Dashboard / My courses / CS23333-OOPUJ-2023 / Lab-01-Java Architecture, Language Basics / Lab-01-Logic Building

Status	Finished
Started	Tuesday, 1 October 2024, 9:35 AM
Completed	Tuesday, 1 October 2024, 9:59 AM
Duration	23 mins 48 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 n=sc.nextInt();
        if(n%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 %)

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

	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.*;
public class sa{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int m=sc.nextInt();
        if(n<0)
        n*=-1;
        if(m<0)
        m*=-1;
        int sum=n%10+m%10;
        System.out.print(sum);
    }
}
```

	Input	Expected	Got	
✓		11	11	✓

	267 154			
✓	267 -154	11	11	✓
✓	-267 154	11	11	✓
✓	-267 -154	11	11	✓

Passed all tests! ✓

◄ Lab-01-MCQ

Jump to...

Is Even? ►

Dashboard / My courses / CS23333-OOPUJ-2023 / Lab-02-Flow Control Statements / Lab-02-Logic Building

Status	Finished
Started	Sunday, 6 October 2024, 2:32 PM
Completed	Sunday, 6 October 2024, 3:07 PM
Duration	35 mins 4 secs

Question **1**

Correct

Marked out of 5.00

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

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

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

Input Format

Input containing an integer and a boolean value.

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

You have to print the time you should get up.

Example Input:

1 false

Output:

6:00

Example Input:

5 false

Output:

5:00

Example Input:

1 true

Output:

9:00

For example:

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

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class WakeUpTime{
    public static void main(String [] args){
        Scanner sc=new Scanner(System.in);
        int day=sc.nextInt();
        boolean vacation=sc.nextBoolean();
        String wakeUpTime=getWakeUpTime(day,vacation);
                System.out.println(wakeUpTime);
        sc.close();
    }
    public static String getWakeUpTime(int day,boolean vacation){
        String time;
        if(vacation){
            if(day==1||day==7){
                time="9:00";
            }
            else{
                time="7:00";
            }
        }else{
            if(day==1||day==7){
                time="6:00";
            }
            else{
                time="5:00";
            }
        }
        return time;
    }
}
```

	Input	Expected	Got	
✓	1 false	6:00	6:00	✓
✓	5 false	5:00	5:00	✓
✓	1 true	9:00	9:00	✓

Passed all tests! ✓

Question **2**

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
import java.util.Scanner;
public class FactorialTriallingZeros{
    public static int countTrailingZeros(int n)
    {
        int count=0;
        for (int i = 5; n / i >= 1;i*=5)
            count += n / i;

        return count;
    }

    public static void main(String[] args)
    {
        Scanner sc= new Scanner(System.in);
        int n=sc.nextInt();
        System.out.println(countTrailingZeros(n));
        sc.close();
    }
}

```

	Input	Expected	Got	
✓	3	0	0	✓

✓	60	14	14	✓
✓	100	24	24	✓
✓	1024	253	253	✓

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 %)

```
import java.util.Scanner;
public class CustomSequence{
    public static int findTerm(int n){
        if(n==1)
        return 0;
        if(n==2 || n==3)
        return 1;
        if(n==4)
        return 2;
        if(n==5)
        return 4;
        int []sequence=new int[n];
        sequence[0]=0;
        sequence[1]=1;
        sequence[2]=1;
        sequence[3]=2;
        sequence[4]=4;
        for(int i=5;i<n;i++)
        {
            sequence[i]=sequence[i-1]+sequence[i-2]+sequence[i-3];
        }
        return sequence[n-1];
    }
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int n=sc.nextInt();
        int term=findTerm(n);
        System.out.println(term);
    }
}
```

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

Passed all tests! ✓

◄ Lab-02-MCQ

Jump to...

Lab-03-MCQ ►

Dashboard / My courses / CS23333-OOPUJ-2023 / Lab-03-Arrays / Lab-03-Logic Building

Status	Finished
Started	Sunday, 6 October 2024, 3:10 PM
Completed	Sunday, 6 October 2024, 4:01 PM
Duration	51 mins 24 secs

Question **1**

Correct

Marked out of 5.00

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

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

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

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

input1 represents the number of elements in the array.

input2 represents the array of integers.

Example 1:

input1 = 16

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

Expected output = 62

Explanation:

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

Example 2:

input1 = 11

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

Expected output = -1

Explanation:

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

Example 3:

input1 = 16

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

Expected output = 174

Explanation:

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

26 + 92) + (12 + 0 + 12) = 174.

For example:

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

Answer: (penalty regime: 0 %)

```java
import java.util.Scanner;
public class Longest{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        System.out.print("");
        int input1=sc.nextInt();
        int[] input2=new int[input1];
        System.out.println("");
        for(int i=0;i<input1;i++){
            input2[i]=sc.nextInt();
        }
        int maxLength=0;
        int currentLength=0;
        int currentSum=0;
        int maxSum=0;
        for(int num:input2){
            if(num>=0){
                currentLength++;
                currentSum+=num;
            }
            else{
                if(currentLength>maxLength){
                    maxLength=currentLength;
                    maxSum=currentSum;
                }
                else if(currentLength==maxLength){
                    maxSum+=currentSum;
                }
                currentLength=0;
                currentSum=0;
            }
        }
        if(currentLength>maxLength){
            maxSum=currentSum;
        }
        else if(currentLength==maxLength){
            maxSum+=currentSum;
        }
        if(maxLength==0){
            System.out.println(-1);
        }
        else{
            System.out.print(maxSum);
        }
        sc.close();
    }
}
```

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

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

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

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

After the operations are done, return the resultant array.

Example 1:

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

input2 = {1, 5, 6, 9}

Expected Output = {-72, -36, 27, 0}

Explanation:

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

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

{(1 - 9), (5 - 9), (6 - 9), (9 - 9)} = {-8, -4, -3, 0}

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

{(-8 x 9), (-4 x 9), (3 x 9), (0 x 9)} = {-72, -36, -27, 0}

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

Example 2:

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

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

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

Explanation:

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

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

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

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

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

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

Example 3:

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

input2 = {-9, 9}

Expected Output = {-162, 0}

Explanation:

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

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

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

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

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

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

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

For example:

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

Answer: (penalty regime: 0 %)

```java
import java.util.Scanner;
public class AOp{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        int input1=sc.nextInt();
        int[] input2=new int[input1];
        for(int i=0;i<input1;i++)
        {
            input2[i]=sc.nextInt();
        }
        int max=Integer.MIN_VALUE;
        for(int num:input2){
            if(num>max){
                max=num;
            }
        }
        for(int i=0;i<input1;i++)
        {
            input2[i]=input2[i]-max;
        }
        for(int i=0;i<input1;i++){
            input2[i]=input2[i]*max;
        }
        for(int i=0;i<input1;i++)
        {
            System.out.print(input2[i]);
            if(i<input1-1){
                System.out.print(" ");
            }
        }
        sc.close();
    }
}
```

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

Passed all tests! ✓

Question 3

Correct

Marked out of 5.00

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

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

This is explained below:

Example 1:

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

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

Step 1:

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

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

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

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

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

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

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

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

Step 2:

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

{1, 25, 16, 49, 16}

Step 3:

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

Note:

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

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

Example 2:

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

Step 1:

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

{1, 0, 4, 0, 6}

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

Step 2:

{1, 0, 16, 0, 36}

Step 3:

The final result = 53.

For example:

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

Answer: (penalty regime: 0 %)

