n = sc.nextInt();
7         int[] arr = new int[n];
8         for (int i = 0 ; i<n ; i++){
9             arr[i] = sc.nextInt();
10        }
11
12        int maxL = 0 , currentL = 0 , currentS = 0 , maxS = 0 ;
13        boolean hasPositive = false ;
14
15        for (int i = 0 ; i<n ; i++){
16            if (arr[i]>=0){
17                hasPositive = true ;
18                currentL++;
19                currentS += arr[i];
20            }
21            else {
22                if (currentL > maxL){
23                    maxL = currentL;
24                    maxS = currentS;
25                }
26                else if (currentL == maxL){
27                    maxS += currentS ;
28                }
29                currentL = 0;
30                currentS = 0;
31            }
32        }
33        if (currentL > maxL){
34            maxL = currentL;
35            maxS = currentS;
36        }
37        else if (currentL == maxL){
38            maxS += currentS;
39        }
40        if (!hasPositive){
41            System.out.println(-1);
42        }
43        else {
44            System.out.println(maxS);
45        }
46    }
47 }

```

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

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

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
3 public class Array{
4     public static void main (String[] args){
5         Scanner sc = new Scanner(System.in);
6         int n = sc.nextInt();
7         int[] arr = new int[n];
8         for (int i = 0 ; i<n ; i++){
9             arr[i] = sc.nextInt();
10        }
11        int max = arr[0];
12        for (int i = 1 ; i<n ; i++){
13            if(arr[i]> max){
14                max = arr[i];
15            }
16        }
17        for (int i=0 ; i<n ; i++){
18            arr[i] = (arr[i] - max) * max;
19        }
20        for (int i=0 ; i<n ; i++){
21            System.out.print(arr[i]);
22            if (i != n-1){
23                System.out.print(" ");
24            }
25        }
26    }
27 }
28 }
```

	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! ✓

◀ Lab-03-MCQ

Jump to...

Simple Encoded Array ►

Question 1

Correct

Marked out of 5.00

Create a Class Mobile with the attributes listed below,

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

Define a Parameterized constructor to initialize the above instance variables.

Define getter and setter methods for the attributes above.

for example : setter method for manufacturer is

```
void setManufacturer(String manufacturer){
    this.manufacturer= manufacturer;
}
```

```
String getManufacturer(){
    return manufacturer;}
```

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

For example:

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

Answer: (penalty regime: 0 %)

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

```
34
35     }
36     public int getCost(){
37         return cost;
38     }
39     public String toString(){
40         return "manufacturer = " + manufacturer + "\n" + "operating_system = " + operating_system + "\n" + "color
41     }
42 }
43
44 public class Main {
45     public static void main(String[] args){
46         Mobile myMobile = new Mobile("Samsung", "Andriod", "Black", 500);
47         // System.out.println(myMobile.toString());
48         myMobile.setManufacturer("Redmi");
49         myMobile.setOperatingSystem("Andriod");
50         myMobile.color = "Blue";
51         myMobile.setCost(34000);
52 }
```

	Test	Expected	Got	
✓	1	manufacturer = Redmi operating_system = Andriod color = Blue cost = 34000	manufacturer = Redmi operating_system = Andriod color = Blue cost = 34000	✓

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

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

Student()

Student(String name)

Student(String name, int rollno)

Input:

No input

Output:**No-arg constructor is invoked****1 arg constructor is invoked****2 arg constructor is invoked****Name =null , Roll no = 0****Name =Rajalakshmi , Roll no = 0****Name =Lakshmi , Roll no = 101****For example:**

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

Answer: (penalty regime: 0 %)

```

1 public class Student{
2     private String name ;
3     private int rollNo;
4
5     public Student(){
6         System.out.println("No-arg constructor is invoked");
7         this.name = null;
8         this.rollNo = 0;
9     }
10    public Student(String name){
11        System.out.println("1 arg constructor is invoked");
12        this.name = name ;
13        this.rollNo = 0;
14    }
15    public Student(String name, int rollNo){
16        System.out.println("2 arg constructor is invoked");
17        this.name = name;
18        this.rollNo = rollNo;
19    }
20    public String getDetails(){
21        return "Name =" + name +", Roll no =" + rollNo;
22    }
23    public static void main(String[] args){
24        Student student1 = new Student();
25        Student student2 = new Student("Rajalakshmi");
26        Student student3 = new Student("Lakshmi", 101);
27
28        System.out.println(student1.getDetails());
29        System.out.println(student2.getDetails());
30        System.out.println(student3.getDetails());
31    }
32 }
```



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

Passed all tests! ✓



Question 3

Correct

Marked out of 5.00

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

Area of Circle = πr^2

Circumference = $2\pi r$

Input:

2

Output:

Area = 12.57

Circumference = 12.57

For example:

Test	Input	Result
1	4	Area = 50.27 Circumference = 25.13

Answer: (penalty regime: 0 %)

Reset answer

```

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

41
42

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

Passed all tests! ✓

[◀ Lab-04-MCQ](#)

Jump to...

[Number of Primes in a specified range ►](#)

Question 1

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

```
29     AndroidMobile androidMobile = new AndroidMobile();
30
31     androidMobile.newFeature();
32     androidMobile.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! ✓

//

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    public void admitted() {
11        System.out.println("A student admitted in "+collegeName);
12    }
13 }
14 class Student extends College{
15
16     protected String studentName;
17     protected String department;
18
19     public Student(String collegeName, String studentName, String department) {
20         // initialize the instance variables
21         super(collegeName);
22         this.studentName = studentName;
23         this.department = department ;
24     }
25     @Override
26     public String toString(){
27         // return the details of the student
28     }
```

```

28     return collegeName + "\n" + studentName + "\n" + department;
29 }
30 }
31 class CSE extends Student {
32 public CSE (String collegeName , String studentName){
33     super(collegeName, studentName, "CSE");
34 }
35 }
36 public class Main {
37 public static void main (String[] args) {
38     CSE student = new CSE("REC","Venkatesh");
39     student.admitted();                                // invoke the admitted() method
40     System.out.println(student.toString());
41 }
42 }

```

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

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

For example:

Result

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

Answer: (penalty regime: 0 %)

[Reset answer](#)

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

```

43  // Constructor to initialize account number and balance
44  public SavingsAccount(String accountNumber, double balance) {
45      // Call the parent class constructor
46      super(accountNumber, balance);
47  }
48
49  // Override the withdraw method from the parent class
50  @Override
51  public void withdraw(double amount) {
52      // Check if the withdrawal would cause the balance to drop below $100

```

	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! ✓

◀ 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 2nd position is b

Word2 is za, both are not same alphabets

position value of z is 26

position value of a is 1

max – min will be $26 - 1 = 25$

Alphabet which comes in 25th position is y

word3 is ee, both are same hence take e

Hence the output is BYE

For example:

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

Answer: (penalty regime: 0 %)

```

1 import java.util.*;
2 class diff{
3     char different(char a, char b){
4         if ((int)a != (int)b)
5             return (char)((int)'a' + ((int)a-(int)b) - 1);
6         return a;
7     }
8 }
9 public class Main{
10    public static void main(String[] args){
11        Scanner scan = new Scanner(System.in);
12        diff z = new diff();
13        String q = scan.nextLine();
14        StringBuffer ans = new StringBuffer();
15        StringBuffer temp = new StringBuffer();
16        for(int i = 0;i < q.length();i++){
17            if(q.charAt(i) == ':'){
18                temp.append(" ");
19            }
20            else{
21                temp.append(Character.toString(q.charAt(i)));
22            }
23        }
24        String h = temp.toString();
25        for(int i = 0;i < temp.length();i++){
26            if(i%3 == 0){
27                ans.append(Character.toString(z.different(h.charAt(i),h.charAt(i+1))));
28            }
29        }
30        System.out.print(ans.toString().toUpperCase());
31    }
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.*;  
2 public class mix{  
3     public static void main(String[] args){
```

