Status	Finished
Started	Tuesday, 1 October 2024, 8:23 AM
Completed	Tuesday, 1 October 2024, 8:52 AM
Duration	28 mins 44 secs

Question **1**

Correct

Marked out of 5.00

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

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

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

For example:

Input	Result
123	2
456	1

Answer: (penalty regime: 0 %)

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

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

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

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

The last digit should be returned as a positive number.

For example,

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

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

For example:

Input	Result
197	7
-197	7

Answer: (penalty regime: 0 %)

```java
import java.util.Scanner;
public class last{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int a=sc.nextInt();

        if(a<0){
           a=-1*a;
        }
        System.out.println(a%10);
    }
}
```

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

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

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

For example,

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

Below is the explanation:

Last digit of the 267 is 7

Last digit of the 154 is 4

Sum of 7 and 4 = 11

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

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

i.e.

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

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

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

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

For example:

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

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class sums
{
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        int a=sc.nextInt();
        int b=sc.nextInt();
        if(a<0)
        {
            a=-1*a;
        }
        if(b<0)
        {
            b=-1*b;
        }
        System.out.println((a%10)+(b%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? ►

Status	Finished
Started	Tuesday, 1 October 2024, 9:11 AM
Completed	Saturday, 5 October 2024, 1:01 PM
Duration	4 days 3 hours

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 %)

Reset answer

```
// Java program to count trailing 0s in n!
import java.io.*;
import java.util.Scanner;

class prog {
    // Function to return trailing 0s in factorial of n
    static int findTrailingZeros(int n) {
        if (n < 0) // Negative Number Edge Case
            return -1;

        // Initialize result
        int count = 0;

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

        return count;
    }

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

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

```

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

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Write a Java program to input a number from user and print it into words using for loop. How to display number in words using loop in Java programming.

Logic to print number in words in Java programming.

Example

Input

1234

Output

One Two Three Four

Input:

16

Output:

one six

For example:

Test	Input	Result
1	45	Four Five
2	13	One Three
3	87	Eight Seven

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class num{
    public static void main(String[] args){
        Scanner s=new Scanner(System.in);
        String input=s.nextLine();
        String[] w={"Zero","One","Two","Three","Four","Five","Six","Seven","Eight","Nine"};
        for(int i=0;i<input.length();i++){
            int d=Character.digit(input.charAt(i),10);
            System.out.print(w[d]+" ");
        }

    }
}
```

	Test	Input	Expected	Got	
✓	1	45	Four Five	Four Five	✓
✓	2	13	One Three	One Three	✓
✓	3	87	Eight Seven	Eight Seven	✓

Passed all tests! ✓

Question **3**

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 %)

```java
import java.util.Scanner;

public class Sequence {

    public static int findNthTerm(int n) {
        if (n == 1) {
            return 0;
        } else if (n == 2 || n == 3) {
            return 1;
        }

        int term1 = 0, term2 = 1, term3 = 1;

        for (int i = 4; i <= n; i++) {
            int nextTerm = term1 + term2 + term3;
            term1 = term2;
            term2 = term3;
            term3 = nextTerm;
        }

        return term3;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.println();
        int n = scanner.nextInt();


        int result = findNthTerm(n);
        System.out.println(result);

        scanner.close();
```

```
    }
}

```

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

Passed all tests! ✓

◄ Lab-02-MCQ

Jump to...

Lab-03-MCQ ►

Status	Finished
Started	Monday, 7 October 2024, 4:24 PM
Completed	Monday, 7 October 2024, 4:49 PM
Duration	25 mins 8 secs

Question **1**

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 x 9), (-4 x 9), (3 x 9), (0 x 9)} = {-72, -36, -27, 0}

So, the expected output is the resultant array {-72, -36, -27, 0}.

Example 2:

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

input2 = {10, 87, 63, 42, 2}

Expected Output = {-6699, 0, -2088, -3915, -7395}

Explanation:

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

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

{(10 - 87), (87 - 87), (63 - 87), (42 - 87), (2 - 87)} = {-77, 0, -24, -45, -85}

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

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

So, the expected output is the resultant array {-6699, 0, -2088, -3915, -7395}.

Example 3:

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

input2 = {-9, 9}

Expected Output = {-162, 0}

Explanation:

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

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

{(-9 - 9), (9 - 9)} = {-18, 0}

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

{(-18 x 9), (0 x 9)} = {-162, 0}

So, the expected output is the resultant array {-162, 0}.

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

For example:

Input	Result
4 1 5 6 9	-72 -36 -27 0

Input	Result
5 10 87 63 42 2	-6699 0 -2088 -3915 -7395
2 -9 9	-162 0

Answer: (penalty regime: 0 %)

```
import java.util.*;
public class Main{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int arr[]=new int[n];
        for(int i=0;i<n;i++){
            arr[i]=sc.nextInt();
        }
        int max=arr[0];
        for(int i=1;i<n;i++){
            if(max<arr[i]){
                max=arr[i];
            }
        }
        int[] res=new int[n];
        for(int i=0;i<n;i++){
            res[i]=arr[i]-max;
        }
        for(int i=0;i<n;i++){
            res[i]=res[i]*max;
        }
        for(int val:res){
            System.out.print(val+" ");
        }
    }
}
```

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

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

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

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

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

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

input1 represents the number of elements in the array.

input2 represents the array of integers.

Example 1:

input1 = 16

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

Expected output = 62

Explanation:

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

Example 2:

input1 = 11

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

Expected output = -1

Explanation:

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

Example 3:

input1 = 16

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

Expected output = 174

Explanation:

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

For example:

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

Answer: (penalty regime: 0 %)

```
import java.util.*;
public class Main{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int arr[]=new int[n];
        for(int i=0;i<n;i++){
            arr[i]=sc.nextInt();
        }
```

```
        int maxLen=0,curLen=0,curSum=0,longSum=0;
        for(int i=0;i<n;i++){
            if(arr[i]>=0){
                curLen++;
                curSum+=arr[i];
            }else{
                if(curLen>maxLen){
                    maxLen=curLen;
                    longSum=curSum;
                }else if(curLen==maxLen){
                    longSum+=curSum;
                }
                curLen=0;
                curSum=0;
            }
        }
        if(curLen>maxLen){
            longSum=curSum;
        }else if(curLen==maxLen){
            longSum+=curSum;
        }
        if(maxLen==0){
            System.out.println(-1);
        }else{
            System.out.println(longSum);
        }
    }
}
```

	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

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 %)

```
import java.util.*;
public class Main{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
```

```java
        int arr[]=new int[n];
        for(int i=0;i<n;i++){
            arr[i]=sc.nextInt();
        }
        int pos[]=new int[n];
        for(int i=0;i<n;i++){
            if(i<arr.length){
                int num=arr[i];
                int temp=num;
                for(int j=0;j<i;j++){
                    temp/=10;
                }
                pos[i]=temp;
            }else{
                pos[i]=0;
            }
        }
        int sum=0;
        for(int val:pos){
            sum+=val*val;
        }
        System.out.println(sum);
    }
}
```

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

Passed all tests! ✓

◄ Lab-03-MCQ

Jump to...

Simple Encoded Array ►

Status	Finished
Started	Monday, 7 October 2024, 4:52 PM
Completed	Monday, 7 October 2024, 5:21 PM
Duration	28 mins 22 secs

Question **1**

Correct

Marked out of 5.00

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

Student()

Student(String name)

Student(String name, int rollno)

Input:

No input

Output:

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

For example:

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

Answer: (penalty regime: 0 %)