```
import java.util.Scanner;
public class Sod{
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        System.out.print("");
        int input2=sc.nextInt();
        int[] input1=new int[input2];
        System.out.println("");
        for(int i=0;i<input2;i++)
        {
            input1[i]=sc.nextInt();
        }
        int[] dp=new int [input2];
        for(int i=0;i<input2;i++)
        {
            int number=input1[i];
            int digit=(number/(int)Math.pow(10,i))%10;
            dp[i]=(number<Math.pow(10,i))?0:digit;
        }
        int sum=0;
        for(int value:dp){
            sum+=value*value;
        }
        System.out.println(""+sum);
        sc.close();
    }
}
```

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

Passed all tests! ✓

◀ Lab-03-MCQ

Jump to...

Simple Encoded Array ▶

Dashboard / My courses / CS23333-OOPUJ-2023 / Lab-04-Classes and Objects / Lab-04-Logic Building

Status	Finished
Started	Saturday, 5 October 2024, 12:27 PM
Completed	Saturday, 5 October 2024, 1:04 PM
Duration	36 mins 37 secs

Question **1**

Correct

Marked out of 5.00

Create a Class Mobile with the attributes listed below,

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

Define a Parameterized constructor to initialize the above instance variables.

Define getter and setter methods for the attributes above.

for example : setter method for manufacturer is

void setManufacturer(String manufacturer){

this.manufacturer= manufacturer;

}

String getManufacturer(){

return manufacturer;}

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

For example:

Test	Result

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

Answer: (penalty regime: 0 %)

```
public class Mobile{
    private String manufacturer;
    private String operatingSystem;
    private int cost;
    private String color;
    public Mobile(String manufacturer,String operatingSystem,int cost,String color)
        this.manufacturer=manufacturer;
        this.operatingSystem=operatingSystem;
        this.cost=cost;
        this.color=color;
    }
    public String toString(){
        return "manufacturer = " + manufacturer+ "\noperating_system = "+operatingS
    }
    public static void main(String args[]){
        Mobile mobile=new Mobile("Redmi","Andriod",34000,"Blue");
        System.out.println(mobile.toString());
    }
}
```

	Test	Expected	Got	

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

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

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 %)

```
public class Student{
    private String name;
    private int rollNo;
    public Student(){
        this.name=null;
        this.rollNo=0;
        System.out.println("No-arg constructor is invoked");
        System.out.println("No-arg constructor is invoked");

    }
    public Student(String name){
        this.name=name;
        this.rollNo=0;
        System.out.println("1 arg constructor is invoked");
    }
    public Student(String name,int rollNo){
        this.name=name;
        this.rollNo=rollNo;
        System.out.println("2 arg constructor is invoked");
    }
    public void displayInfo(){
        System.out.println("Name ="+this.name+" , Roll no = "+this.rollNo);
    }
    public static void main(String args[]){
        Student student1=new Student();
        Student student2=new Student("Rajalakshmi");
        Student student3=new Student("Lakshmi",101);
        student1.displayInfo();
        student2.displayInfo();
        student3.displayInfo();
    }
}
```

	Test	Expected	Got	

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

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

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

Area of Circle = πr^2

Circumference = 2πr

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){
       this.radius=radius;

    }
    public void setRadius(double radius){

       this.radius=radius;

    }
    public double getRadius()    {

       return radius;

    }
    public double calculateArea()  {
       return Math.PI*radius*radius;

    }
    public double calculateCircumference()    {

       return  2*Math.PI*radius;
    }
}
class prog{
    public static void main(String[] args)  {
        Scanner sc= new Scanner(System.in);
        double r=sc.nextDouble();
        Circle circle= new Circle(r);
        System.out.printf("Area = "+String.format("%.2f\n", circle.calculateArea())
        System.out.printf("Circumference = %.2f",circle.calculateCircumference());
        sc.close();
}}

```

	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	Sunday, 6 October 2024, 7:26 PM
Completed	Sunday, 6 October 2024, 8:18 PM
Duration	52 mins 9 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 collegeNameP) {
    // initialize the instance variables
    collegeName=collegeNameP;
    }
    }

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

String studentName;
String depart;

public Student(String collegeNameP, String studentNameP,String departP) {
   // initialize the instance variables
   super(collegeNameP);
   studentName=studentNameP;
   depart=departP;

}

public String toString(){
    // return the details of the student
    return "CollegeName : "+collegeName+"\nStudentName : "+studentName+"\nDepartmer
}
}
class prog {
public static void main (String[] args) {
        Student  s1 = new Student("REC","Venkatesh","CSE");
        s1.admitted();                          // invoke the admitted() metho
        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

```
    }

    // Override the withdraw method from the parent class
    @Override
    public void withdraw(double amount) {
        // Check if the withdrawal would cause the balance to drop below $100
        if (getBalance() - amount < 100) {
            // Print a message if the minimum balance requirement is not met
            System.out.println("Minimum balance of $100 required!");
        } else {
            // Call the parent class withdraw method
            super.withdraw(amount);
        }
    }
}

class prog {

    public static void main(String[] args) {
        // Print message to indicate creation of a BankAccount object
        System.out.println("Create a Bank Account object (A/c No. BA1234) with in
        // Create a BankAccount object (A/c No. "BA1234") with initial balance of
        BankAccount BA1234 = new BankAccount("BA1234", 500);
        // Print message to indicate deposit action
        System.out.println("Deposit $1000 into account BA1234:");
        // Deposit $1000 into account BA1234
        BA1234.deposit(1000);
        // Print the new balance after deposit
       System.out.println("New balance after depositing $1000: $"+BA1234.getBalan

        // Print message to indicate withdrawal action
        System.out.println("Withdraw $600 from account BA1234:");
        // Withdraw $600 from account BA1234
        BA1234.withdraw(600);
        // Print the new balance after withdrawal
        System.out.println("New balance after withdrawing $600: $" + BA1234.getBa

        // Print message to indicate creation of another SavingsAccount object
        System.out.println("Create a SavingsAccount object (A/c No. SA1000) with
        // Create a SavingsAccount object (A/c No. "SA1000") with initial balance
        SavingsAccount SA1000 = new SavingsAccount("SA1000", 300);

        // Print message to indicate withdrawal action
        System.out.println("Try to withdraw $250 from SA1000!");
        // Withdraw $250 from SA1000 (balance falls below $100)
        SA1000.withdraw(250);
        // Print the balance after attempting to withdraw $250
        System.out.println("Balance after trying to withdraw $250: $" + SA1000.ge
    }
}
```

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

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

Passed all tests! ✓

◀ Lab-05-MCQ Jump to... Is Palindrome Number? ▶

Dashboard / My courses / CS23333-OOPUJ-2023 / Lab-06-String, StringBuffer / Lab-06-Logic Building

Status	Finished
Started	Sunday, 6 October 2024, 8:24 PM
Completed	Sunday, 6 October 2024, 9:14 PM
Duration	49 mins 38 secs

Question **1**

Correct

Marked out of 5.00

Given 2 strings input1 & input2.