```

4   Scanner scan = new Scanner(System.in);
5   String g = scan.nextLine();
6   int n = scan.nextInt(), ones, flag = 0;
7   StringBuffer temp = new StringBuffer();
8   StringBuffer temp1 = new StringBuffer();
9   int space = 0;
10  while (n > 0){
11      ones = (n %10) - 1;
12      for(int i = 0; i < g.length();i++){
13          if (g.charAt(i) == ' '){
14              space = space + 1;
15          }
16          else if(space == ones && flag == 0){
17              temp.append(Character.toString(g.charAt(i)));
18          }
19          else if(space == ones && flag == 1){
20              temp1.append(Character.toString(g.charAt(i)));
21          }
22      }
23      space = 0 ;
24      flag = 1;
25      n = n /10;
26  }
27  rew m = new rew();
28  System.out.println(m.r(temp1.toString()) + " " + m.r(temp.toString()));
29 }
30 }
31 class rew{
32     String r(String a){
33         int le = a.length(),n,q;
34         StringBuffer temp3 = new StringBuffer();
35         if(le % 2 == 1){
36             n = ((int)(le/2));
37             q = ((int)(le/2));
38         }
39         else{
40             n = ((int)(le/2)) - 1;
41             q = ((int)(le/2));
42         }
43         for(int i = n;i >= 0;i--){
44             temp3.append(Character.toString(a.charAt(i)));
45         }
46         for(int i = q;i < le;i++){
47             temp3.append(Character.toString(a.charAt(i)));
48         }
49         return temp3.toString();
50     }
51 }
52

```

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

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

Given 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.*;
2
3
4 public class HelloWorld {
5     public static void main(String[] args) {
6         Scanner scan = new Scanner(System.in);
7         String a = scan.nextLine();
8         String b = scan.nextLine();
9         StringBuffer ab = new StringBuffer();
10        if(a.trim().isEmpty() && b.trim().isEmpty()){
11            System.out.print("null");
12        }
13        else{
14            for(int i = 0;i < a.length();i++){
15                if (a.charAt(i) != ' ') {
16                    ab.append(Character.toString(a.charAt(i)));
17                }
18            }
19            for(int i = 0;i < b.length();i++){
20                if (b.charAt(i) != ' ') {
21
22
23
24
25
26
27
28
29
2
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
5
60
61
62
63
64
65
66
67
68
69
6
70
71
72
73
74
75
76
77
78
79
7
80
81
82
83
84
85
86
87
88
89
8
90
91
92
93
94
95
96
97
98
99
9
100
101
102
103
104
105
106
107
108
109
10
110
111
112
113
114
115
116
117
118
119
11
120
121
122
123
124
125
126
127
128
129
12
130
131
132
133
134
135
136
137
138
139
13
140
141
142
143
144
145
146
147
148
149
14
150
151
152
153
154
155
156
157
158
159
15
160
161
162
163
164
165
166
167
168
169
16
170
171
172
173
174
175
176
177
178
179
17
180
181
182
183
184
185
186
187
188
189
18
190
191
192
193
194
195
196
197
198
199
19
1
200
201
202
203
204
205
206
207
208
209
20
210
211
212
213
214
215
216
217
218
219
21
220
221
222
223
224
225
226
227
228
229
22
230
231
232
233
234
235
236
237
238
239
23
240
241
242
243
244
245
246
247
248
249
24
250
251
252
253
254
255
256
257
258
259
25
260
261
262
263
264
265
266
267
268
269
26
270
271
272
273
274
275
276
277
278
279
27
280
281
282
283
284
285
286
287
288
289
28
290
291
292
293
294
295
296
297
298
299
29
2
300
301
302
303
304
305
306
307
308
309
30
310
311
312
313
314
315
316
317
318
319
31
320
321
322
323
324
325
326
327
328
329
32
330
331
332
333
334
335
336
337
338
339
33
340
341
342
343
344
345
346
347
348
349
34
350
351
352
353
354
355
356
357
358
359
35
360
361
362
363
364
365
366
367
368
369
36
370
371
372
373
374
375
376
377
378
379
37
380
381
382
383
384
385
386
387
388
389
38
390
391
392
393
394
395
396
397
398
399
39
3
400
401
402
403
404
405
406
407
408
409
40
410
411
412
413
414
415
416
417
418
419
41
420
421
422
423
424
425
426
427
428
429
42
430
431
432
433
434
435
436
437
438
439
43
440
441
442
443
444
445
446
447
448
449
44
450
451
452
453
454
455
456
457
458
459
45
460
461
462
463
464
465
466
467
468
469
46
470
471
472
473
474
475
476
477
478
479
47
480
481
482
483
484
485
486
487
488
489
48
490
491
492
493
494
495
496
497
498
499
49
4
500
501
502
503
504
505
506
507
508
509
50
510
511
512
513
514
515
516
517
518
519
51
520
521
522
523
524
525
526
527
528
529
52
530
531
532
533
534
535
536
537
538
539
53
540
541
542
543
544
545
546
547
548
549
54
550
551
552
553
554
555
556
557
558
559
55
560
561
562
563
564
565
566
567
568
569
56
570
571
572
573
574
575
576
577
578
579
57
580
581
582
583
584
585
586
587
588
589
58
590
591
592
593
594
595
596
597
598
599
59
5
600
601
602
603
604
605
606
607
608
609
60
610
611
612
613
614
615
616
617
618
619
61
620
621
622
623
624
625
626
627
628
629
62
630
631
632
633
634
635
636
637
638
639
63
640
641
642
643
644
645
646
647
648
649
64
650
651
652
653
654
655
656
657
658
659
65
660
661
662
663
664
665
666
667
668
669
66
670
671
672
673
674
675
676
677
678
679
67
680
681
682
683
684
685
686
687
688
689
68
690
691
692
693
694
695
696
697
698
699
69
6
700
701
702
703
704
705
706
707
708
709
70
710
711
712
713
714
715
716
717
718
719
71
720
721
722
723
724
725
726
727
728
729
72
730
731
732
733
734
735
736
737
738
739
73
740
741
742
743
744
745
746
747
748
749
74
750
751
752
753
754
755
756
757
758
759
75
760
761
762
763
764
765
766
767
768
769
76
770
771
772
773
774
775
776
777
778
779
77
780
781
782
783
784
785
786
787
788
789
78
790
791
792
793
794
795
796
797
798
799
79
7
800
801
802
803
804
805
806
807
808
809
80
810
811
812
813
814
815
816
817
818
819
81
820
821
822
823
824
825
826
827
828
829
82
830
831
832
833
834
835
836
837
838
839
83
840
841
842
843
844
845
846
847
848
849
84
850
851
852
853
854
855
856
857
858
859
85
860
861
862
863
864
865
866
867
868
869
86
870
871
872
873
874
875
876
877
878
879
87
880
881
882
883
884
885
886
887
888
889
88
890
891
892
893
894
895
896
897
898
899
89
8
900
901
902
903
904
905
906
907
908
909
90
910
911
912
913
914
915
916
917
918
919
91
920
921
922
923
924
925
926
927
928
929
92
930
931
932
933
934
935
936
937
938
939
93
940
941
942
943
944
945
946
947
948
949
94
950
951
952
953
954
955
956
957
958
959
95
960
961
962
963
964
965
966
967
968
969
96
970
971
972
973
974
975
976
977
978
979
97
980
981
982
983
984
985
986
987
988
989
98
990
991
992
993
994
995
996
997
998
99
9