```java
import java.util.*;
class Student{
    private String name;
    private int rollNo;
    public Student(){
        System.out.println("No-arg constructor is invoked");
        this.name=null;
        this.rollNo=0;
    }
    public Student(String name){
        System.out.println("1 arg constructor is invoked");
        this.name=name;
        this.rollNo=0;
    }
    public Student(String name,int rollNo){
        System.out.println("2 arg constructor is invoked");
        this.name=name;
        this.rollNo=rollNo;
    }
    public void display(){
        System.out.println("Name ="+(name!=null?name:"null")+" , Roll no = "+rollNo);
    }
}
public class Main{
    public static void main(String[] args){
        Student stu1=new Student();
        Student stu2=new Student("Rajalakshmi");
        Student stu3=new Student("Lakshmi",101);
        stu1.display();
        stu2.display();
        stu3.display();
    }
}
```

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

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Create a Class Mobile with the attributes listed below,

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

Define a Parameterized constructor to initialize the above instance variables.

Define getter and setter methods for the attributes above.

for example : setter method for manufacturer is

void setManufacturer(String manufacturer){

this.manufacturer= manufacturer;

}

String getManufacturer(){

return manufacturer;}

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

For example:

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

Answer: (penalty regime: 0 %)

```
class Mobile{
    private String m;
    private String os;
    public String c;
    private int cost;

    public Mobile(String m,String os,String c,int cost){
        this.m=m;
        this.os=os;
        this.c=c;
        this.cost=cost;
    }
    public void setManufacturer(String m){
        this.m=m;
    }
    public void setOperatingSystem(String os){
        this.os=os;
    }
    public void setColor(String color){
        this.c=c;
    }
    public void setCost(int cost){
        this.cost=cost;
    }

    public String getManufacturer(){
        return m;
    }
    public String getOperatingSystem(){
        return os;
    }
    public String getColor(){
        return c;
    }
    public int getCost(){
        return cost;
    }

    @Override
```

```
    public String toString(){
        return "manufacturer = "+ m +"\n" +
        "operating_system = " + os + "\n" +
        "color = " + c + "\n" + "cost = " + cost;
    }
}
public class Prog{
    public static void main(String[] args){
        Mobile mobile=new Mobile("Redmi","Andriod","Blue",34000);
        System.out.println(mobile);
    }
}
```

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

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

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

Area of Circle = πr^2

Circumference = $2\pi r$

Input:

2

Output:

Area = 12.57
Circumference = 12.57

For example:

Test	Input	Result
1	4	Area = 50.27 Circumference = 25.13

Answer: (penalty regime: 0 %)

Reset answer

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


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


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


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

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

    }
}

```

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

Passed all tests! ✓

◄ Lab-04-MCQ

Jump to...

Number of Primes in a specified range ►

Status	Finished
Started	Monday, 7 October 2024, 5:22 PM
Completed	Monday, 7 October 2024, 5:50 PM
Duration	28 mins 10 secs

Question **1**

Correct

Marked out of 5.00

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

College:

String collegeName;

public College() { }

public admitted() { }

Student:

String studentName;

String department;

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

public toString()

Expected Output:

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

For example:

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

Answer: (penalty regime: 0 %)

Reset answer

```
class College
{
protected   String collegeName;

public College(String collegeName) {
    this.collegeName=collegeName;

    }

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

String studentName;
String department;

public Student(String collegeName, String studentName,String depart) {
   super(collegeName);
   this.studentName=studentName;;
   this.department=depart;
}

public String toString(){
    return "CollegeName : "+collegeName+"\n"+
    "StudentName : "+studentName+"\n"+
    "Department : "+department;
}
}
public class Main {
public static void main (String[] args) {
        Student  s1 = new Student("REC","Venkatesh","CSE");
        s1.admitted();
```

```
        System.out.println(s1.toString());
}
}
```

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

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

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

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

For example:

Result

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

Answer: (penalty regime: 0 %)

Reset answer

```java
class ba{
    int bal;
    ba(int b){
        this.bal=b;
    }
    void deposit(int a){
        bal+=a;
    }
    void withdraw(int a){
        bal-=a;
    }
    int gb(){
        return bal;
    }
}
class sa extends ba{
    sa(int b){
        super(b);
    }
    void withdraw(int a){
        if((bal-a)<100){
            System.out.println("Minimum balance of $100 required!");
        }
        else{
            bal-=a;
        }
    }
}
public class hello{
    public static void main(String[] args){
        ba BA1234=new ba(500);
        sa SA1000=new sa(300);
        System.out.println("Create a Bank Account object (A/c No. BA1234) with initial balance of $500:");
        System.out.println("Deposit $1000 into account BA1234:");
        BA1234.deposit(1000);
        System.out.println("New balance after depositing $1000: $"+BA1234.gb()+".0");
        System.out.println("Withdraw $600 from account BA1234:");
        BA1234.withdraw(600);
        System.out.println("New balance after withdrawing $600: $"+BA1234.gb()+".0");
        System.out.println("Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300:");
        System.out.println("Try to withdraw $250 from SA1000!");
        SA1000.withdraw(250);
        System.out.println("Balance after trying to withdraw $250: $"+SA1000.gb()+".0");
    }
}
```

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

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

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

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

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

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

class Mobile{

}

class CameraMobile extends Mobile {

}

class AndroidMobile extends CameraMobile {

}

expected output:

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

For example:

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

Answer: (penalty regime: 0 %)

```
class m{
    m(){
        System.out.println("Basic Mobile is Manufactured");
    }
}
class cm{
    void nf(){
        System.out.println("Camera Mobile with 5MG px");
    }
    cm(){
        System.out.println("Camera Mobile is Manufactured");
    }
}
class am{
    void an(){
        System.out.println("Touch Screen Mobile is Manufactured");
    }
    am(){
        System.out.println("Android Mobile is Manufactured");
    }
}
public class hello{
    public static void main(String [] args){
        m mm=new m();
        cm c=new cm();
        am a=new am();
        c.nf();
        a.an();
    }
}
```

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

Passed all tests! ✓

◄ Lab-05-MCQ

Jump to...

Is Palindrome Number? ►

Status	Finished
Started	Monday, 7 October 2024, 5:51 PM
Completed	Monday, 7 October 2024, 6:18 PM
Duration	27 mins 37 secs

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 %)

```java
import java.util.*;
public class hello{
    public static void main(String [] args){
        Scanner sc=new Scanner(System.in);
        String s=sc.next();
        String st="";
        String stri="abcdefghijklmnopqrstuvwxyz";
        for(int i=0;i<s.length()-1;i++){
            int a=s.charAt(i);
            int b=s.charAt(i+1);
            if(a==58 || b==58){
                continue;
            }
            if(a==b){
                char str=s.charAt(i);
                st+=str;
            }
            else{
                int c=a-b;
                char d=stri.charAt(c-1);
                st+=d;
            }
        }
        System.out.println(st.toUpperCase());
    }
}
```

	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 %)

```
import java.util.*;
public class hello{
    public static void main(String [] args){
        Scanner sc=new Scanner(System.in);
        String stc=sc.nextLine();
        String[] str=stc.split(" ");
        int a=sc.nextInt();
        int b=a%10;
        int c=a/10;
        String st="";
```

```
        int len=str[c-1].length();
        int mid=len/2;
        if(len%2==0)
        {
        for(int i=mid-1;i>=0;i--){
            st+=str[c-1].charAt(i);
        }
        }
        else
        {
            for(int i=mid;i>=0;i--){
                st+=str[c-1].charAt(i);
            }
        }
        for(int i=mid;i<str[c-1].length();i++)
        {
            st+=str[c-1].charAt(i);
        }
        int len1=str[b-1].length();
        int m1=len1/2;
        st+=" ";
        if(len1%2==0){
            for(int i=m1-1;i>=0;i--){
                st+=str[b-1].charAt(i);
            }
        }
        else
        {
            for(int i=m1;i>=0;i--){
                st+=str[b-1].charAt(i);
            }
        }
        for(int i=m1;i<str[b-1].length();i++){
            st+=str[b-1].charAt(i);
        }
        System.out.println(st);
    }
}
```

	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 %)

```
import java.util.Scanner;
import java.util.Set;
import java.util.TreeSet;

public class ConcatenateAndSort {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String input1 = scanner.nextLine();
        String input2 = scanner.nextLine();

        if (input1.trim().isEmpty() && input2.trim().isEmpty()) {
            System.out.println("null");
        } else {
            String combined = input1 + input2;
            Set<Character> charSet = new TreeSet<>((a, b) -> Character.compare(b, a));
            for (char c : combined.toCharArray()) {
                if (c != ' ') {
                    charSet.add(c);
                }
            }
            StringBuilder result = new StringBuilder();
            for (char c : charSet) {
                result.append(c);
            }
            System.out.println(result.toString());
        }
        scanner.close();
```

```
    }
}
```