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

Assumption 1:

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

Assumption 2:

Both inputs will be in lower case.

Example 1:

Input 1: apple

Input 2: orange

Output: rponlgea

Example 2:

Input 1: fruits

Input 2: are good

Output: utsroigfeda

Example 3:

Input 1: ""

Input 2: ""

Output: null

For example:

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

Answer: (penalty regime: 0 %)

```
import java.util.*;
public class S{
```

```
public class S{
    public static String solve(String a,String b){
        if((a==null||a.trim().isEmpty())&& (b==null||b.trim().isEmpty()))
        return "null";
        String combined=a+b;
        Set<Character> uniqueChars=new HashSet<>();
        for(char c:combined.toCharArray()){
            if(Character.isAlphabetic(c)){
                uniqueChars.add(c);
            }
        }
        char[] charArray=new char[uniqueChars.size()];
        int i=0;
        for(char c:uniqueChars){
            charArray[i++]=c;
        }
        Arrays.sort(charArray);
        return new StringBuilder(new String(charArray)).reverse().toString();
    }
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        String input1=sc.nextLine();
        String input2=sc.nextLine();
        System.out.println(solve(input1,input2));
    }
}
```

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

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

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 %)

```
import java.util.Scanner;
public class prog{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        String input=sc.nextLine();
        String[] words=input.split(":");
        StringBuilder result=new StringBuilder();
        for(String word:words){
            char c1=word.charAt(0);
            char c2=word.charAt(1);
            if(c1==c2){
                result.append(Character.toUpperCase(c1));
            }else{
                int pos1=c1-'a'+1;
                int pos2=c2-'a'+1;
                int diff=Math.abs(pos1-pos2);
                char newChar=(char)('a'+diff-1);
                result.append(Character.toUpperCase(newChar));
            }
        }
        System.out.println(result.toString());
    }
}
```

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

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

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

For example:

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

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

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

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

If the word to be processed is "Nice":

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

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

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

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

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

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

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

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

Expected output:

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

Example 1:

input1 = "Today is a Nice Day"

input2 = 41

output = "iNce doTday"

Example 2:

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

input2 = 39

output = "naMngo arGpes"

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

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

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

For example:

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

Answer: (penalty regime: 0 %)

```java
import java.util.*;
class prog
{
    public static void main(String []args){
        Scanner o=new Scanner(System.in);
        String s=o.nextLine();
        int n=o.nextInt();
        String result=processWords(s,n);
        System.out.println(result);
    }
    public static String processWords(String input1,int input2){
        String[] words=input1.split(" ");
        int firstIndex=(input2/10)-1;
        int secondIndex=(input2%10)-1;
        String firstWordProcessed=processWord(words[firstIndex]);
        String secondWordProcessed=processWord(words[secondIndex]);
        return firstWordProcessed+" "+secondWordProcessed;
    }
    public static String processWord(String word)
    {
        int length=word.length();
        int mid=length/2;
        String l,f;
        if(length%2==0)
        {
            f=word.substring(0,mid);
            f=new StringBuilder(f).reverse().toString();
            l=word.substring(mid);
            return f+l;
        }else{
            f=word.substring(0,mid+1);
            f=new StringBuilder(f).reverse().toString();
            l=word.substring(mid);
        }
        return f+l;
    }
    }
}
```

	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! ✓

◀ Lab-06-MCQ

Jump to...

Return second word in Uppercase ▶

Dashboard / My courses / CS23333-OOPUJ-2023 / Lab-07-Interfaces / Lab-07-Logic Building

Status	Finished
Started	Sunday, 6 October 2024, 9:25 PM
Completed	Sunday, 6 October 2024, 10:22 PM
Duration	57 mins 30 secs

Question **1**

Correct

Marked out of 5.00

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

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

RBI interface has two more methods default and static method.

default void policyNote() {

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

}

static void regulations(){

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

}

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

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

Sample Input/Output:

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

For example:

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

Answer: (penalty regime: 0 %)

```
interface RBI{
    String parentBank="RBI";
    double rateofInterest();
    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 in 2024.");
    }
}
class SBI implements RBI{
```

```
class SBI implements RBI{
    @Override
    public double rateofInterest(){
        return 7.6;
    }
}
class Karur implements RBI{
    @Override
    public double rateofInterest(){
        return 7.4;
    }
}
public class Main{
    public static void main(String[] args){
        SBI sbi=new SBI();
        Karur karur=new Karur();
        sbi.policyNote();
        RBI.regulations();
        System.out.println("SBI rate of interest: "+sbi.rateofInterest()+" per annu
        System.out.println("Karur rate of interest: "+karur.rateofInterest()+" per 
    }
}
```

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

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

Create 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

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

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

}
class College implements Football {
    private String homeTeam;
    private String visitingTeam;
    private int homeScore;
    private int visitingScore;
    public void setHomeTeam(String name){
        this.homeTeam=name;
    }

public void setVisitingTeam(String name){
        this.visitingTeam=name;
}
public void homeTeamScored(int points){
    this.homeScore=points;
}
public void visitingTeamScored(int points){
  this.visitingScore=points;
}
public void displayResult()
{
    System.out.println(homeTeam+" "+homeScore+" scored");
    System.out.println(visitingTeam+" "+visitingScore+" scored");
    if(homeScore>visitingScore){
        System.out.println(homeTeam+" is the winner!");
    }
    else if(visitingScore>homeScore){
        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){
        Scanner sc= new Scanner(System.in);
        College match=new College();
        String  homeTeam=sc.nextLine();
        String visitingTeam=sc.nextLine();
        int homeScore=sc.nextInt();
```

	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 **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 %)

```java
import java.util.Scanner;
interface Playable{
    void play();
}
class Football implements Playable{
    String name;
    public Football(String name){
        this.name=name;
    }
    @Override
    public void play(){
        System.out.println(name+" is Playing football");
    }
}
class Volleyball implements Playable{
    String name;
    public Volleyball(String name){
        this.name=name;
    }
    @Override
    public void play(){
        System.out.println(name+" is Playing volleyball");
    }
}
class Basketball implements Playable{
    String name;
    public Basketball(String name){
        this.name=name;
    }
    @Override
    public void play(){
        System.out.println(name+" is Playing basketball");
    }
}
public class Main{
    public static void main(String[] args){
        Scanner sc=new Scanner(System.in);
        String footballPlayerName=sc.nextLine();
        String volleyballPlayerName=sc.nextLine();
        String basketballPlayerName=sc.nextLine();
        Playable footballPlayer=new Football(footballPlayerName);
        Playable volleyballPlayer=new Volleyball(volleyballPlayerName);
        Playable basketballPlayer=new Basketball(basketballPlayerName);
        footballPlayer.play();
        volleyballPlayer.play();
        basketballPlayer.play();
    }
}
```

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

Passed all tests! ✓

◀ Lab-07-MCQ

Jump to...

Generate series and find Nth element ▶

Dashboard / My courses / CS23333-OOPUJ-2023 / Lab-08 - Polymorphism, Abstract Classes, final Keyword / Lab-08-Logic Building

Status	Finished
Started	Saturday, 16 November 2024, 7:50 PM
Completed	Saturday, 16 November 2024, 8:37 PM
Duration	46 mins 52 secs

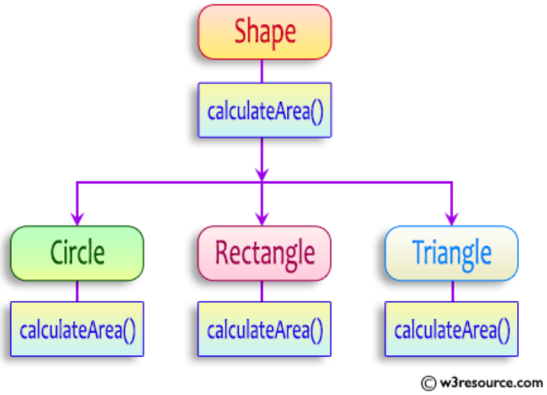
Question **1**

Correct

Marked out of 5.00

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

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



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

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

sample Input :

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

5 // length of the rectangle

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

4 // base of the triangle

3 // height of the triangle

OUTPUT:

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

For example:

Test	Input	Result

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

Answer: (penalty regime: 0 %)