```

```

21         ab.append(Character.toString(b.charAt(1)));
22     }
23 }
24 char[] d = ab.toString().toCharArray();
25 Arrays.sort(d);
26 for(int i = d.length - 1;i >= 1;i--){
27     if(d[i] != d[i-1])
28         System.out.print(d[i]);
29 }
30 System.out.print(d[0]);
31 }
32 }
33 }
34 }
35 }
```

	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 // Define the RBI interface
2 interface RBI {
3     // Variable declaration
4     String parentBank = "RBI";
5
6     // Abstract method
7     double rateOfInterest();
8
9     // Default method
10    default void policyNote() {
11        System.out.println("RBI has a new Policy issued in 2023");
12    }
13
14     // Static method
15    static void regulations() {
16        System.out.println("RBI has updated new regulations in 2024.");
17    }
18 }
19
20 // SBI class implementing RBI interface
21 class SBI implements RBI {
22     // Implementing the abstract method
23     public double rateOfInterest() {
24         return 7.6;
25     }
26 }
27
28 // Karur class implementing RBI interface
29 class Karur implements RBI {
30 }
```

```

30     // Implementing the abstract method
31     public double rateOfInterest() {
32         return 7.4;
33     }
34 }
35
36 // Main class to test the functionality
37 public class Main {
38     public static void main(String[] args) {
39         // RBI policies and regulations
40         RBI rbi = new SBI(); // Can be any class implementing RBI
41         rbi.policyNote(); // Default method
42         RBI.regulations(); // Static method
43
44         // SBI bank details
45         SBI sbi = new SBI();
46         System.out.println("SBI rate of interest: " + sbi.rateOfInterest() + " per annum.");
47
48         // Karur bank details
49         Karur karur = new Karur();
50         System.out.println("Karur rate of interest: " + karur.rateOfInterest() + " per annum.");
51     }
52 }
```

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

Passed all tests! ✓

Question 2

Correct

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
3 // Define the Playable interface
4 interface Playable {
5     // Abstract method to play the respective sport
6     void play();
7 }
8
9 // Football class implementing Playable interface
10 class Football implements Playable {
11     String name;
12
13     // Constructor
14     public Football(String name) {
15         this.name = name;
16     }
17
18     // Override the play method
19     public void play() {
20         System.out.println(name + " is Playing football");
21     }
22 }
23
24 // Volleyball class implementing Playable interface
25 class Volleyball implements Playable {
26     String name;
```

```

26     String name;
27
28     // Constructor
29     public Volleyball(String name) {
30         this.name = name;
31     }
32
33     // Override the play method
34     public void play() {
35         System.out.println(name + " is Playing volleyball");
36     }
37 }
38
39 // Basketball class implementing Playable interface
40 class Basketball implements Playable {
41     String name;
42
43     // Constructor
44     public Basketball(String name) {
45         this.name = name;
46     }
47
48     // Override the play method
49     public void play() {
50         System.out.println(name + " is Playing basketball");
51     }
52 }
```

	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  
3 interface Sports {  
4     void setHomeTeam(String name);  
5     void setVisitingTeam(String name);  
6 }  
7  
8 interface Football extends Sports {  
9     void homeTeamScored(int points);  
10    void visitingTeamScored(int points);  
11 }  
12  
13 class College implements Football {  
14     private String homeTeam;  
15     private String visitingTeam;  
16     private int homeTeamPoints = 0;  
17     private int visitingTeamPoints = 0;  
18  
19     public void setHomeTeam(String name) {  
20         this.homeTeam = name;  
21     }  
22  
23     public void setVisitingTeam(String name) {  
24         this.visitingTeam = name;  
25     }  
26  
27     public void homeTeamScored(int points) {  
28         // logic to calculate points  
29     }  
30 }
```

```

28     homeTeamPoints += points;
29     System.out.println(homeTeam + " " + points + " scored");
30 }
31
32 public void visitingTeamScored(int points) {
33     visitingTeamPoints += points;
34     System.out.println(visitingTeam + " " + points + " scored");
35 }
36
37 public void winningTeam() {
38     if (homeTeamPoints > visitingTeamPoints) {
39         System.out.println(homeTeam + " is the winner!");
40     } else if (homeTeamPoints < visitingTeamPoints) {
41         System.out.println(visitingTeam + " is the winner!");
42     } else {
43         System.out.println("It's a tie match.");
44     }
45 }
46
47
48 public class Main {
49     public static void main(String[] args) {
50         Scanner sc = new Scanner(System.in);
51
52         // Get home team name

```

	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 ▶

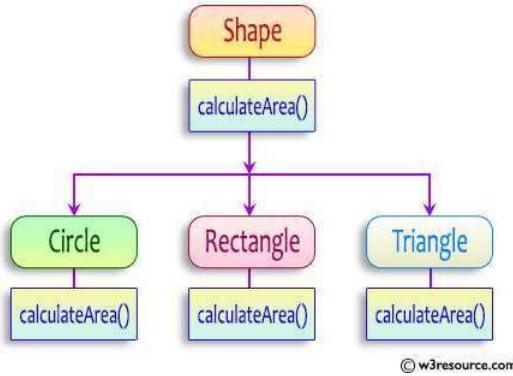
Question 1

Correct

Marked out of 5.00

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

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



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

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

sample Input :

```
4 // radius of the circle to calculate area PI*r*r
5 // length of the rectangle
6 // breadth of the rectangle to calculate the area of a rectangle
4 // base of the triangle
3 // height of the triangle
```

OUTPUT:

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

For example:

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

Answer: (penalty regime: 0 %)

```
1 import java.util.Scanner;
2
3 // Abstract class Shape
4 abstract class Shape {
5     public abstract double calculateArea();
```

```

6 }
7
8 // Circle class
9 class Circle extends Shape {
10     private double radius;
11
12     public Circle(double radius) {
13         this.radius = radius;
14     }
15
16     @Override
17     public double calculateArea() {
18         return Math.PI * radius * radius; // Area of circle:  $\pi r^2$ 
19     }
20 }
21
22 // Rectangle class
23 class Rectangle extends Shape {
24     private double length;
25     private double breadth;
26
27     public Rectangle(double length, double breadth) {
28         this.length = length;
29         this.breadth = breadth;
30     }
31
32     @Override
33     public double calculateArea() {
34         return length * breadth; // Area of rectangle: length * breadth
35     }
36 }
37
38 // Triangle class
39 class Triangle extends Shape {
40     private double base;
41     private double height;
42
43     public Triangle(double base, double height) {
44         this.base = base;
45         this.height = height;
46     }
47
48     @Override
49     public double calculateArea() {
50         return 0.5 * base * height; // Area of triangle:  $0.5 * base * height$ 
51     }
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 2

Correct

Marked out of 5.00

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

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

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

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

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

input2: String array.

Example 1:

input1: 3

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

output: oreoapple

Example 2:

input1: 2

input2: {"Mango", "banana"}

output: no matches found

Explanation:

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

Hence the output is no matches found.