	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 ►

Status	Finished
Started	Monday, 7 October 2024, 6:19 PM
Completed	Monday, 7 October 2024, 6:40 PM
Duration	21 mins 6 secs

Question **1**

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

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

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

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

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

```
        String hname;
        Scanner sc= new Scanner(System.in);
       hname=sc.next();
        String  vteam=sc.next();
        int htpoints=sc.nextInt();
        int vtpoints=sc.nextInt();
   College s= new College();
    s.setHomeTeam(hname);
    s.setVisitingTeam(vteam);
    s.homeTeamScored(htpoints);
    s.visitingTeamScored(vtpoints);
```

	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! ✓

Question **2**

Correct

Marked out of 5.00

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

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

RBI interface has two more methods default and static method.

default void policyNote() {

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

}

static void regulations(){

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

}

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

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

Sample Input/Output:

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

For example:

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

Answer: (penalty regime: 0 %)

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

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

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

create 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:

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

For example:

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

Answer: (penalty regime: 0 %)

```
import java.util.*;
interface p{
    void pl();
}
class f implements p{
    String name;
    public f(String n){
        this.name=n;
    }
    public void pl(){
        System.out.println(name+" is Playing football");
    }
}
class v implements p{
    String name;
    public v(String n){
        this.name=n;
    }
    public void pl(){
        System.out.println(name+" is Playing volleyball");
    }
}
class b implements p{
    String name;
    public b(String n){
        this.name=n;
    }
    public void pl(){
        System.out.println(name+" is Playing basketball");
    }
}
public class hello{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        f f1=new f(sc.next());
        v v1=new v(sc.next());
        b b1=new b(sc.next());
        f1.pl();
        v1.pl();
```

```
        b1.pl();
    }
}
```

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

Passed all tests! ✓

◄ Lab-07-MCQ

Jump to...

Generate series and find Nth element ►

Status	Finished
Started	Monday, 7 October 2024, 6:41 PM
Completed	Monday, 7 October 2024, 7:09 PM
Duration	27 mins 58 secs

Question **1**

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 %)

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

    }
}
```

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

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

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

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

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

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

sample Input :

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

5 // length of the rectangle

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

4 // base of the triangle

3 // height of the triangle

OUTPUT:

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

For example:

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

Answer: (penalty regime: 0 %)

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

```
    r(double l,double b){
        this.l=l;
        this.b=b;
    }
    public double calculateArea(){
        double a=l*b;
        System.out.printf("Area of a Rectangle: %.2f\n",a);
        return a;
    }
}
class t extends s{
    double b;
    double h;
    t(double b,double h){
        this.b=b;
        this.h=h;
    }
    public double calculateArea(){
        double a=b*h*0.5;
        System.out.printf("Area of a Triangle: %.2f\n",a);
        return a;
    }
}
public class hello{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        double r1=sc.nextDouble();
        c c1=new c(r1);
        double l1=sc.nextDouble();
        double b1=sc.nextDouble();
        r r2=new r(l1,b1);
        double b2=sc.nextDouble();
        double h2=sc.nextDouble();
        t t1=new t(b2,h2);
```

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

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

1. Final Variable:

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

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

2. Final Method:

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

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

3. Final Class:

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

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

you should delete any piece of code.

For example:

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

Answer: (penalty regime: 0 %)

Reset answer

```
class FinalExample {

    // Final variable
                int maxSpeed = 120;

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

class SubClass extends FinalExample {


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

class prog {
    public static void main(String[] args) {
        FinalExample obj = new FinalExample();
        obj.displayMaxSpeed();

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

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

Passed all tests! ✓

◄ Lab-08-MCQ

Jump to...

FindStringCode ►

Status	Finished
Started	Monday, 21 October 2024, 10:37 AM
Completed	Monday, 21 October 2024, 11:18 AM
Duration	41 mins 38 secs

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

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

        n = 37;
        trynumber(n);
    }
    public static void trynumber(int n) {
        try {
            checkEvenNumber(n);
            System.out.println(n + " is even.");
        }
        catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }

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

```

	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

Write a Java program to handle ArithmeticException and ArrayIndexOutOfBoundsException.

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

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

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

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

Input:

5

10 0 20 30 40

Output:

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

Input:

3
10 20 30

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

For example:

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

Answer: (penalty regime: 0 %)

```java
import java.util.Scanner;
public class ExceptionHandlingDemo {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int size = sc.nextInt();
        int[] array = new int[size];
        for (int i = 0; i < size; i++) {
            array[i] = sc.nextInt();
        }
        try {
            int result = array[0] / array[1];
            System.out.println("Element at index " + size + ": " + array[size]);
        }
        catch (ArithmeticException e) {
            System.out.println("java.lang.ArithmeticException: / by zero");
        }
        catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("java.lang.ArrayIndexOutOfBoundsException: " + e.getMessage());
        }
        finally {
            System.out.println("I am always executed");
        }
    }
}
```

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

	Test	Input	Expected	Got	
✓	2	3 10 20 30	java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3 I am always executed	java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3 I am always executed	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

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

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

Sample Input:

3
5 2 1

Sample Output:

8

Sample Input:

2

1 g

Sample Output:

```
You entered bad data.
```

For example:

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

Answer: (penalty regime: 0 %)

Reset answer

```java
import java.util.Scanner;
import java.util.InputMismatchException;
class prog {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int length = sc.nextInt();
        int[] name = new int[length];
        int sum = 0;
        try {
            for (int i = 0; i < length; i++) {
                name[i] = sc.nextInt();
                sum += name[i];
            }
            System.out.println(sum);
        } catch (InputMismatchException e) {
            System.out.println("You entered bad data.");
        }
    }
}

```

	Input	Expected	Got	
✓	3 5 2 1	8	8	✓

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

Passed all tests! ✓

◄ Lab-09-MCQ

Jump to...

The “Nambiar Number” Generator ►

Status	Finished
Started	Wednesday, 6 November 2024, 12:09 PM
Completed	Wednesday, 6 November 2024, 12:40 PM
Duration	30 mins 31 secs

Question **1**

Correct

Marked out of 1.00

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

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

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

Approach:

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

Answer: (penalty regime: 0 %)

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

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

Passed all tests! ✓

Question **2**

Correct

Marked out of 1.00

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

```
import java.util.ArrayList;
import java.util.Scanner;
public class Prog {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        ArrayList<Integer> list = new ArrayList<Integer>();
        for (int i = 0; i < n; i++)
            list.add(sc.nextInt());
        System.out.println("ArrayList: " + list);

        list.set(1, 100);

        System.out.println("Index of 100 = " + list.indexOf(100));

        System.out.println("LastIndex of 100 = " + list.lastIndexOf(100));

        System.out.println(list.contains(200)); // Output: false

        System.out.println("Size Of ArrayList = " + list.size());

        list.add(1, 500);
        list.remove(3);
        System.out.print("ArrayList: " + list);
    }
}
```

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

Passed all tests! ✓

Question **3**

Correct

Marked out of 1.00

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 %)

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

```

	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 ►

Status	Finished
Started	Sunday, 17 November 2024, 12:02 PM
Completed	Sunday, 17 November 2024, 12:35 PM
Duration	32 mins 57 secs

Question **1**

Correct

Marked out of 1.00

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

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

This class permits the null element.