```
import java.util.*;
abstract class shape{
    abstract void calculatearea();
}
class circle extends shape{
    float r;
    public circle(float radius){
        r=radius;
    }
    void calculatearea()
    {
        System.out.printf("Area of a circle: %.2f%n",(Math.PI*r*r));
    }
}
class rectangle extends shape{
    float l,b;
    public rectangle(float length,float breadth)
    {
        l=length;
        b=breadth;

    }
     void calculatearea()
     {
         System.out.printf("Area of a Rectangle: %.2f%n",(l*b));
     }
}
class triangle extends shape{
    float b,h;
    public triangle(float base,float height)
    {
        b=base;
        h=height;

    }
    void calculatearea()
    {
        System.out.printf("Area of a Triangle: %.2f%n",((0.5)*b*h));
    }
}
public class main{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        float radius,length,breadth,base,height;
        radius=s.nextFloat();
        length=s.nextFloat();
        breadth=s.nextFloat();
        base=s.nextFloat();
        height=s.nextFloat();
        circle obj1=new circle(radius);
        rectangle obj2=new rectangle(length,breadth);
```

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

Passed all tests! ✓

Question **2**

Correct

Marked out of 5.00

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

```java
class FinalExample {

    // Final variable
        final int maxSpeed = 120;

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

class SubClass extends FinalExample {

    public void displayMaxSpeed2() {
        System.out.println("Cannot override a final method");
    }

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

class prog {
    public static void main(String[] args) {
        FinalExample obj = new FinalExample();
        obj.displayMaxSpeed();
        System.out.print("\n");

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

```

	Test	Expected	Got	

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

Passed all tests! ✓

Question **3**

Correct

Marked out of 5.00

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.Scanner;
public class main{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);
        int n=s.nextInt();
        String str[]=new String[n];
        for(int i=0;i<n;i++)
        str[i]=s.next();
        int f=0;
        String str2="";
        for(int i=0;i<n;i++)
        {
            str[i]=str[i].toLowerCase();

            char ch1=str[i].charAt(0);
            char ch2=str[i].charAt(str[i].length()-1);
            if((ch1=='a'||ch1=='e'||ch1=='i'||ch1=='o'||ch1=='u')&&(ch2=='a'||ch2
            {
                f=1;
                str2+=str[i];
            }
        }
        if(f==0)
        System.out.print("no matches found");
        else
        System.out.print(str2);
    }
}
```

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

Passed all tests! ✓

◀ Lab-08-MCQ

Jump to...

FindStringCode ▶

Dashboard / My courses / CS23333-OOPUJ-2023 / Lab-09-Exception Handling / Lab-09-Logic Building

Status	Finished
Started	Saturday, 16 November 2024, 8:38 PM
Completed	Saturday, 16 November 2024, 9:10 PM
Duration	31 mins 25 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;
    // call the trynumber(n);
    trynumber(n);

  }

  public static void trynumber(int n) {
    try {
        //call the checkEvenNumber()
      checkEvenNumber(n);
      System.out.println(n + " is even.");
    } catch (ArithmeticException e) {
      System.out.println("Error: "+e.getMessage());
    }
  }

  public static void checkEvenNumber(int number) {
    if (number % 2 != 0) {
      throw new ArithmeticException(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 %)

```
import java.util.Scanner;
public class main{
    public static void main(String args[])
    {
        Scanner s=new Scanner(System.in);

        try
        {
            int n=s.nextInt();
            int a[]=new int[n];

            for(int i=0;i<a.length;i++)
            {
            a[i]=s.nextInt();
            }
            for(int i=0;i<a.length;i++)
            {
             int result=a[i]/a[i+1];
            }
        }
        catch(ArithmeticException e)
        {
            System.out.println(e);
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println(e);
        }
        finally
        finally
        {
            System.out.println("I am always executed");
        }
    }
}
```

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

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

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

          System.out.println("You entered bad data.");
     }






  }
 }
```

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

Passed all tests! ✓

◄ Lab-09-MCQ

Jump to...

The “Nambiar Number” Generator ►

Dashboard / My courses / CS23333-OOPUJ-2023 / Lab-10- Collection- List / Lab-10-Logic Building

Status	Finished
Started	Saturday, 16 November 2024, 9:10 PM
Completed	Saturday, 16 November 2024, 9:19 PM
Duration	8 mins 41 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 %)

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

    }
}
```

	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());

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

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

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

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

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

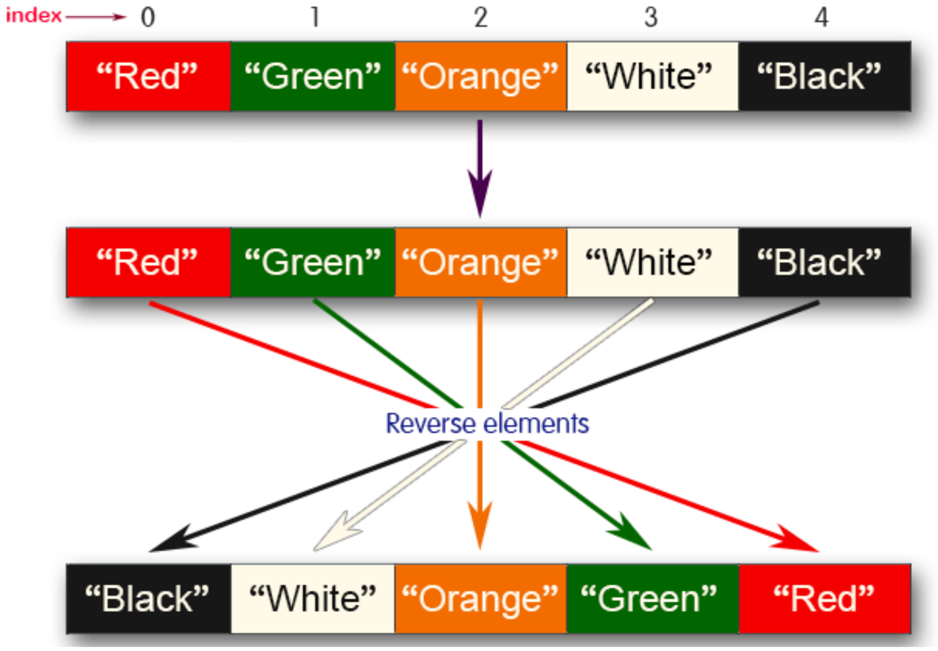
Passed all tests! ✓

Question **3**

Correct

Marked out of 1.00

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



```
Sample input and Output:
Red
Green
Orange
White
Black
Sample output
List before reversing :
[Red, Green, Orange, White, Black]
List after reversing :
[Black, White, Orange, Green, Red]
```