Example 3:

input1: 3

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

output: ateace

For example:

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

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2
3 public class VowelStringExtractor {
4
5     // Method to extract strings with vowels as first and last characters
6     public static String extractVowelStrings(String[] stringArray) {
7         StringBuilder result = new StringBuilder();
8         String vowels = "aeiouAEIOU"; // String containing all vowels
9
10        // Iterate through the array of strings
11        for (String s : stringArray) {

```

```

12     // Check if the string is not empty and if both the first and last characters are vowels
13     if (s.length() > 0 && vowels.indexOf(s.charAt(0)) != -1 && vowels.indexOf(s.charAt(s.length() - 1)) != -1)
14         result.append(s); // Append matching string to the result
15     }
16 }
17
18 // Return the concatenated string in lowercase or "no matches found"
19 return result.length() > 0 ? result.toString().toLowerCase() : "no matches found";
20 }
21
22 public static void main(String[] args) {
23     Scanner scanner = new Scanner(System.in);
24
25     // Input for the number of strings
26
27     int n = scanner.nextInt();
28     scanner.nextLine(); // Consume the newline character
29
30     // Input for the strings in one line
31
32     String input = scanner.nextLine();
33     String[] strings = input.split(" "); // Split input into an array
34
35     // Process and output the result
36     String result = extractVowelStrings(strings);
37     System.out.println(result);
38
39     scanner.close(); // Close the scanner
40 }
41
42
43

```

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

Passed all tests! ✓

Question 3

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 // Final class definition
2 final class FinalExample {
3     // Final variable
4     final int MAX_SPEED = 120; // Constant value
5
6     // Final method
7     public final void display() {
8         System.out.println("The maximum speed is: " + MAX_SPEED + " km/h");
9     }
10 }
11
12 // Main class to test the final class
13 public class Test {
14     public static void main(String[] args) {
15         // Create an instance of FinalExample
16         FinalExample example = new FinalExample();
17         example.display();
18
19         // Uncommenting the following line will result in a compile-time error
20         // because FinalExample is a final class and cannot be subclassed.
21         // class SubclassExample extends FinalExample { }
22
23         System.out.println("This is a subclass of FinalExample.");
24     }
}
```

25 }
26 |

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

Passed all tests! ✓

[◀ Lab-08-MCQ](#)

Jump to...

[FindStringCode ►](#)

Question 1

Correct

Marked out of 5.00

Write a Java program to create a method that takes an integer as a parameter and throws an exception if the number is odd.

Sample input and Output:

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

Fill the preloaded answer to get the expected output.

For example:

Result
82 is even. Error: 37 is odd.

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

1 class prog {
2     public static void main(String[] args) {
3         int n1 = 82;
4         int n2 = 37;
5         // call the trynumber(n);
6         try {
7             trynumber(n1);
8         } catch (Exception e) {
9             System.out.println(e.getMessage());
10        }
11        try {
12            trynumber(n2);
13        }
14    }
15    catch (Exception e) {
16        System.out.println(e.getMessage());
17    }
18 }
19
20
21 public static void trynumber(int n) throws Exception {
22     checkEvenNumber(n);
23     System.out.println(n + " is even.");
24 }
25
26
27 public static void checkEvenNumber(int number) throws Exception {
28     if (number % 2 != 0) {
29         throw new Exception ("Error: " + number + " is odd.");
30     }
31 }
32
33 }
```

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

Passed all tests! ✓

Question 2

Correct

Marked out of 5.00

In the following program, an array of integer data is to be initialized.

During the initialization, if a user enters a value other than an integer, it will throw an InputMismatchException exception.

On the occurrence of such an exception, your program should print "You entered bad data."

If there is no such exception it will print the total sum of the array.

```
/* Define try-catch block to save user input in the array "name"
```

```
If there is an exception then catch the exception otherwise print the total sum of the array. */
```

Sample Input:

```
3  
5 2 1
```

Sample Output:

```
8
```

Sample Input:

```
2
```

```
1 g
```

Sample Output:

```
You entered bad data.
```

For example:

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

Answer: (penalty regime: 0 %)

```
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  
9 int[] name = new int[length];  
10 int sum=0;//save the total sum of the array.  
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 name[i] = sc.nextInt(); // User input  
19 sum += name[i]; // Add to total sum  
20 } // Print the total sum of the array  
21 System.out.println(sum);  
22 } catch (InputMismatchException e) {  
23 System.out.println("You entered bad data.");  
24  
25 } finally {  
26 sc.close();
```

```
27 }  
28 }  
29 }  
30 }
```

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

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

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
3 public class ExceptionHandlingExample {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         try {
8             int n = scanner.nextInt();
9
10            // Initialize the array
11            int[] array = new int[n];
12
13            for (int i = 0; i < n; i++) {
14                array[i] = scanner.nextInt();
15            }
16
17            // Try dividing the element at index 0 by the element at index 1
18            int result = array[0] / array[1]; // Potential ArithmaticException
19
20            // System.out.println("Division result: " + result);
21
22            // Try accessing an element beyond the array bounds
23            //System.out.println("Accessing element at index " + n);
24            System.out.println("Element: " + array[n]); // Potential ArrayIndexOutOfBoundsException
25
26        } catch (ArithmaticException e) {
27            // Handle division by zero
28            System.out.println(e.toString());
29        } catch (ArrayIndexOutOfBoundsException e) {
30            ...
31        }
32    }
33}
```

```

30     // Handle accessing an out-of-bound array index
31     System.out.println(e.toString());
32 } finally {
33     // This block is always executed
34     System.out.println("I am always executed");
35 }
36
37 // Close the scanner
38 scanner.close();
39 }
40 }
41

```

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

Passed all tests! ✓

◀ 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.ArrayList;
2 import java.util.Scanner;
3
4 ✓ public class Main {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7
8         int n = scanner.nextInt();
9
10        ArrayList<Integer> list = new ArrayList<>();
11
12        for (int i = 0; i < n; i++) {
13            list.add(scanner.nextInt());
14        }
15
16        int firstElement = list.get(0);
17
18        // Get the last element
19        int lastElement = list.get(list.size() - 1);
20
21        // Display results
22        System.out.println("ArrayList: " + list);
23        System.out.println("First : " + firstElement + ", Last : " + lastElement);
24
25        scanner.close();
26    }
27 }
```

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

Passed all tests! ✓

Question 2

Correct

Marked out of 1.00

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

```
list.set();
list.indexOf();
list.lastIndexOf()
list.contains()
list.size();
list.add();
list.remove();
```

The above methods are used for the below Java program.

Answer: (penalty regime: 0 %)

[Reset answer](#)

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

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

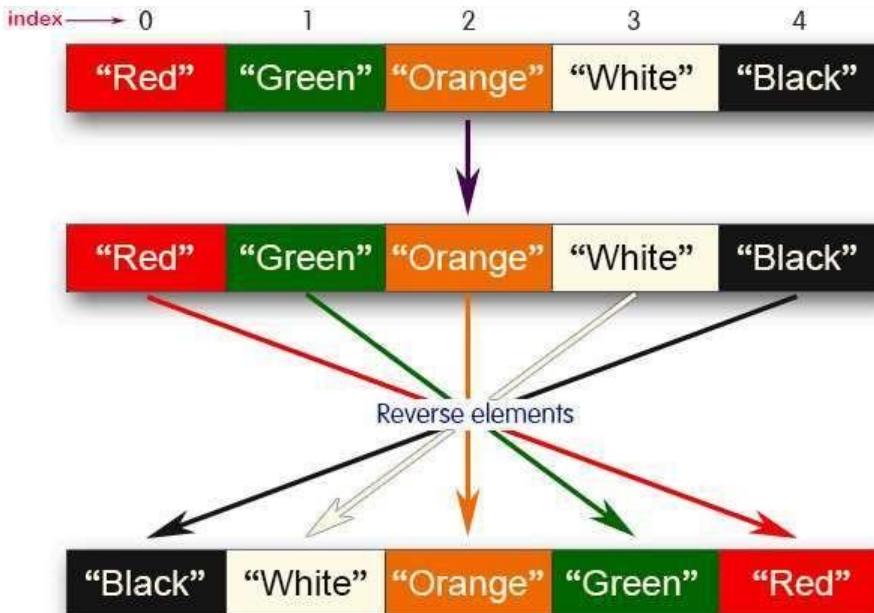
Passed all tests! ✓

Question 3

Correct

Marked out of 1.00

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



Sample input and Output:

Red

Green

Orange

White

Black

Sample output

List before reversing :

[Red, Green, Orange, White, Black]

List after reversing :

[Black, White, Orange, Green, Red]

Answer: (penalty regime: 0 %)