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

Java HashSet Features

A few important features of HashSet are mentioned below:

- Implements Set Interface.
- The underlying data structure for HashSet is Hashtable.
- As it implements the Set Interface, duplicate values are not allowed.
- Objects that you insert in HashSet are not guaranteed to be inserted in the same order. Objects are inserted based on their hash code.
- NULL elements are allowed in HashSet.
- HashSet also implements **Serializable** and **Cloneable** interfaces.
-
```
public class HashSet<E> extends AbstractSet<E> implements Set<E>, Cloneable, Serializable
Sample Input and Output:
5
90
56
45
78
25
78
Sample Output:
78 was found in the set.
Sample Input and output:
3
2
7
9
5
Sample Input and output:
5 was not found in the set.
```

Answer: (penalty regime: 0 %)

Reset answer

```java
import java.util.HashSet;
import java.util.Scanner;
public class Prog {
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int n = sc.nextInt();
    HashSet<Integer> numbers = new HashSet<>();
    for (int i = 0; i < n; i++) {
      numbers.add(sc.nextInt());
    }
    int skey = sc.nextInt();
    if (numbers.contains(skey)) {
      System.out.println(skey + " was found in the set.");
    } else {
      System.out.println(skey + " was not found in the set.");
    }
    sc.close();
  }
}

```

	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 %)

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

```

	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

```java
import java.util.HashMap;
import java.util.Map.Entry;
import java.util.Set;
import java.util.Scanner;
public class Prog {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<String, Integer>();
        String name;
        int num;
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        sc.nextLine();
        for (int i = 0; i < n; i++) {
            name = sc.nextLine();
            num = sc.nextInt();
            map.put(name, num);
            sc.nextLine();
        }
        Set<Entry<String, Integer>> entrySet = map.entrySet();
        for (Entry<String, Integer> entry : entrySet) {
            System.out.println(entry.getKey() + " : " + entry.getValue());
        }
        System.out.println("----------");
        HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();
        anotherMap.put("SIX", 6);
        anotherMap.put("SEVEN", 7);
        anotherMap.putAll(map);
        entrySet = anotherMap.entrySet();
        for (Entry<String, Integer> entry : entrySet) {
            System.out.println(entry.getKey() + " : " + entry.getValue());
        }
        map.putIfAbsent("FIVE", 5);
        Integer value = map.get("TWO");
        System.out.println(value);
        System.out.println(map.containsKey("ONE"));
        System.out.println(map.containsValue(3));
        System.out.println(map.size());
        sc.close();
    }
}

```

	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 ►

Status	Finished
Started	Sunday, 17 November 2024, 12:36 PM
Completed	Sunday, 17 November 2024, 12:56 PM
Duration	20 mins 31 secs

Question **1**

Correct

Marked out of 5.00

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

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

Each alphabet is represented by a sequence of 0s.

This is as mentioned below:

Z : 0

Y : 00

X : 000

W : 0000

V : 00000

U : 000000

T : 0000000

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

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

Example 1:

input1: 010010001

The decoded string (original word) will be: ZYX

Example 2:

input1: 00001000000000000000000100000000000100000000010000000000001

The decoded string (original word) will be: WIPRO

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

For example:

Input	Result
010010001	ZYX
00001000000000000000000100000000000100000000010000000000001	WIPRO

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class Decoder {
    public static String decodeString(String encodedString) {
        String alphabet = "ZYXWVUTSRQPONMLKJIHGFEDCBA";
        StringBuilder decodedWord = new StringBuilder();
        String[] zeroGroups = encodedString.split("1");
        for (String group : zeroGroups) {
            if (!group.isEmpty()) {
                int numZeros = group.length();
                decodedWord.append(alphabet.charAt(numZeros - 1));
            }
        }
        return decodedWord.toString();
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String encodedString = scanner.nextLine();
        System.out.println(decodeString(encodedString));
    }
}
```

	Input	Expected	Got	
✓	010010001	ZYX	ZYX	✓
✓	00001000000000000000001000000000001000000000010000000000001	WIPRO	WIPRO	✓

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

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

Get the ASCII values of all the extracted alphabets.

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

Return that single digit as output.

Note:

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

Example 1:

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

input2: {'b', 'c'}

output: 8

Explanation:

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

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

98 + 99 = 197

1 + 9 + 7 = 17

1 + 7 = 8

For example:

Input	Result
a b c b c	8

Answer: (penalty regime: 0 %)

```
import java.util.HashSet;
public class CommonAlphabets {
    public static int getSingleDigitSum(int sum) {
        while (sum >= 10) {
            int tempSum = 0;
            while (sum > 0) {
                tempSum += sum % 10;
                sum /= 10;
            }
            sum = tempSum;
        }
        return sum;
    }
    public static int findCommonAsciiSum(char[] input1, char[] input2) {
        HashSet<Character> set1 = new HashSet<>();
        for (char c : input1) {
            set1.add(c);
        }
        int sum1 = 0;
        for (char c : input2) {
            if (set1.contains(c)) {
                sum1 += (int) c;
            }
        }
        return sum1;
    }
    public static void main(String[] args) {
        char[] input1 = {'a', 'b', 'c'};
        char[] input2 = {'b', 'c'};
```

```
        int sum1 = findCommonAsciiSum(input1, input2);
        int result = getSingleDigitSum(sum1);
        System.out.println(result);
    }
}

```

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

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

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 %)

```
import java.util.Scanner;
public class ReverseWordsWithCaseControl {
    public static String reverseWordsWithCase(String sentence, int case_option) {
        String[] words = sentence.split(" ");
        StringBuilder result = new StringBuilder();
        for (String word : words) {
            String reversedWord = reverseWord(word);
            if (case_option == 1) {
                reversedWord = reverseCaseWithOriginalPosition(reversedWord, word);
            }
           result.append(reversedWord).append(" ");
        }return result.toString().trim();
    }
    private static String reverseWord(String word) {
        StringBuilder reversed = new StringBuilder(word);
        return reversed.reverse().toString();
```

```java
    }
    private static String reverseCaseWithOriginalPosition(String reversedWord, String originalWord) {
        StringBuilder result = new StringBuilder(reversedWord);
            for (int i = 0; i < originalWord.length(); i++) {
            char originalChar = originalWord.charAt(i);
            char reversedChar = reversedWord.charAt(i);
            if (Character.isUpperCase(originalChar)) {
                result.setCharAt(i, Character.toUpperCase(reversedChar));
            } else if (Character.isLowerCase(originalChar)) {
                result.setCharAt(i, Character.toLowerCase(reversedChar));
            }
        }
return result.toString();
    }
    public static void printResult(String input, int case_option) {
        String result = reverseWordsWithCase(input, case_option);
        System.out.println(result);
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        String input = scanner.nextLine();
        int case_option = scanner.nextInt();
        printResult(input, case_option);
        scanner.close();
    }
}
```

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

Passed all tests! ✓

◄ Lab-12-MCQ

Jump to...

Identify possible words ►

PROGRAM CODE

1. Login And Signup Page

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class LoginPage extends JFrame {
    private JTextField usernameField;
    private JPasswordField passwordField;

    public LoginPage() {
        // Set up the frame
        setTitle("Login Page");
        setSize(400, 250);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new GridLayout(4, 1, 10, 10));

        // Set background color
        getContentPane().setBackground(new Color(102, 51, 153));  // Medium purple

        // Create components
        JLabel usernameLabel = new JLabel("Username:");
        JLabel passwordLabel = new JLabel("Password:");

        usernameField = new JTextField();
        passwordField = new JPasswordField();