Answer: (penalty regime: 0 %)

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

    }
}
```

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

Passed all tests! ✓

◄ Lab-10-MCQ

Jump to...

Lab-11-MCQ ►

Dashboard / My courses / CS23333-OOPUJ-2023 / Lab-11-Set, Map / Lab-11-Logic Building

Status	Finished
Started	Sunday, 17 November 2024, 5:38 PM
Completed	Sunday, 17 November 2024, 6:01 PM
Duration	22 mins 59 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

```
import java.util.HashSet;
import java.util.Scanner;

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;
import java.util.Set;

public class CompareSets {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n1 = sc.nextInt();
        sc.nextLine();
        Set<String> set1 = new HashSet<>();
        for (int i = 0; i < n1; i++) {
            set1.add(sc.nextLine());
        }
        int n2 = sc.nextInt();
        sc.nextLine();
        Set<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		Cricket Hockey Volleyball Football	Cricket Hockey Volleyball Football	✓

		5 Football Hockey Cricket Volleyball Basketball 7 Golf Cricket Badminton Football Hockey Volleyball Throwball			
✓	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

```
import java.util.HashMap;
import java.util.Map.Entry;
import java.util.Set;
import java.util.Scanner;

class prog {
    public static void main(String[] args) {
        HashMap<String, Integer> map = new HashMap<String, Integer>();

        String name;
        int num;
        Scanner sc = new Scanner(System.in);
        int n = sc.nextInt();
        for (int i = 0; i < n; i++) {
            name = sc.next();
            num = sc.nextInt();
            map.put(name, num);
        }
        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");
        if (value != null) {
            System.out.println(value);
        } else {
            System.out.println("Key 'TWO' not found");
        }
        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 ►

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

Status	Finished
Started	Sunday, 17 November 2024, 6:01 PM
Completed	Sunday, 17 November 2024, 6:56 PM
Duration	54 mins 29 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: 00001000000000000000000001000000000001000000000100000000000001

The decoded string (original word) will be: WIPRO

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

For example:

Input	Result
010010001	ZYX
00001000000000000000000001000000000001000000000100000000000001	WIPRO

Answer: (penalty regime: 0 %)

```java
import java.util.Scanner;

public class BinaryDecoder {
    public static String decode(String input) {
        StringBuilder decodedWord = new StringBuilder();
        String[] parts = input.split("1");

        for (String part : parts) {
            int length = part.length();
            if (length > 0 && length <= 26) {
                char letter = (char) ('A' + (26 - length));
                decodedWord.append(letter);
            }
        }

        return decodedWord.toString();
    }

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

        System.out.print("");
        String input = scanner.nextLine();

        String result = decode(input);
        System.out.println("" + result);

        scanner.close();
    }
}

```

	Input	Expected	Got	
✓	010010001	ZYX	ZYX	✓
✓	00001000000000000000000100000000000100000000010000000000001	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 %)

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

public class CommonAsciiSum {
    public static int commonAsciiSingleDigitSum(char[] input1, char[] input2) {
        Set<Character> set1 = new HashSet<>();
        Set<Character> commonChars = new HashSet<>();

        for (char c : input1) {
            set1.add(c);
        }
        for (char c : input2) {
            if (set1.contains(c)) {
                commonChars.add(c);
            }
        }

        int sum1 = 0;
        for (char c : commonChars) {
            sum1 += (int) c;
        }

        while (sum1 >= 10) {
            int tempSum = 0;
            while (sum1 > 0) {
                tempSum += sum1 % 10;
                sum1 /= 10;
            }
            sum1 = tempSum;
        }

        return sum1;
    }

    public static void main(String[] args) {
        char[] input1 = {'a', 'b', 'c'};
        char[] input2 = {'b', 'c'};
        System.out.println(commonAsciiSingleDigitSum(input1, input2)); // Output:
```

```
        System.out.println(commonAsciiSingleDigitSum(input1, input2)); // Output:
    }
}