```

1 import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.Scanner;
4
5 public class ReverseArrayList {
6
7     public static void main(String[] args) {
8         Scanner scanner = new Scanner(System.in);
9         ArrayList<String> list = new ArrayList<>();
10
11         // Prompting the user to enter elements
12
13         int n = scanner.nextInt();
14         scanner.nextLine(); // Consume newline
15
16
17         for (int i = 0; i < n; i++) {
18
19             String element = scanner.nextLine();
20             list.add(element);
21         }
22
23         // Printing the list before reversing
24         System.out.println("List before reversing :");
25         System.out.println(list);
    
```

```

26
27     // Reversing the ArrayList
28     Collections.reverse(list);
29
30     // Printing the list after reversing
31     System.out.println("List after reversing :");
32     System.out.println(list);
33
34     scanner.close();
35 }
36 }
```

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

Passed all tests! ✓

◀ Lab-10-MCQ

Jump to...

Lab-11-MCQ ▶

Question 1

Correct

Marked out of 1.00

Java HashSet class implements the Set interface, backed by a hash table which is actually a [HashMap](#) instance.

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

This class permits the null element.

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

Java HashSet Features

A few important features of HashSet are mentioned below:

- Implements [Set Interface](#).
- The underlying data structure for HashSet is [Hashtable](#).
- As it implements the Set Interface, duplicate values are not allowed.
- Objects that you insert in HashSet are not guaranteed to be inserted in the same order. Objects are inserted based on their hash code.
- NULL elements are allowed in HashSet.
- HashSet also implements **Serializable** and **Cloneable** interfaces.

• `public class HashSet<E> extends AbstractSet<E> implements Set<E>, Cloneable, Serializable`
Sample Input and Output:

5

90

56

45

78

25

78

Sample Output:

78 was found in the set.

Sample Input and output:

3

2

7

9

5

Sample Input and output:

5 was not found in the set.

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

1 import java.util.HashSet;
2 import java.util.Scanner;
3
4 class prog {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         int n = sc.nextInt();
8
9         // Create a HashSet object called numbers
10        HashSet<Integer> numbers = new HashSet<>();
11
12        // Add values to the set
13        for (int i = 0; i < n; i++) {
14            numbers.add(sc.nextInt());
15        }
16
17        int skey = sc.nextInt();
18
19        // Check if skey is in the set and show the result
20        if (numbers.contains(skey)) {
21            System.out.println("Element found in the set");
22        } else {
23            System.out.println("Element not found in the set");
24        }
25    }
26}
```

```
21     System.out.println(skey + " was found in the set. );  
22 } else {  
23     System.out.println(skey + " was not found in the set.");  
24 }  
25  
26 // Display which numbers between 1 and 10 are in the set  
27 for (int i = 1; i <= 10; i++) {  
28     if (numbers.contains(i)) {  
29     }  
30 }  
31  
32 sc.close();  
33 }  
34 }
```

	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
4 public class Main {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         // Read the first set
9         int n1 = sc.nextInt();
10        sc.nextLine(); // Consume newline character
11        HashSet<String> set1 = new HashSet<>();
12        for (int i = 0; i < n1; i++) {
13            set1.add(sc.nextLine());
14        }
15
16        // Read the second set
17        int n2 = sc.nextInt();
18        sc.nextLine(); // Consume newline character
19        HashSet<String> set2 = new HashSet<>();
20        for (int i = 0; i < n2; i++) {
21            set2.add(sc.nextLine());
22        }
23
24        // Retain only elements that are present in both sets
25        set1.retainAll(set2);
26
27        // Output the result
28        for (String element : set1) {
```

```

29         System.out.println(element);
30     }
31
32     sc.close();
33 }
34 }
```

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

Passed all tests! ✓

✓

Question 3

Correct

Marked out of 1.00

Java HashMap Methods

[containsKey\(\)](#). Indicate if an entry with the specified key exists in the map[containsValue\(\)](#). Indicate if an entry with the specified value exists in the map[putIfAbsent\(\)](#) Write an entry into the map but only if an entry with the same key does not already exist[remove\(\)](#). Remove an entry from the map[replace\(\)](#) [Write to an entry in the map only if it exists](#)[size\(\)](#). Return the number of entries in the map

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

Answer: (penalty regime: 0 %)[Reset answer](#)

```

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

```
50     System.out.println(value);  
51  
52     // Checking whether key 'ONE' exists in map
```

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

Passed all tests! ✓

◀ Lab-11-MCQ

Jump to...

TreeSet example ►



Question 1

Correct

Marked out of 5.00

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

Get the ASCII values of all the extracted alphabets.

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

Return that single digit as output.

Note:

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

Example 1:

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

input2: {'b', 'c'}

output: 8

Explanation:

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

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

$$98 + 99 = 197$$

$$1 + 9 + 7 = 17$$

$$1 + 7 = 8$$

For example:

Input	Result
a b c	8
b c	

Answer: (penalty regime: 0 %)

```

1 import java.util.HashSet;
2 import java.util.Scanner;
3 import java.util.Set;
4
5 public class CommonAsciiSum {
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8
9         // Input for first array
10        //System.out.print("Enter characters for input1 (space-separated): ");
11        String[] input1Array = scanner.nextLine().split(" ");
12        char[] input1 = new char[input1Array.length];
13        for (int i = 0; i < input1Array.length; i++) {
14            input1[i] = input1Array[i].charAt(0);
15        }
16
17        // Input for second array
18        //System.out.print("Enter characters for input2 (space-separated): ");
19        String[] input2Array = scanner.nextLine().split(" ");
20        char[] input2 = new char[input2Array.length];
21        for (int i = 0; i < input2Array.length; i++) {
22            input2[i] = input2Array[i].charAt(0);
23        }
24    }
25 }
```

```

23     }
24
25     scanner.close();
26
27     System.out.println(getSingleDigitAsciiSum(input1, input2));
28 }
29
30 public static int getSingleDigitAsciiSum(char[] input1, char[] input2) {
31     Set<Character> set1 = new HashSet<>();
32     Set<Character> commonChars = new HashSet<>();
33
34     // Add elements of input1 to set1
35     for (char c : input1) {
36         set1.add(c);
37     }
38
39     // Find common elements between input1 and input2
40     for (char c : input2) {
41         if (set1.contains(c)) {
42             commonChars.add(c);
43         }
44     }
45
46     // Calculate the ASCII sum of common characters
47     int sum = 0;
48     for (char c : commonChars) {
49         sum += (int) c;
50     }
51
52     // Reduce the sum to a single digit

```

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

Passed all tests! ✓

//

Question 2

Correct

Marked out of 5.00

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

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

Each alphabet is represented by a sequence of 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 (0000000000000000000000000000).