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

```
JButton signupButton = new JButton("Sign Up");

// Set fonts
Font labelFont = new Font("Verdana", Font.BOLD, 14); // Bold font for labels
Font buttonFont = new Font("Verdana", Font.BOLD, 14); // Bold font for buttons

// Style components
usernameLabel.setFont(labelFont);
passwordLabel.setFont(labelFont);

// Set text color for username and password labels to white
usernameLabel.setForeground(Color.WHITE); // White text for username label
passwordLabel.setForeground(Color.WHITE); // White text for password label

// Set text color for username and password fields to purple
usernameField.setForeground(new Color(102, 51, 153)); // Purple text for username
passwordField.setForeground(new Color(102, 51, 153)); // Purple text for password
usernameField.setBackground(Color.WHITE); // White background for username field
passwordField.setBackground(Color.WHITE);  // White background for password field

loginButton.setFont(buttonFont);
signupButton.setFont(buttonFont);

// Set button background color to white
loginButton.setBackground(Color.WHITE);
signupButton.setBackground(Color.WHITE);

// Change text color of buttons to purple
loginButton.setForeground(new Color(102, 51, 153)); // Purple text for Login button
signupButton.setForeground(new Color(102, 51, 153)); // Purple text for Sign Up button

// Set border for better visibility of the buttons
```

```
    loginButton.setBorder(BorderFactory.createLineBorder(Color.DARK_GRAY, 1));
    signupButton.setBorder(BorderFactory.createLineBorder(Color.DARK_GRAY, 1));

    // Add components to frame
    add(usernameLabel);
    add(usernameField);
    add(passwordLabel);
    add(passwordField);
    add(loginButton);
    add(signupButton);

    // Add button listeners
    loginButton.addActionListener(new LoginAction());
    signupButton.addActionListener(new SignupAction());

    setVisible(true);
}

// ActionListener for the login button
class LoginAction implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        String username = usernameField.getText();
        String password = new String(passwordField.getPassword());

        // Check the login credentials (username and password)
        if (username.equals("admin") && password.equals("admin")) {
            JOptionPane.showMessageDialog(LoginPage.this, "Login Successful!");
            dispose(); // Close the current login page
            new HomePage(username); // Pass the logged-in username to the HomePage
        } else {
            JOptionPane.showMessageDialog(LoginPage.this, "Invalid username or password");
```

```
            }
        }
    }

    // ActionListener for the signup button
    class SignupAction implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            dispose();
            new SignupPage(); // Open the signup page
        }
    }

    public static void main(String[] args) {
        new LoginPage(); // Show login page
    }
}

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

public class SignupPage extends JFrame {
    private JTextField usernameField;
    private JPasswordField passwordField, confirmPasswordField;

    public SignupPage() {
```

```
// Set up frame
setTitle("Sign Up Page");
setSize(400, 300);
setLocationRelativeTo(null);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new BorderLayout(10, 10)); // BorderLayout with 10px gaps

// Set the background color of the entire frame to purple
getContentPane().setBackground(new Color(102, 51, 153));  // Purple background

// Create a panel for the form components (username, password fields, etc.)
JPanel formPanel = new JPanel();
formPanel.setLayout(new GridLayout(4, 2, 10, 10)); // Grid layout for form fields
formPanel.setBackground(new Color(102, 51, 153));  // Set form panel background to purple

// Create components
JLabel usernameLabel = new JLabel("Username:");
JLabel passwordLabel = new JLabel("Password:");
JLabel confirmPasswordLabel = new JLabel("Confirm Password:");

usernameField = new JTextField(20);
passwordField = new JPasswordField(20);
confirmPasswordField = new JPasswordField(20);

// Set text color for labels to white
usernameLabel.setForeground(Color.WHITE);
passwordLabel.setForeground(Color.WHITE);
confirmPasswordLabel.setForeground(Color.WHITE);

// Set text color inside text fields to purple and background to white
Color purple = new Color(102, 51, 153); // Purple text color
usernameField.setForeground(purple);  // Purple text for username
```

```
passwordField.setForeground(purple); // Purple text for password
confirmPasswordField.setForeground(purple); // Purple text for confirm password
usernameField.setBackground(Color.WHITE); // White background for username field
passwordField.setBackground(Color.WHITE);  // White background for password field
confirmPasswordField.setBackground(Color.WHITE);  // White background for confirm password field

// Add components to form panel
formPanel.add(usernameLabel);
formPanel.add(usernameField);
formPanel.add(passwordLabel);
formPanel.add(passwordField);
formPanel.add(confirmPasswordLabel);
formPanel.add(confirmPasswordField);

// Add the form panel to the center of the frame
add(formPanel, BorderLayout.CENTER);

// Create buttons panel (to be placed at the bottom)
JPanel buttonPanel = new JPanel();
buttonPanel.setLayout(new FlowLayout(FlowLayout.CENTER, 20, 10)); // FlowLayout with gap between
buttons

JButton saveButton = new JButton("Save");
JButton cancelButton = new JButton("Back");

// Set fonts for buttons
Font buttonFont = new Font("Verdana", Font.BOLD, 14);
saveButton.setFont(buttonFont);
cancelButton.setFont(buttonFont);

// Set preferred size of buttons
saveButton.setPreferredSize(new Dimension(100, 30)); // Reduced button size
cancelButton.setPreferredSize(new Dimension(100, 30)); // Reduced button size
```

```
    // Set background colors for the buttons
    saveButton.setBackground(Color.WHITE); // White background for Save button
    cancelButton.setBackground(Color.WHITE);  // White background for Back button

    // Set foreground (text color) of buttons to purple
    saveButton.setForeground(purple);       // Text color for Save button
    cancelButton.setForeground(purple);     // Text color for Back button

    // Add buttons to the button panel
    buttonPanel.add(saveButton);
    buttonPanel.add(cancelButton);

    // Add the button panel to the SOUTH region of the frame
    add(buttonPanel, BorderLayout.SOUTH);

    // Add button listeners
    saveButton.addActionListener(new SaveAction());
    cancelButton.addActionListener(new CancelAction());

    setVisible(true);
}

// ActionListener for the Save button
class SaveAction implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        String username = usernameField.getText();
        String password = new String(passwordField.getPassword());
        String confirmPassword = new String(confirmPasswordField.getPassword());

        // Validate input
```

```
        if (username.isEmpty() || password.isEmpty() || confirmPassword.isEmpty()) {
            JOptionPane.showMessageDialog(SignupPage.this, "Please fill all fields.");
            return;
        }

        if (!password.equals(confirmPassword)) {
            JOptionPane.showMessageDialog(SignupPage.this, "Passwords do not match.");
            return;
        }

        // Save the user in the database
        saveUserToDatabase(username, password);
    }
}

// ActionListener for the Back to Login button
class CancelAction implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        dispose(); // Close the signup page
        new LoginPage(); // Show the login page
    }
}

// Method to save the user data into the database
private void saveUserToDatabase(String username, String password) {
    String url = "jdbc:mysql://localhost/student_management";
    String dbUsername = "root"; // Your MySQL username
    String dbPassword = "2105"; // Your MySQL password
    String insertQuery = "INSERT INTO users (username, password) VALUES (?, ?)";

    try (Connection conn = DriverManager.getConnection(url, dbUsername, dbPassword);
```

```
            PreparedStatement stmt = conn.prepareStatement(insertQuery)) {

            stmt.setString(1, username);
            stmt.setString(2, password);

            // Execute the insert query
            int rowsAffected = stmt.executeUpdate();

            if (rowsAffected > 0) {
                JOptionPane.showMessageDialog(SignupPage.this, "Sign Up Successful!");
                dispose();
                new LoginPage(); // Redirect to login page after successful signup
            } else {
                JOptionPane.showMessageDialog(SignupPage.this, "Error occurred while saving user.");
            }

        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(SignupPage.this, "Database connection error: " +
ex.getMessage());
            ex.printStackTrace();
        }
    }

    public static void main(String[] args) {
        new SignupPage(); // Show signup page
    }
}
```

2. Home Page

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
```

```
import java.awt.event.ActionListener;
import java.awt.image.BufferedImage;
import javax.imageio.ImageIO;
import java.io.File;
import java.io.IOException;

public class HomePage extends JFrame {

    private ImagePanel imagePanel;
    private Timer slideshowTimer;
    private int currentImageIndex = 0;
    private float alpha = 1.0f; // Transparency level (1.0 is fully opaque)
    private final int FADE_SPEED = 50; // Adjust fade speed (increase delay for slower fade)
    private final float FADE_INCREMENT = 0.02f; // Adjust fade increment (smaller values for slower fade)
    private boolean fadingOut = true; // Track whether we're fading out or in