```

	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 %)

```java
import java.util.Scanner;

public class SentenceReverser {
    public static String reverseWords(String sentence, int caseOption) {
        String[] words = sentence.split(" ");
        StringBuilder reversedSentence = new StringBuilder();

        for (String word : words) {
            StringBuilder reversedWord = new StringBuilder(word).reverse();

            if (caseOption == 1) {
                for (int i = 0; i < word.length(); i++) {
                    char originalChar = word.charAt(i);
                    char reversedChar = reversedWord.charAt(i);

                    if (Character.isUpperCase(originalChar)) {
                        reversedWord.setCharAt(i, Character.toUpperCase(reversedCha
                    } else if (Character.isLowerCase(originalChar)) {
                        reversedWord.setCharAt(i, Character.toLowerCase(reversedCha
                    }
                }
            }

            if (reversedSentence.length() > 0) {
                reversedSentence.append(" ");
            }
            reversedSentence.append(reversedWord);
        }

        return reversedSentence.toString();
    }

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

        System.out.print("");
        String inputSentence = scanner.nextLine();

        System.out.print("");
        int caseOption = scanner.nextInt();

        String result = reverseWords(inputSentence, caseOption);
        System.out.println("" + result);

        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 ►

CONNECT 4 GAME

A MINI-PROJECT BY:

KARTHIKHA SRE M	230701143
KAVIYA MADHIRAJU	230701148

in partial fulfillment of the award of the degree

OF

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI

An Autonomous Institute

CHENNAI

NOVEMBER 2024

BONAFIDE CERTIFICATE

Certified that this project **“CONNECT 4 GAME”** is the Bonafide work of **“KARTHIKHA SRE M, KAVIYA MADHIRAJU”** who carried out theproject work under my supervision.

Submitted for the practical examination held on ____________________

SIGNATURE

Ms. DHARSHINI BS
Assistant Professor(SG),
Computer Science and Engineering,
Rajalakshmi Engineering College
(Autonomous),
Thandalam,Chennai-602105

SIGNATURE

Ms. V.JANANEE
Assistant Professor(SG),
Computer Science and Engineering,
Rajalakshmi Engineering College
(Autonomous),
Thandalam,Chennai-602105

INTERNAL EXAMINER

EXTERNAL EXAMINER

ABSTRACT

CONNECT 4 GAME

Connect 4 is a classic board game, reimagined as a 5x6 grid duel between a human and an AI. The objective is simple: connect four of your pieces vertically or horizontally before your opponent. Yet, beneath this simple premise lies a world of strategic depth.

The AI, driven by an adapted algorithm inspired by the Tower of Hanoi problem, is a challenging opponent. This customized algorithm is designed to assess the board, pinpoint potential winning opportunities, and counter the player's tactics. By breaking down the gameplay into smaller, manageable subproblems, the AI can predict multiple moves ahead, implementing strategies that maximize its chances of success.

By challenging players to think ahead and react to the AI's moves, Connect 4 provides a captivating glimpse into the world of artificial intelligence and strategic gameplay. The game encourages critical thinking, strategic planning, and adaptability, making it an engaging experience for players of all ages.

TABLE OF CONTENTS

1 INTRODUCTION

1.1 OVERVIEW

1.2 TECHNOLOGY STACK

1.3 PROJECT SCOPE

1.4 FEATURES

2 SYSTEM REQUIREMENTS

2.1 HARDWARE

2.2 SOFTWARE

3 SAMPLE CODE

3.1 CONTROLLER

3.2 SERVICES

3.3 DATABASE CONNECTIVITY

3.4 MAIN FUNCTION

4 VISUAL REPRESENTATIONS

4.1 GAME BOARD

4.2 WIN/LOSS SCREEN

5 CONCLUSIONS

6 REFERENCES

INTRODUCTION

1.1 OVERVIEW

This project brings the classic Connect 4 game to life using JavaFX for a visually appealing interface and MySQL for data storage. Players can compete against a computer opponent. Game results are tracked in a MySQL database, allowing users to see their past performance.

1.2 TECHNOLOGY STACK

- **JavaFX:** Creates an interactive and user-friendly interface.
- **MySQL:** Stores game data and player records.
- **JDBC (Java Database Connectivity):** Facilitates communication between the Java application and the database.

1.3 PROJECT SCOPE

This project offers a fun and engaging platform for Connect 4 enthusiasts:

- **Intuitive Interface:** Enjoy a smooth and user-friendly experience.
- **AI Opponent:** Challenge yourself in single-player mode.
- **Detailed History:** Track past games and analyze performance.
- **Player Statistics:** See your wins, losses, and overall progress.

1.4 FEATURES

- **Player Name Input:** Players can enter their name before starting the game.
- **AI Opponent:** Play against a challenging AI opponent.
- **Game History:** Store game results, including player name and outcome, in a MySQL database.

SYSTEM REQUIREMENTS

2.1 HARDWARE

- Processor: Intel i5 or equivalent
- Memory: Minimum 4GB RAM
- Hard Disk: 500 GB of free space
- Graphics Card: Capable of supporting JavaFX graphics

2.2 SOFTWARE

- Programming Language: Java
- GUI Library: JavaFX
- Database: MySQL
- Database Connectivity: JDBC (Java Database Connectivity)
- Operating System: Windows 10 or later
- IDE (Integrated Development Environment): Visual Studio Code

SAMPLE CODE

3.1 CONTROLLER

BoardController.java:

```
package lk.ijse.dep.controller;


import com.jfoenix.controls.JFXButton;


import javafx.animation.KeyFrame;
import javafx.animation.Timeline;
import javafx.animation.TranslateTransition;
import javafx.application.Platform;
import javafx.event.ActionEvent;
import javafx.scene.Group;
import javafx.scene.control.Label;
import javafx.scene.layout.AnchorPane;
import javafx.scene.layout.Pane;
import javafx.scene.layout.VBox;
import javafx.scene.shape.Circle;
import javafx.scene.shape.Rectangle;
import javafx.util.Duration;
import lk.ijse.dep.service.AiPlayer;
import lk.ijse.dep.service.Board;
import lk.ijse.dep.service.BoardImpl;
import lk.ijse.dep.service.BoardUi;
import lk.ijse.dep.service.HumanPlayer;
import lk.ijse.dep.service.Piece;
import lk.ijse.dep.service.Player;
import lk.ijse.dep.service.Winner;

public class BoardController implements BoardUi{
```

```
private static final int RADIUS = 42;

public Label lblStatus;
public Group grpCols;
public AnchorPane root;
public Pane pneOver;
public JFXButton btnPlayAgain;

private String playerName;
private boolean isAiPlaying;
private boolean isGameOver;

private Player humanPlayer;
private Player aiPlayer;

private PlayerDAO playerDAO = new PlayerDAO();
private GameHistoryDAO gameHistoryDAO = new GameHistoryDAO();

private void initializeGame() {
   Board newBoard = new BoardImpl(this);
   humanPlayer = new HumanPlayer(newBoard);
   aiPlayer = new AiPlayer(newBoard);
}

public void initialize() {
   initializeGame();
   grpCols.getChildren().stream().map(n -> (VBox) n).forEach(vbox ->
  vbox.setOnMouseClicked(mouseEvent -> colOnClick(vbox)));
}
```

```
private void colOnClick(VBox col) {
    if (!isAiPlaying && !isGameOver)
  humanPlayer.movePiece(grpCols.getChildren().indexOf(col));
}

public void initData(String playerName) {
    this.playerName = playerName;
}

@Override
public void update(int col, boolean isHuman) {
    if (isGameOver) return;
    VBox vCol = (VBox) grpCols.lookup("#col" + col);
    if (vCol.getChildren().size() == 5)
            throw new RuntimeException("Double check your logic, no space available
  within the column: " + col);
    if (!isHuman) {
        vCol.getStyleClass().add("col-ai");
    }
    Circle circle = new Circle(RADIUS);
    circle.getStyleClass().add(isHuman ? "circle-human" : "circle-ai");
    vCol.getChildren().add(0, circle);
    if (vCol.getChildren().size() == 5) vCol.getStyleClass().add("col-filled");
    TranslateTransition tt = new TranslateTransition(Duration.millis(250), circle);
    tt.setFromY(-50);
    tt.setToY(circle.getLayoutY());
    tt.playFromStart();
    lblStatus.getStyleClass().clear();
    lblStatus.getStyleClass().add(isHuman ? "ai" : "human");
    if (isHuman) {
```

```
    isAiPlaying = true;
    grpCols.getChildren().stream().map(n -> (VBox) n).forEach(vbox ->
 vbox.getStyleClass().remove("col-human"));
    KeyFrame delayFrame = new KeyFrame(Duration.millis(300), actionEvent -> {
      if (!isGameOver) lblStatus.setText("Wait, AI is playing");
    });
    KeyFrame keyFrame = new KeyFrame(Duration.seconds(0.5), actionEvent -> {
      if (!isGameOver) aiPlayer.movePiece(-1);
    });
    new Timeline(delayFrame, keyFrame).playFromStart();
  } else {
    KeyFrame delayFrame = new KeyFrame(Duration.millis(300), actionEvent -> {
      grpCols.getChildren().stream().map(n -> (VBox) n).forEach(vbox -> {
        vbox.getStyleClass().remove("col-ai");
        vbox.getStyleClass().add("col-human");
      });
    });
    new Timeline(delayFrame).playFromStart();
    isAiPlaying = false;
    lblStatus.setText(playerName + ", it is your turn now!");
  }
}