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

Example 1:

input1: 010010001

The decoded string (original word) will be: ZYX

Example 2:

input1: 000010000000000000000010000000000001000000000000100000000000001

The decoded string (original word) will be: WIPRO

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

For example:

Input	Result
010010001	ZYX
000010000000000000000010000000000001000000000000100000000000001	WIPRO

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2
3 public class Decoder {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         //System.out.print("Enter the encoded string: ");
7         String input = scanner.nextLine();
8         scanner.close();
9
10        System.out.println(decode(input));
11    }
12
13    public static String decode(String encoded) {
14        StringBuilder decodedWord = new StringBuilder();
15        String[] parts = encoded.split("1");
16
17        for (String part : parts) {
18            int zeroCount = part.length();
19            // Calculate the corresponding character by counting zeros
20        }
21    }
22}
```

```
21     char decodedChar = (char) ('A' + (26 - zeroCount));
22     decodedWord.append(decodedChar);
23 }
24
25 return decodedWord.toString();
26 }
27 }
```

	Input	Expected	Got	
✓	010010001	ZYX	ZYX	✓
✓	000010000000000000000000100000000000100000000010000000000001	WIPRO	WIPRO	✓

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 SeigOlonhcet ErolaGnab".

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

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

NOTE:

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
3 public class WordReversal {
4
5     public static String reverseWords(String sentence, int caseOption) {
6         StringBuilder modifiedSentence = new StringBuilder();
7
8         // Split the sentence into words based on spaces
9         String[] words = sentence.split(" ");
10
11        for (String word : words) {
12            if (caseOption == 0) {
13                modifiedSentence.append(new StringBuilder(word).reverse().toString());
14            } else {
15                modifiedSentence.append(word);
16            }
17        }
18
19        return modifiedSentence.toString();
20    }
21}
```

```

11   for (int i = 0; i < words.length; i++) {
12     String reversedWord = reverseWord(words[i], caseOption);
13
14     // Append the reversed word to the modified sentence
15     modifiedSentence.append(reversedWord);
16
17     // Add space between words except after the last word
18     if (i < words.length - 1) {
19       modifiedSentence.append(" ");
20     }
21   }
22
23   return modifiedSentence.toString();
24 }
25
26 private static String reverseWord(String word, int caseOption) {
27   StringBuilder reversedChars = new StringBuilder(word).reverse();
28
29   // Apply case reversal only if caseOption is 1
30   if (caseOption == 1) {
31     for (int i = 0; i < word.length(); i++) {
32       char originalChar = word.charAt(i);
33       char reversedChar = reversedChars.charAt(i);
34
35       if (Character.isAlphabetic(originalChar)) {
36         // Retain the case of the original character
37         if (Character.isUpperCase(originalChar)) {
38           reversedChars.setCharAt(i, Character.toUpperCase(reversedChar));
39         } else {
40           reversedChars.setCharAt(i, Character.toLowerCase(reversedChar));
41         }
42       }
43     }
44   }
45
46   return reversedChars.toString();
47 }
48
49 public static void main(String[] args) {
50   Scanner scanner = new Scanner(System.in);
51
52   // Get input from user

```

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

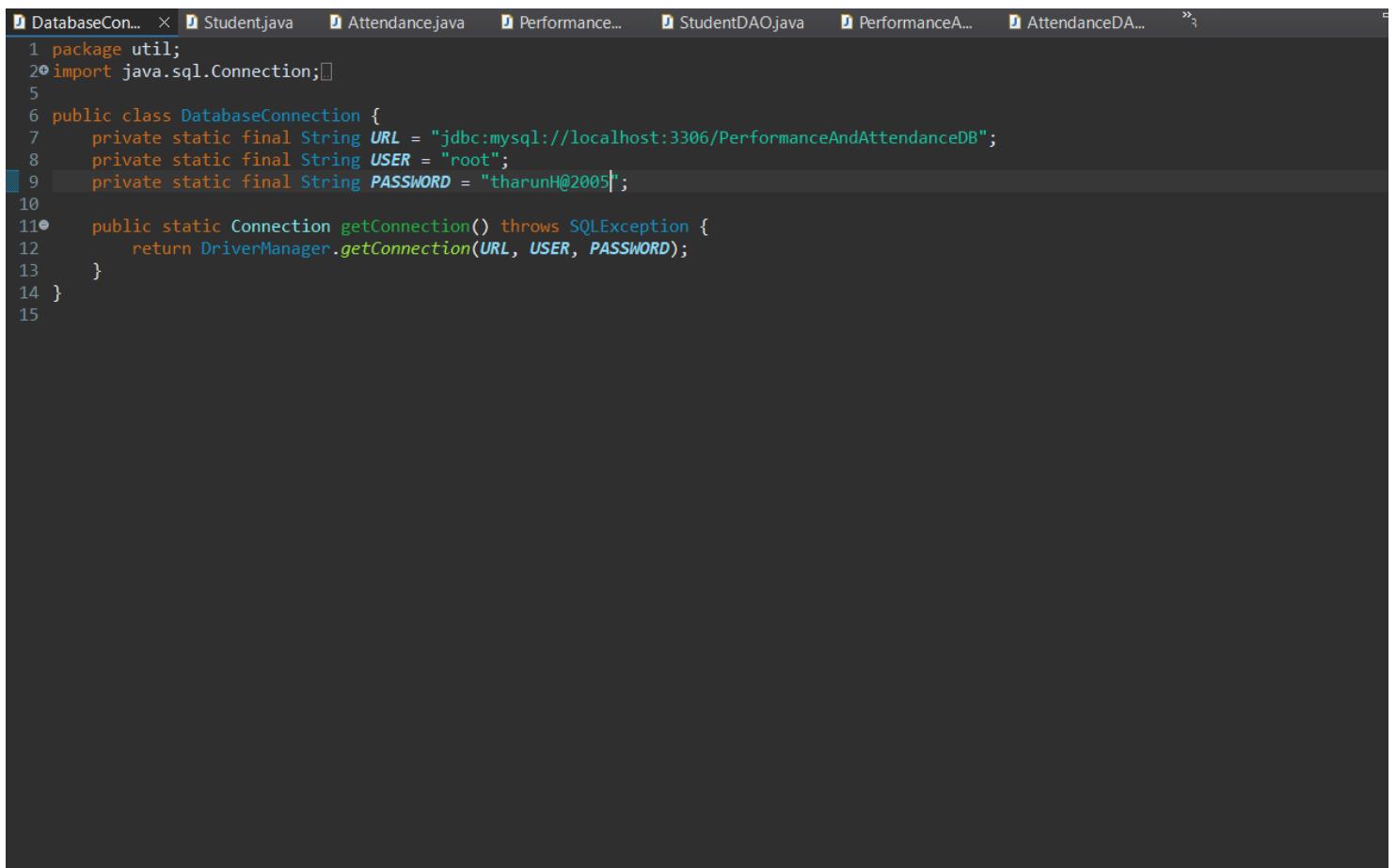
Passed all tests! ✓

[◀ Lab-12-MCQ](#)

Jump to...