    // List of image file paths for the slideshow
    private String[] imagePaths = {
        "C:\\Users\\Shyaan Shaikh\\OneDrive\\Documents\\Shayaan Shaikh\\Syllabus\\Java\\rec.jpg",
        "C:\\Users\\Shyaan Shaikh\\OneDrive\\Documents\\Shayaan Shaikh\\Syllabus\\Java\\images.jpg",
        "C:\\Users\\Shyaan Shaikh\\OneDrive\\Documents\\Shayaan Shaikh\\Syllabus\\Java\\images2.jpg",
        "C:\\Users\\Shyaan Shaikh\\OneDrive\\Documents\\Shayaan Shaikh\\Syllabus\\Java\\images3.jpg",
        "C:\\Users\\Shyaan Shaikh\\OneDrive\\Documents\\Shayaan Shaikh\\Syllabus\\Java\\image4.jpg"
    };

    public HomePage(String username) {
        // Set up the frame
        setTitle("Home Page");
        setSize(700, 400); // Further increase frame width for a landscape layout
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());
```

```
// Background color set to dull white
Color dullWhite = new Color(245, 245, 245);  // Dull white color

// Button color set to medium purple
Color mediumPurple = new Color(102, 51, 153);  // Medium purple color

// Create buttons panel for Home, Academics, Result, Edit, and List at the top
JPanel buttonPanel = new JPanel(new GridLayout(1, 5, 10, 10));
buttonPanel.setBackground(dullWhite);  // Set background color to dull white

// Create each button with medium purple color
JButton homeButton = createButton("Home", mediumPurple);
JButton academicsButton = createButton("Academics", mediumPurple);
JButton resultButton = createButton("Result", mediumPurple);
JButton editButton = createButton("Edit", mediumPurple);
JButton listButton = createButton("List", mediumPurple);

// Add action listeners to buttons
homeButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Reload the page or reset the slideshow
        currentImageIndex = 0; // Reset to the first image
        startSlideshowWithFadeEffect(); // Restart the slideshow
    }
});

academicsButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Navigate to the Academics page (replace with actual page)
        new AcademicsPage(); // Assuming you have a separate page for Academics
        dispose(); // Close current HomePage
```

```
    }
});

resultButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Navigate to the Result page (replace with actual page)
        new ResultPage(); // Assuming you have a separate page for Results
        dispose(); // Close current HomePage
    }
});

editButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Open the EditPage for editing
        new EditPage(); // Assuming you have a separate page for editing students
        dispose(); // Close current HomePage
    }
});

listButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        // Open the ListPage for displaying students
        new StudentListPage(); // Assuming you have a page for student list
        dispose(); // Close current HomePage
    }
});

// Add buttons to the panel
buttonPanel.add(homeButton);
buttonPanel.add(academicsButton);
buttonPanel.add(resultButton);
buttonPanel.add(editButton);
```

```
    buttonPanel.add(listButton);
    add(buttonPanel, BorderLayout.NORTH);

    // Create the ImagePanel for fading effect in the center
    imagePanel = new ImagePanel();
    add(imagePanel, BorderLayout.CENTER);

    // Create a new panel for the logout button at the bottom right
    JPanel logoutPanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));
    logoutPanel.setBackground(dullWhite);  // Match background color

    // Change the logout button to red color
    Color redColor = new Color(255, 0, 0);  // Red color
    JButton logoutButton = createButton("Logout", redColor);
    logoutButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            // Perform logout action (close HomePage and open LoginPage)
            new LoginPage(); // Assuming you have a LoginPage to navigate back to
            dispose(); // Close current HomePage
        }
    });

    logoutPanel.add(logoutButton);
    add(logoutPanel, BorderLayout.SOUTH);  // Place the logout panel at the bottom

    // Start the slideshow with fade effect
    startSlideshowWithFadeEffect();

    setVisible(true);
}

// Start the slideshow with a fade effect
```

```
private void startSlideshowWithFadeEffect() {
    slideshowTimer = new Timer(FADE_SPEED, new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            if (fadingOut) {
                alpha -= FADE_INCREMENT; // Decrease alpha for fade out
                if (alpha <= 0) {
                    fadingOut = false; // Switch to fade in
                    currentImageIndex = (currentImageIndex + 1) % imagePaths.length;
                    updateImage();
                }
            } else {
                alpha += FADE_INCREMENT; // Increase alpha for fade in
                if (alpha >= 1) {
                    fadingOut = true; // Start fading out again
                }
            }
            imagePanel.repaint();
        }
    });
    updateImage();
    slideshowTimer.start();
}

// Helper method to update the image for the next slide
private void updateImage() {
    imagePanel.setImage(loadImage(imagePaths[currentImageIndex]));
    alpha = 0.0f; // Start new image at transparent state
}

// Load image from the file system
private BufferedImage loadImage(String path) {
```

```
    try {
        return ImageIO.read(new File(path));
    } catch (IOException e) {
        e.printStackTrace();
        return null;
    }
}

// Helper method to create styled buttons with specific colors
private JButton createButton(String text, Color color) {
    JButton button = new JButton(text);
    Font boldFont = new Font("Verdana", Font.BOLD, 10);
    button.setFont(boldFont);
    button.setBackground(color);
    button.setForeground(Color.WHITE);
    button.setFocusPainted(false);
    button.setPreferredSize(new Dimension(80, 30));
    return button;
}

// Inner class to handle image drawing with alpha transparency
private class ImagePanel extends JPanel {
    private BufferedImage image;

    public void setImage(BufferedImage image) {
        this.image = image;
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        if (image != null) {
```

```
            Graphics2D g2d = (Graphics2D) g;
            g2d.setComposite(AlphaComposite.getInstance(AlphaComposite.SRC_OVER, alpha));
            // Draw image with further increased width (600 px) and reduced height (200 px) centered
            int x = (getWidth() - 600) / 2;
            int y = (getHeight() - 200) / 2;
            g2d.drawImage(image, x, y, 600, 200, null);
        }
    }
  }

  public static void main(String[] args) {
    new HomePage("User");
  }
}
```

3. Edit Page

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

public class EditPage extends JFrame {
  private JTextField studentIdField, studentNameField, studentMajorField, studentGPAField;

  public EditPage() {
    // Set up the frame
    setTitle("Edit Student Information");
    setSize(500, 400);
```

```
setLocationRelativeTo(null);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setLayout(new BorderLayout());

// Create a panel to hold student input fields
JPanel inputPanel = new JPanel(new GridLayout(5, 2, 10, 10));
inputPanel.setPreferredSize(new Dimension(500, 150));

JLabel studentIdLabel = new JLabel("Student ID:");
JLabel studentNameLabel = new JLabel("Name:");
JLabel studentMajorLabel = new JLabel("Major:");
JLabel studentGPALabel = new JLabel("GPA:");

studentIdField = new JTextField(15);
studentNameField = new JTextField(15);
studentMajorField = new JTextField(15);
studentGPAField = new JTextField(15);

inputPanel.add(studentIdLabel);
inputPanel.add(studentIdField);
inputPanel.add(studentNameLabel);
inputPanel.add(studentNameField);
inputPanel.add(studentMajorLabel);
inputPanel.add(studentMajorField);
inputPanel.add(studentGPALabel);
inputPanel.add(studentGPAField);

add(inputPanel, BorderLayout.NORTH);

// Create buttons panel for Add and Back
JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 20, 20));
```

```
    // Change "Add" button to medium purple color
    Color mediumPurple = new Color(102, 51, 153);  // Medium purple color
    JButton addButton = createButton("Add", mediumPurple);
    buttonPanel.add(addButton);

    add(buttonPanel, BorderLayout.CENTER);

    // Create Back button in the bottom right corner with a red background
    JButton backButton = createButton("Back", Color.RED);  // Set the Back button color to red
    JPanel backButtonPanel = new JPanel(new FlowLayout(FlowLayout.RIGHT));
    backButtonPanel.add(backButton);
    add(backButtonPanel, BorderLayout.SOUTH);

    // Action listeners for buttons
    addButton.addActionListener(new AddButtonListener());
    backButton.addActionListener(new BackButtonListener());

    setVisible(true);
}

private JButton createButton(String text, Color backgroundColor) {
    JButton button = new JButton(text);
    Font boldFont = new Font("Verdana", Font.BOLD, 12);
    button.setFont(boldFont);
    button.setBackground(backgroundColor);
    button.setForeground(Color.WHITE); // White text color
    button.setFocusPainted(false);
    button.setPreferredSize(new Dimension(100, 30));
    return button;
}

// ActionListener for the "Add" button
```

```
class AddButtonListener implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        String studentId = studentIdField.getText();
        String studentName = studentNameField.getText();
        String studentMajor = studentMajorField.getText();
        String studentGPA = studentGPAField.getText();

        if (studentId.isEmpty() || studentName.isEmpty() || studentMajor.isEmpty() ||
studentGPA.isEmpty()) {
            JOptionPane.showMessageDialog(EditPage.this, "Please fill all fields.");
            return;
        }

        // Insert student data into the database
        try (Connection conn = DBConnection.getConnection()) {
            String sql = "INSERT INTO students (student_id, name, major, gpa) VALUES (?, ?, ?, ?)";
            try (PreparedStatement stmt = conn.prepareStatement(sql)) {
                stmt.setString(1, studentId);
                stmt.setString(2, studentName);
                stmt.setString(3, studentMajor);
                stmt.setString(4, studentGPA);
                stmt.executeUpdate();
                JOptionPane.showMessageDialog(EditPage.this, "Student added to database!");
            }
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(EditPage.this, "Error saving student to the database.");
            ex.printStackTrace();
        }

        studentIdField.setText("");
        studentNameField.setText("");
        studentMajorField.setText("");
```

```
            studentGPAField.setText("");
        }
    }

    // ActionListener for the "Back" button
    class BackButtonListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            dispose();  // Close the current EditPage
            new HomePage("User");  // Open the home page with a placeholder username
        }
    }

    public static void main(String[] args) {
        new EditPage();  // Launch the EditPage when executed
    }
}
```