@Override
public void notifyWinner(Winner winner) {
  isGameOver = true;
  lblStatus.getStyleClass().clear();
  lblStatus.getStyleClass().add("final");

  String winnerName = "";
  String winnerType = "";
  String opponentName = "";

  switch (winner.getWinningPiece()) {
    case BLUE:
      winnerName = playerName;
      winnerType = "PLAYER";
      opponentName = "AI";
      lblStatus.setText(String.format("%s, you have won the game!", playerName));
      break;
    case PINK:
      winnerName = "AI";
      winnerType = "AI";
      opponentName = playerName;
      lblStatus.setText("Game is over, AI has won the game!");
      break;
    case EMPTY:
      winnerName = "None";
      winnerType = "TIED";
      opponentName = "None";
      lblStatus.setText("Game is tied!");
      break;
  }

  // Insert the game result into the game_history table
  gameHistoryDAO.insertGameHistory(playerName, winnerName, opponentName);

  // If the player won, update their games_won count in the players table
  if ("PLAYER".equals(winnerType)) {
    playerDAO.updateGamesWon(playerName);
    }

    // Highlight the winning area if there is a winner
    if (winner.getWinningPiece() != Piece.EMPTY) {
      VBox vCol = (VBox) grpCols.lookup("#col" + winner.getCol1());
      Rectangle rect = new Rectangle((winner.getCol2() - winner.getCol1() + 1) *
vCol.getWidth(),
              (winner.getRow2() - winner.getRow1() + 1) * (((RADIUS + 2) * 2)));
      rect.setId("rectOverlay");
      root.getChildren().add(rect);
      rect.setLayoutX(vCol.localToScene(0, 0).getX());
      rect.setLayoutY(vCol.localToScene(0, 0).getY() + (4 - winner.getRow2()) *
((RADIUS + 2) * 2));
      rect.getStyleClass().add("winning-rect");
    }

    pneOver.setVisible(true);
    pneOver.toFront();
    Platform.runLater(btnPlayAgain::requestFocus);
  }

  public void btnPlayAgainOnAction(ActionEvent actionEvent) {
    initializeGame();
    isAiPlaying = false;
    isGameOver = false;
    pneOver.setVisible(false);
    lblStatus.getStyleClass().clear();
    lblStatus.setText("LET'S PLAY !");
    grpCols.getChildren().stream().map(n -> (VBox) n).forEach(vbox -> {
      vbox.getChildren().clear();
      vbox.getStyleClass().remove("col-ai");
```

```
                vbox.getStyleClass().remove("col-filled");
                vbox.getStyleClass().add("col-human");
            });
            root.getChildren().remove(root.lookup("#rectOverlay"));
        }
    }
```

CreatePlayerController.java:

```
package lk.ijse.dep.controller;

import java.io.IOException;

import com.jfoenix.controls.JFXButton;
import com.jfoenix.controls.JFXTextField;

import javafx.application.Platform;
import javafx.event.ActionEvent;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.scene.control.Alert;
import javafx.scene.input.MouseEvent;
import javafx.scene.shape.CubicCurve;
import javafx.stage.Stage;
import lk.ijse.dep.util.DEPAlert;

public class CreatePlayerController {
    public JFXTextField txtName;
    public JFXButton btnPlay;
    public CubicCurve curve;
    private PlayerDAO playerDAO = new PlayerDAO();
    public void btnPlayOnAction(ActionEvent actionEvent) throws IOException {
        String name = txtName.getText();
        if (name.isBlank()){
            new DEPAlert(Alert.AlertType.ERROR, "Error", "Empty Name", "Name can't be empty").show();
            txtName.requestFocus();
            txtName.selectAll();
            return;
        }else if (!name.matches("[A-Za-z ]+")){
            new DEPAlert(Alert.AlertType.WARNING, "Error", "Invalid Name", "Please enter a valid name").show();
            txtName.requestFocus();
            txtName.selectAll();
            return;
        }
        playerDAO.addPlayer(name);
        Stage stage = new Stage();
        FXMLLoader fxmlLoader = new FXMLLoader(getClass().getResource("/view/Board.fxml"));
        stage.setScene(new Scene(fxmlLoader.load()));
        ((BoardController)(fxmlLoader.getController())).initData(name);
        stage.setResizable(false);
        stage.setTitle("Connect 4 Game - Player: " + name);
        stage.show();
        stage.centerOnScreen();
//        stage.setOnCloseRequest(Event::consume);
        btnPlay.getScene().getWindow().hide();
        Platform.runLater(stage::sizeToScene);
    }

    public void rootOnMouseExited(MouseEvent mouseEvent) {
        curve.setControlX2(451.8468017578125);
        curve.setControlY2(-36);
    }

    public void rootOnMouseMove(MouseEvent mouseEvent) {
        curve.setControlX2(mouseEvent.getX());
    }
}
```

GameHistoryDAO.java:

```
package lk.ijse.dep.controller;

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

import lk.ijse.dep.util.DBConnection;

public class GameHistoryDAO {

    // Insert a new game record into the game_history table
    public void insertGameHistory(String playerName, String winner, String opponentName) {
        String sql = "INSERT INTO game_history (player_name, winner, opponent_name)
VALUES (?, ?, ?)";
        try (Connection connection = DBConnection.getInstance();
             PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setString(1, playerName);
            statement.setString(2, winner);
            statement.setString(3, opponentName);
            statement.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

PlayerDAO.java:

```
package lk.ijse.dep.controller;

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

import lk.ijse.dep.util.DBConnection;

public class PlayerDAO {

    // Insert a new player into the players table
    public void addPlayer(String playerName) {
        String sql = "INSERT INTO players (name, games_won) VALUES (?, ?)";
        try (Connection connection = DBConnection.getInstance();
             PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setString(1, playerName); // Set the player name
            statement.setInt(2, 0); // Set initial games won to 0
            statement.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    // Update the number of games won for a player
    public static void updateGamesWon(String playerName) {
        String sql = "UPDATE players SET games_won = games_won + 1 WHERE name = ?";
        try (Connection connection = DBConnection.getInstance();
             PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setString(1, playerName);
            statement.executeUpdate();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

3.2 SERVICES

AiPlayer.java:

```
package lk.ijse.dep.service;

import static java.lang.Math.max;
import static java.lang.Math.min;

public class AiPlayer extends Player {
    Winner winner;
    int randomNum;
    boolean isTrue;
    public AiPlayer(Board newBoard) {
        super(newBoard);
    }