[Identify possible words ▶](#)

STUDENT ATTENDANCE AND PERFORMANCE SYSTEM



```
1 package util;
2 import java.sql.Connection;
3
4 public class DatabaseConnection {
5     private static final String URL = "jdbc:mysql://localhost:3306/PerformanceAndAttendanceDB";
6     private static final String USER = "root";
7     private static final String PASSWORD = "tharunH@2005";
8
9     public static Connection getConnection() throws SQLException {
10         return DriverManager.getConnection(URL, USER, PASSWORD);
11     }
12 }
13
14 }
```

```
1 package model;
2
3 public class Student {
4     private int id;
5     private String name;
6     private String email;
7
8     public Student(int id, String name, String email) {
9         this.id = id;
10        this.name = name;
11        this.email = email;
12    }
13
14    public int getId() {
15        return id;
16    }
17
18    public String getName() {
19        return name;
20    }
21
22    public String getEmail() {
23        return email;
24    }
25 }
26
```

```
1 package model;
2
3 import java.sql.Date;
4
5 public class Attendance {
6     private int id;
7     private int studentId;
8     private Date date;
9     private String status;
10
11    public Attendance(int id, int studentId, Date date, String status) {
12        this.id = id;
13        this.studentId = studentId;
14        this.date = date;
15        this.status = status;
16    }
17
18    public int getId() {
19        return id;
20    }
21
22    public int getStudentId() {
23        return studentId;
24    }
25
26    public Date getDate() {
27        return date;
28    }
29
30    public String getStatus() {
31        return status;
32    }
33 }
34
```

```
1 package model;
2
3 public class Performance {
4     private int id;
5     private int studentId;
6     private String subject;
7     private String grade;
8
9     public Performance(int id, int studentId, String subject, String grade) {
10         this.id = id;
11         this.studentId = studentId;
12         this.subject = subject;
13         this.grade = grade;
14     }
15
16     public int getId() {
17         return id;
18     }
19
20     public int getStudentId() {
21         return studentId;
22     }
23
24     public String getSubject() {
25         return subject;
26     }
27
28     public String getGrade() {
29         return grade;
30     }
31 }
32 }
```

```

1 package dao;
2
3 import model.Student;
4
5
6 public class StudentDAO {
7     public void addStudent(Student student) {
8         String query = "INSERT INTO Students (name, email) VALUES (?, ?)";
9         try (Connection conn = DatabaseConnection.getConnection();
10             PreparedStatement stmt = conn.prepareStatement(query)) {
11             stmt.setString(1, student.getName());
12             stmt.setString(2, student.getEmail());
13             stmt.executeUpdate();
14         } catch (SQLException e) {
15             e.printStackTrace();
16         }
17     }
18 }
19
20 // StudentDAO.java
21 public Student getStudentById(int studentId) {
22     Student student = null;
23     try (Connection conn = DatabaseConnection.getConnection();
24         PreparedStatement stmt = conn.prepareStatement("SELECT * FROM students WHERE id = ?")) {
25         stmt.setInt(1, studentId);
26         ResultSet rs = stmt.executeQuery();
27         if (rs.next()) {
28             student = new Student(rs.getInt("id"), rs.getString("name"), rs.getString("email"));
29         }
30     } catch (SQLException e) {
31         e.printStackTrace();
32     }
33     return student;
34 }
35
36
37
38 public List<Student> getAllStudents() {
39     List<Student> students = new ArrayList<>();
40     String query = "SELECT * FROM Students";
41     try (Connection conn = DatabaseConnection.getConnection();
42         Statement stmt = conn.createStatement();
43         ResultSet rs = stmt.executeQuery(query)) {
44         while (rs.next()) {
45             Student student = new Student(rs.getInt("id"), rs.getString("name"), rs.getString("email"));
46             students.add(student);
47         }
48     } catch (SQLException e) {
49         e.printStackTrace();
50     }
51     return students;
52 }
53
54

```

```

37
38 public List<Student> getAllStudents() {
39     List<Student> students = new ArrayList<>();
40     String query = "SELECT * FROM Students";
41     try (Connection conn = DatabaseConnection.getConnection();
42         Statement stmt = conn.createStatement();
43         ResultSet rs = stmt.executeQuery(query)) {
44         while (rs.next()) {
45             Student student = new Student(rs.getInt("id"), rs.getString("name"), rs.getString("email"));
46             students.add(student);
47         }
48     } catch (SQLException e) {
49         e.printStackTrace();
50     }
51     return students;
52 }
53
54

```

```
1 package PerformanceAndAttendanceSystem/src/util/DatabaseConnection.java
2
3 import dao.StudentDAO;
4
5 public class PerformanceAndAttendanceUI {
6     private JFrame frame;
7     private JTextField nameField, emailField, studentIdField, subjectField, gradeField;
8     private JComboBox<String> statusComboBox;
9     private JTextArea displayArea;
10    private StudentDAO studentDAO;
11    private AttendanceDAO attendanceDAO;
12    private PerformanceDAO performanceDAO;
13
14    public PerformanceAndAttendanceUI() {
15        studentDAO = new StudentDAO();
16        attendanceDAO = new AttendanceDAO();
17        performanceDAO = new PerformanceDAO();
18        initializeUI();
19    }
20
21    private void initializeUI() {
22        frame = new JFrame("Performance and Attendance System");
23        frame.setSize(500, 700);
24        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
25        frame.setLayout(new FlowLayout());
26
27        frame.add(new JLabel("Student Name:"));
28        nameField = new JTextField(20);
29        frame.add(nameField);
30
31        frame.add(new JLabel("Student Email:"));
32        emailField = new JTextField(20);
33        frame.add(emailField);
34
35        JButton addStudentButton = new JButton("Add Student");
36        addStudentButton.addActionListener(e -> addStudent());
37        frame.add(addStudentButton);
38    }
39}
```

```
84•    private void addStudent() {
85        String name = nameField.getText();
86        String email = emailField.getText();
87        if (!name.isEmpty() && !email.isEmpty()) {
88            studentDAO.addStudent(new Student(0, name, email));
89            JOptionPane.showMessageDialog(frame, "Student added successfully!");
90            nameField.setText("");
91            emailField.setText("");
92        } else {
93            JOptionPane.showMessageDialog(frame, "Please enter all student details.");
94        }
95    }
96
97•    private void addAttendance() {
98        try {
99            int studentId = Integer.parseInt(studentIdField.getText());
100           String status = (String) statusComboBox.getSelectedItem();
101           Attendance attendance = new Attendance(0, studentId, new Date(System.currentTimeMillis()), status);
102           attendanceDAO.addAttendance(attendance);
103           JOptionPane.showMessageDialog(frame, "Attendance added successfully!");
104        } catch (NumberFormatException e) {
105            JOptionPane.showMessageDialog(frame, "Please enter a valid student ID.");
106        }
107    }
108
109•    private void addPerformance() {
110        try {
111            int studentId = Integer.parseInt(studentIdField.getText());
112            String subject = subjectField.getText();
113            String grade = gradeField.getText();
114            if (!subject.isEmpty() && !grade.isEmpty()) {
115                Performance performance = new Performance(0, studentId, subject, grade);
116                performanceDAO.addPerformance(performance);
117                JOptionPane.showMessageDialog(frame, "Performance added successfully!");
118                subjectField.setText("");
119                gradeField.setText("");
120            } else {
121            
```

```

127
128● private void displayStudentRecords() {
129     try {
130         int studentId = Integer.parseInt(studentIdField.getText());
131
132         // Fetch student details
133         Student student = studentDAO.getStudentById(studentId);
134         if (student == null) {
135             JOptionPane.showMessageDialog(frame, "No student found with ID: " + studentId);
136             return;
137         }
138
139         // Display student details
140         displayArea.setText("Student Details:\n");
141         displayArea.append("ID: " + student.getId() + "\n");
142         displayArea.append("Name: " + student.getName() + "\n");
143         displayArea.append("Email: " + student.getEmail() + "\n\n");
144
145         // Fetch and display attendance records
146         List<Attendance> attendanceList = attendanceDAO.getAttendanceByStudentId(studentId);
147         displayArea.append("Attendance Records:\n");
148         for (Attendance attendance : attendanceList) {
149             displayArea.append("Date: " + attendance.getDate() + ", Status: " + attendance.getStatus() + "\n");
150         }
151
152         // Fetch and display performance records
153         displayArea.append("\nPerformance Records:\n");
154         List<Performance> performanceList = performanceDAO.getPerformanceByStudentId(studentId);
155         for (Performance performance : performanceList) {
156             displayArea.append("Subject: " + performance.getSubject() + '|', Grade: " + performance.getGrade() + "\n");
157         }
158
159     } catch (NumberFormatException e) {
160         JOptionPane.showMessageDialog(frame, "Please enter a valid student ID.");
161     }
162