4. Academics Page

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class AcademicsPage extends JFrame {

    private JTextField searchField;
    private JTextArea resultArea;

    // Database credentials
```

```
private static final String DB_URL = "jdbc:mysql://localhost:3306/student_management";
private static final String USER = "root";  // Replace with your DB username
private static final String PASSWORD = "2105";  // Replace with your DB password

public AcademicsPage() {
    setTitle("Academics Page");
    setSize(500, 400);
    setLocationRelativeTo(null);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLayout(new BorderLayout());  // Use BorderLayout for easy positioning

    // Panel for the search bar
    JPanel searchPanel = new JPanel();
    JLabel label = new JLabel("Search by Student ID or Name:");
    label.setFont(new Font("Verdana", Font.PLAIN, 14));

    searchField = new JTextField(20);
    searchPanel.add(label);
    searchPanel.add(searchField);

    // Text area to display search results
    resultArea = new JTextArea(10, 40);
    resultArea.setEditable(false);
    resultArea.setFont(new Font("Verdana", Font.PLAIN, 12));
    JScrollPane scrollPane = new JScrollPane(resultArea);

    // Add search panel to the top of the frame
    add(searchPanel, BorderLayout.NORTH);

    // Add text area for results in the center
    add(scrollPane, BorderLayout.CENTER);
```

```
// Create search and back buttons (placed at the bottom)
JButton searchButton = new JButton("Search");
JButton backButton = new JButton("Back");

// Set properties for the buttons
searchButton.setPreferredSize(new Dimension(80, 30)); // Set button size
searchButton.setBackground(new Color(102, 51, 153)); // Set background color to purple
searchButton.setForeground(Color.WHITE); // Set text color to white
searchButton.setFont(new Font("Verdana", Font.BOLD, 12)); // Set button font

backButton.setPreferredSize(new Dimension(80, 30)); // Set button size
backButton.setBackground(Color.RED); // Set red background for back button
backButton.setForeground(Color.WHITE); // Set text color to white
backButton.setFont(new Font("Verdana", Font.BOLD, 12)); // Set button font

// Add buttons to the bottom panel
JPanel buttonPanel = new JPanel();
buttonPanel.setLayout(new FlowLayout(FlowLayout.CENTER)); // Center the buttons
buttonPanel.setBackground(Color.WHITE);  // Set background to white
buttonPanel.add(searchButton);
buttonPanel.add(backButton); // Add the back button as well
add(buttonPanel, BorderLayout.SOUTH);

// Action listener for the search button
searchButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String query = searchField.getText().trim();
        if (!query.isEmpty()) {
            String result = getStudentInfo(query);
            resultArea.setText(result);  // Display the result in the text area
        } else {
```

```
            resultArea.setText("Please enter a Student ID or Name.");
        }
    }
    });

    // Action listener for the back button
    backButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            dispose();  // Close the current AcademicsPage
            new HomePage("User");  // Open the HomePage with a placeholder username
        }
    });

    setVisible(true);
}

// Method to fetch student info from the database
private String getStudentInfo(String query) {
    String result = "";
    Connection conn = null;
    PreparedStatement stmt = null;
    ResultSet rs = null;

    try {
        // Establish a connection to the database
        conn = DriverManager.getConnection(DB_URL, USER, PASSWORD);

        // Query to search by Student ID or Name
        String sql = "SELECT * FROM students WHERE student_id = ? OR name = ?";
        stmt = conn.prepareStatement(sql);
        stmt.setString(1, query);
```

```
        stmt.setString(2, query);

        // Execute query
        rs = stmt.executeQuery();

        // Process the result
        if (rs.next()) {
            result = "Student ID: " + rs.getString("student_id") + "\n" +
                     "Name: " + rs.getString("name") + "\n" +
                     "Major: " + rs.getString("major") + "\n" +
                     "GPA: " + rs.getString("gpa");
        } else {
            result = "No student found with ID or Name: " + query;
        }
    } catch (SQLException e) {
        e.printStackTrace();
        result = "Database error: " + e.getMessage();
    } finally {
        try {
            if (rs != null) rs.close();
            if (stmt != null) stmt.close();
            if (conn != null) conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    return result;
}

public static void main(String[] args) {
    new AcademicsPage();  // Launch the AcademicsPage when executed
```