```

```

1 package dao;
2
3● import model.Attendance;
4
5
6 public class AttendanceDAO {
7
8     public void addAttendance(Attendance attendance) {
9         String query = "INSERT INTO Attendance (student_id, date, status) VALUES (?, ?, ?)";
10    try (Connection conn = DatabaseConnection.getConnection();
11        PreparedStatement stmt = conn.prepareStatement(query)) {
12        stmt.setInt(1, attendance.getStudentId());
13        stmt.setDate(2, attendance.getDate());
14        stmt.setString(3, attendance.getStatus());
15        stmt.executeUpdate();
16    } catch (SQLException e) {
17        e.printStackTrace();
18    }
19
20    }
21
22
23● public List<Attendance> getAttendanceByStudentId(int studentId) {
24    List<Attendance> attendanceList = new ArrayList<>();
25    String query = "SELECT * FROM Attendance WHERE student_id = ?";
26    try (Connection conn = DatabaseConnection.getConnection();
27        PreparedStatement stmt = conn.prepareStatement(query)) {
28        stmt.setInt(1, studentId);
29        ResultSet rs = stmt.executeQuery();
30        while (rs.next()) {
31            Attendance attendance = new Attendance(rs.getInt("id"), rs.getInt("student_id"),
32                rs.getDate("date"), rs.getString("status"));
33            attendanceList.add(attendance);
34        }
35    } catch (SQLException e) {
36        e.printStackTrace();
37    }
38    return attendanceList;
39}
40
41

```

```

1 Student.java   Attendance.java  Performance...  StudentDAO.java  PerformanceA...  AttendanceDA...  StudentManag... × »
10 import javax.swing.*;□
5
6 public class StudentManagementSystem {
7
8     private JFrame frame;
9     private JTextField nameField, rollNumberField, marksField;
10    private JButton addButton, deleteButton;
11    private JTable table;
12    private DefaultTableModel tableModel;
13
14    public StudentManagementSystem() {
15        frame = new JFrame("Student Management System");
16        frame.setLayout(new FlowLayout());
17        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
18        frame.setSize(500, 400);
19
20        // Labels and text fields for input
21        frame.add(new JLabel("Name:"));
22        nameField = new JTextField(15);
23        frame.add(nameField);
24
25        frame.add(new JLabel("Roll Number:"));
26        rollNumberField = new JTextField(15);
27        frame.add(rollNumberField);
28
29        frame.add(new JLabel("Marks:"));
30        marksField = new JTextField(15);
31        frame.add(marksField);
32
33        // Button to add student
34        addButton = new JButton("Add Student");
35        frame.add(addButton);
36
37        // Table to display student details
38        tableModel = new DefaultTableModel(new String[]{"Name", "Roll Number", "Marks"}, 0);
39        table = new JTable(tableModel);
40        table.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
41        frame.add(new JScrollPane(table));

```

```

50
37        // Table to display student details
38        tableModel = new DefaultTableModel(new String[]{"Name", "Roll Number", "Marks"}, 0);
39        table = new JTable(tableModel);
40        table.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
41        frame.add(new JScrollPane(table));
42
43        // Button to delete selected student
44        deleteButton = new JButton("Delete Student");
45        frame.add(deleteButton);
46
47        // Action listener to add student
48        addButton.addActionListener(new ActionListener() {
49            public void actionPerformed(ActionEvent e) {
50                String name = nameField.getText();
51                String rollNumber = rollNumberField.getText();
52                String marks = marksField.getText();
53
54                if (!name.isEmpty() && !rollNumber.isEmpty() && !marks.isEmpty()) {
55                    // Add the student to the table
56                    tableModel.addRow(new Object[]{name, rollNumber, marks});
57                    // Clear input fields
58                    nameField.setText("");
59                    rollNumberField.setText("");
60                    marksField.setText("");
61                } else {
62                    JOptionPane.showMessageDialog(frame, "Please fill in all fields", "Error", JOptionPane.ERROR_MESSAGE);
63                }
64            }
65        });
66
67        // Action listener to delete selected student
68        deleteButton.addActionListener(new ActionListener() {
69            public void actionPerformed(ActionEvent e) {
70                int selectedRow = table.getSelectedRow();
71                if (selectedRow != -1) {

```

```

55
56
57
58
59
60
61
62
63
64
65
66
67     // Action listener to delete selected student
68     deleteButton.addActionListener(new ActionListener() {
69         public void actionPerformed(ActionEvent e) {
70             int selectedRow = table.getSelectedRow();
71             if (selectedRow != -1) {
72                 // Remove the selected row from the table
73                 tableView.removeRow(selectedRow);
74             } else {
75                 JOptionPane.showMessageDialog(frame, "Please select a student to delete", "Error", JOptionPane.ERROR_MESSAGE);
76             }
77         }
78     });
79
80     // Make the frame visible
81     frame.setVisible(true);
82 }
83
84 public static void main(String[] args) {
85     new StudentManagementSystem();
86 }
87 }
88

```

```

Student.java  Performance...  StudentDAO.java  PerformanceA...  AttendanceDA...  PerformanceD...  Studentmanag...
1 package dao;
2
3 import model.Performance;
4
5 public class PerformanceDAO {
6     public void addPerformance(Performance performance) {
7         String query = "INSERT INTO Performance (student_id, subject, grade) VALUES (?, ?, ?)";
8         try (Connection conn = DatabaseConnection.getConnection();
9              PreparedStatement stmt = conn.prepareStatement(query)) {
10            stmt.setInt(1, performance.getStudentId());
11            stmt.setString(2, performance.getSubject());
12            stmt.setString(3, performance.getGrade());
13            stmt.executeUpdate();
14        } catch (SQLException e) {
15            e.printStackTrace();
16        }
17    }
18
19    public List<Performance> getPerformanceByStudentId(int studentId) {
20        List<Performance> performanceList = new ArrayList<>();
21        String query = "SELECT * FROM Performance WHERE student_id = ?";
22        try (Connection conn = DatabaseConnection.getConnection();
23              PreparedStatement stmt = conn.prepareStatement(query)) {
24            stmt.setInt(1, studentId);
25            ResultSet rs = stmt.executeQuery();
26            while (rs.next()) {
27                Performance performance = new Performance(rs.getInt("id"), rs.getInt("student_id"),
28                      rs.getString("subject"), rs.getString("grade"));
29                performanceList.add(performance);
30            }
31        } catch (SQLException e) {
32            e.printStackTrace();
33        }
34    }
35    return performanceList;
36 }
37 }
38 }
39 }
40 }
41

```