```
    }
}
```

5. Result Page

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

public class ResultPage extends JFrame {

    private JTextField searchField;
    private JTextArea resultArea;

    // Database credentials
    private static final String DB_URL = "jdbc:mysql://localhost:3306/student_management";
    private static final String USER = "root";  // Replace with your DB username
    private static final String PASSWORD = "2105";  // Replace with your DB password

    public ResultPage() {
        setTitle("Result Page");
        setSize(500, 400);
        setLocationRelativeTo(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());  // Use BorderLayout for easy positioning

        // Panel for the search bar
        JPanel searchPanel = new JPanel();
        JLabel label = new JLabel("Search by Student ID or Name:");
        label.setFont(new Font("Verdana", Font.PLAIN, 14));
```

```
searchField = new JTextField(20);
searchPanel.add(label);
searchPanel.add(searchField);

// Text area to display search results
resultArea = new JTextArea(10, 40);
resultArea.setEditable(false);
resultArea.setFont(new Font("Verdana", Font.PLAIN, 12));
JScrollPane scrollPane = new JScrollPane(resultArea);

// Add search panel to the top of the frame
add(searchPanel, BorderLayout.NORTH);

// Add text area for results in the center
add(scrollPane, BorderLayout.CENTER);

// Create search and back buttons (placed at the bottom)
JButton searchButton = new JButton("Search");
JButton backButton = new JButton("Back");

// Set properties for the buttons
searchButton.setPreferredSize(new Dimension(80, 30)); // Set button size
searchButton.setBackground(new Color(102, 51, 153)); // Medium purple color for Search button
searchButton.setForeground(Color.WHITE); // Set text color to white
searchButton.setFont(new Font("Verdana", Font.BOLD, 12)); // Set button font

backButton.setPreferredSize(new Dimension(80, 30)); // Set button size
backButton.setBackground(Color.RED); // Red color for Back button
backButton.setForeground(Color.WHITE); // Set text color to white
backButton.setFont(new Font("Verdana", Font.BOLD, 12)); // Set button font
```

```
// Add buttons to the bottom panel
JPanel buttonPanel = new JPanel();
buttonPanel.setLayout(new FlowLayout(FlowLayout.CENTER)); // Center the buttons
buttonPanel.setBackground(Color.WHITE);  // Set background to white
buttonPanel.add(searchButton);
buttonPanel.add(backButton); // Add the back button as well
add(buttonPanel, BorderLayout.SOUTH);

// Action listener for the search button
searchButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String query = searchField.getText().trim();
        if (!query.isEmpty()) {
            String result = getStudentGPA(query);
            resultArea.setText(result);  // Display the result in the text area
        } else {
            resultArea.setText("Please enter a Student ID or Name.");
        }
    }
});

// Action listener for the back button
backButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        dispose();  // Close the current ResultPage
        new HomePage("User");  // Open the HomePage with a placeholder username
    }
});

setVisible(true);
```

```
}

// Method to fetch student's GPA from the database
private String getStudentGPA(String query) {
    String result = "";
    Connection conn = null;
    PreparedStatement stmt = null;
    ResultSet rs = null;

    try {
        // Establish a connection to the database
        conn = DriverManager.getConnection(DB_URL, USER, PASSWORD);

        // Query to search by Student ID or Name
        String sql = "SELECT gpa FROM students WHERE student_id = ? OR name = ?";
        stmt = conn.prepareStatement(sql);
        stmt.setString(1, query);
        stmt.setString(2, query);

        // Execute query
        rs = stmt.executeQuery();

        // Process the result
        if (rs.next()) {
            result = "GPA: " + rs.getString("gpa");
        } else {
            result = "No student found with ID or Name: " + query;
        }
    } catch (SQLException e) {
        e.printStackTrace();
        result = "Database error: " + e.getMessage();
    } finally {
```

```
        try {
            if (rs != null) rs.close();
            if (stmt != null) stmt.close();
            if (conn != null) conn.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    return result;
  }

  public static void main(String[] args) {
    new ResultPage();  // Launch the ResultPage when executed
  }
}
```

6. Student List Page

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

public class StudentListPage extends JFrame {
  private DefaultListModel<String> listModel;
  private JList<String> studentList;
```

```
public StudentListPage() {
    // Set up the frame
    setTitle("Student List");
    setSize(500, 400);
    setLocationRelativeTo(null);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    listModel = new DefaultListModel<>();
    studentList = new JList<>(listModel);

    JScrollPane listScrollPane = new JScrollPane(studentList);
    add(listScrollPane, BorderLayout.CENTER);

    // Fetch students from the database and display them
    loadStudentList();

    // Create a delete button to delete the selected student
    JButton deleteButton = new JButton("Delete");
    deleteButton.setFont(new Font("Verdana", Font.BOLD, 12));
    deleteButton.setBackground(Color.RED);  // Set Delete button color to red
    deleteButton.setForeground(Color.WHITE);

    deleteButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            deleteSelectedStudent();
        }
    });

    // Create a back button to return to the home page with medium purple color
    JButton backButton = new JButton("Back");
    Color mediumPurple = new Color(102, 51, 153);  // Medium purple color
```

```
    backButton.setFont(new Font("Verdana", Font.BOLD, 12));
    backButton.setBackground(mediumPurple); // Set Back button color to medium purple
    backButton.setForeground(Color.WHITE);

    backButton.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            // Navigate back to the HomePage (assuming you have a HomePage class)
            new HomePage("User");  // Replace "User" with the actual username if needed
            dispose();  // Close the current StudentListPage
        }
    });

    // Create a panel for the buttons (Delete and Back)
    JPanel buttonPanel = new JPanel();
    buttonPanel.setLayout(new FlowLayout(FlowLayout.CENTER));
    buttonPanel.add(deleteButton);
    buttonPanel.add(backButton);
    add(buttonPanel, BorderLayout.SOUTH);

    setVisible(true);
}

private void loadStudentList() {
    try (Connection conn = DBConnection.getConnection()) {
        String sql = "SELECT name, student_id FROM students";
        try (PreparedStatement stmt = conn.prepareStatement(sql)) {
            ResultSet rs = stmt.executeQuery();

            while (rs.next()) {
                String studentName = rs.getString("name");
                String studentId = rs.getString("student_id");
```

```
                listModel.addElement(studentName + " (" + studentId + ")");
            }
        }
    } catch (SQLException ex) {
        JOptionPane.showMessageDialog(this, "Error fetching students from the database.");
        ex.printStackTrace();
    }
}

private void deleteSelectedStudent() {
    // Get the selected student from the list
    String selectedStudent = studentList.getSelectedValue();

    if (selectedStudent == null) {
        JOptionPane.showMessageDialog(this, "Please select a student to delete.");
        return;
    }

    // Extract student ID from the selected item (format: Name (student_id))
    String studentId = selectedStudent.substring(selectedStudent.indexOf("(") + 1,
selectedStudent.indexOf(")"));

    // Confirm deletion
    int confirm = JOptionPane.showConfirmDialog(this, "Are you sure you want to delete student " +
selectedStudent + "?",
            "Confirm Deletion", JOptionPane.YES_NO_OPTION);

    if (confirm == JOptionPane.YES_OPTION) {
        // Delete the student from the database
        try (Connection conn = DBConnection.getConnection()) {
            String deleteSql = "DELETE FROM students WHERE student_id = ?";
            try (PreparedStatement stmt = conn.prepareStatement(deleteSql)) {
                stmt.setString(1, studentId);
```

```
                int rowsAffected = stmt.executeUpdate();

                if (rowsAffected > 0) {
                    // Remove the student from the list model (UI)
                    listModel.removeElement(selectedStudent);
                    JOptionPane.showMessageDialog(this, "Student deleted successfully.");
                } else {
                    JOptionPane.showMessageDialog(this, "Student deletion failed.");
                }
            }
        } catch (SQLException ex) {
            JOptionPane.showMessageDialog(this, "Error deleting student from the database.");
            ex.printStackTrace();
        }
    }
}

    public static void main(String[] args) {
        new StudentListPage();  // Launch the StudentListPage when executed
    }
}
```

7. Java/MySQL Connectivity

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

public class DBConnection {
    private static final String URL = "jdbc:mysql://localhost:/student_management";
    private static final String USER = "root"; // Your MySQL username
    private static final String PASSWORD = "2105"; // Your MySQL password
```

```
private static Connection connection;

public static Connection getConnection() {
    try {
        // Ensure MySQL driver is loaded
        Class.forName("com.mysql.cj.jdbc.Driver");

        // Create a new connection if none exists or if the connection is closed
        if (connection == null || connection.isClosed()) {
            connection = DriverManager.getConnection(URL, USER, PASSWORD);
            System.out.println("Database connection established successfully.");
        }
    } catch (ClassNotFoundException e) {
        System.out.println("MySQL JDBC Driver not found.");
        e.printStackTrace();
    } catch (SQLException e) {
        System.out.println("SQL Exception: " + e.getMessage());
        System.out.println("Error Code: " + e.getErrorCode());
        System.out.println("SQL State: " + e.getSQLState());
        e.printStackTrace();
    }
    return connection;
}

public static void closeConnection() {
    try {
        if (connection != null && !connection.isClosed()) {
            connection.close();
            System.out.println("Database connection closed.");
        }
    } catch (SQLException e) {
        System.out.println("Error closing database connection.");
```

```
            e.printStackTrace();
        }
    }
}


import java.sql.Connection;  // Import the Connection class from java.sql


public class App {
    public static void main(String[] args) {
        // Example usage of the DBConnection class
        Connection conn = DBConnection.getConnection(); // Use DBConnection, not
        dbconnection


        if (conn != null) {
            System.out.println("Database connection established successfully.");
        } else {
            System.out.println("Failed to establish database connection.");
        }


        // Closing the connection when done
        DBConnection.closeConnection(); // Use DBConnection, not dbconnection
    }
}
```