    @Override
    public void movePiece(int col1) {
        randomNum = bestMove();
        board.updateMove(randomNum, Piece.PINK);
        board.getBoardUi().update(randomNum, isTrue);
        winner = board.findWinner();
        if (winner.getWinningPiece() != Piece.EMPTY) {
            board.getBoardUi().notifyWinner(winner);
        } else {
            if (!board.exitLegalMoves()) {
                board.getBoardUi().notifyWinner(new Winner(Piece.EMPTY));
            }
        }
    }
    private int bestMove() {
        boolean isUserWinning = false;
        int winningCol = 0;
        for (int i = 0; i < 6; ++i) {
            if (board.isLegalMoves(i)) {
                int row = board.findNextAvailableSpot(i);
                board.updateMove(i, Piece.PINK);
                int heuristicVal = minimax(0, false);
                board.updateMove(i, row, Piece.EMPTY);
                if (heuristicVal == 1) {
                    return i;
                } else if (heuristicVal == -1) {
                    isUserWinning = true;
                } else {
                    winningCol = i;
                }
            }
        }
        if (isUserWinning && board.isLegalMoves(winningCol)) {
            return winningCol;
        } else {
            int j;
            do {
                j = (int) ((Math.random() * ((5 - 0) + 1)) + 0);
            } while (!board.isLegalMoves(j));

            return j;
        }
    }
    private int minimax(int depth, boolean maximizingPlayer) {
        Winner winner = board.findWinner();
        if (winner.getWinningPiece() == Piece.PINK) {
            return 1;
        } else if (winner.getWinningPiece() == Piece.PINK) {//
            return -1;
        } else if (board.exitLegalMoves() && depth != 5) {
            int heuristicVal;
            if (!maximizingPlayer) {
                int minEva= 1000;
                for (int i = 0; i < 6; ++i) {
                    if (board.isLegalMoves(i)) {
                        int row = board.findNextAvailableSpot(i);
                        board.updateMove(i, Piece.BLUE);
                        heuristicVal = minimax(depth + 1, true);
                        minEva= min(minEva,heuristicVal);
                        board.updateMove(i, row, Piece.EMPTY);
                        if (heuristicVal == -1) {
                            return minEva;
                        }
                    }
                }
            } else {
                int maxEvA = -1000;
                for (int i = 0; i < 6; ++i) {
                    if (board.isLegalMoves(i)) {
                        int row = board.findNextAvailableSpot(i);
                        board.updateMove(i, Piece.PINK);
                        heuristicVal = minimax(depth + 1, false);
                        maxEvA = max(maxEvA,heuristicVal);
                        board.updateMove(i, row, Piece.EMPTY);
                        if (heuristicVal == 1) {
                            return maxEvA;
                        }
                    }
                }
            }
            return 0;
        } else {
            return 0;
        }
    }

}
```

Board.java:

```
package lk.ijse.dep.service;
public interface Board {
    final int NUM_OF_ROWS =5;
    final int NUM_OF_COLS =6;
    public BoardUi getBoardUi();
    public int findNextAvailableSpot(int col);
    public boolean isLegalMoves(int col);
    public boolean exitLegalMoves();
    public void updateMove(int col, Piece move);
    public void updateMove(int col, int row, Piece move);
    public Winner findWinner();
}
```

BoardImpl.java:

```
package lk.ijse.dep.service;

public class BoardImpl implements Board {
    Piece winningPiece = Piece.EMPTY;
    int col1, col2, row1, row2;
    private final Piece[][] pices;
    private final BoardUi boardController;
    public BoardImpl(BoardUi boardController) {
        pices = new Piece[NUM_OF_COLS][NUM_OF_ROWS];
        this.boardController = boardController;

        for (int i = 0; i < NUM_OF_COLS; i++) {
            for (int j = 0; j < NUM_OF_ROWS; j++) {
                pices[i][j] = Piece.EMPTY;
            }
        }
    }
    @Override
    public BoardUi getBoardUi() {
        return boardController;
    }

    @Override
```

```
public int findNextAvailableSpot(int col) {
    int temp = -1;
    for (int i = 0; i < NUM_OF_ROWS; i++) {
        if (pices[col][i] == Piece.EMPTY) {
            temp = i;
            break;
        }
    }
    return temp;
}
@Override
public boolean isLegalMoves(int col) {
    int isLegal = findNextAvailableSpot(col);
    return isLegal != (-1);
}
@Override
public boolean exitLegalMoves() {
    boolean temp = false;
    for (int i = 0; i < NUM_OF_COLS; i++) {
        for (int j = 0; j < NUM_OF_ROWS; j++) {
            if (pices[i][j] == Piece.EMPTY) {
                temp = true;
                break;
            }
        }
    }
    return temp;
}
@Override
public void updateMove(int col, Piece move) {
        for (int i = 0; i < NUM_OF_ROWS; i++) {
            if (pices[col][i] == Piece.EMPTY) {
                pices[col][i] = move;
                break;
            }
        }
    }
    @Override
    public void updateMove(int col, int row, Piece move){
        pices[col][row]=move;
    }
    @Override
    public Winner findWinner() {
        Piece winningPiece ;

        for (int i = 0; i < NUM_OF_COLS; i++) {
            for (int k = 0; k < 2; k++) {
                if (pices[i][k] == pices[i][k + 1] && pices[i][k + 1] == pices[i][k + 2] && pices[i][k
+ 2] == pices[i][k + 3]) {
                    if(pices[i][k] !=Piece.EMPTY) {
                        winningPiece = pices[i][k];
                        col1 = i;
                        col2 = i;
                        row1 = k;
                        row2 = k + 3;
                        return new Winner(winningPiece, col1, row1, col2, row2);
                    }
                }
            }
        }
        for (int i = 0; i < NUM_OF_ROWS; i++) {
            for (int k = 0; k < 3; k++) {
                if (pices[k][i] == pices[k + 1][i] && pices[k + 1][i] == pices[k + 2][i] && pices[k + 2][i] == pices[k + 3][i]) {
                    if(pices[k][i] !=Piece.EMPTY) {
                        winningPiece = pices[k][i];
                        col1 = k;
                        col2 = k + 3;
                        row1 = i;
                        row2 = i;
                        return new Winner(winningPiece, col1, row1, col2, row2);
                    }
                }
            }
        }
        return new Winner(Piece.EMPTY);
    }
}
```

BoardUi.java:

```
package lk.ijse.dep.service;


public interface BoardUi {
    void update(int col, boolean isHuman);


    void notifyWinner(Winner winner);
}
```

HumanPlayer.java:

```
package lk.ijse.dep.service;
```

```
public class HumanPlayer extends Player{
    boolean isTrue;
    public HumanPlayer(Board newBoard) {
        super(newBoard);
    }
    @Override
    public void movePiece(int col1) {
        isTrue=board.isLegalMoves(col1);
        if(isTrue){
            board.updateMove(col1,Piece.BLUE);
            board.getBoardUi().update(col1,isTrue);
            Winner winner=board.findWinner();
            if (winner.getWinningPiece()!=Piece.EMPTY){
                board.getBoardUi().notifyWinner(winner);
            }else {
                if (!board.exitLegalMoves()){
                    board.getBoardUi().notifyWinner(new Winner(Piece.EMPTY));
                }
            }

        }
    }

}
```

Piece.java:

```
package lk.ijse.dep.service;

public enum Piece {
    PINK,BLUE,EMPTY;
}
```

Player.java:

```
package lk.ijse.dep.service;

public class Player {
    protected Board board;
    public Player(Board board){
        this.board=board;
    }
    public void movePiece(int col1) {}
}
```

Winner.java:

```
package lk.ijse.dep.service;

public class Winner {
    private Piece winningPice;
    private int col1;
    private int row1;
    private int col2;
    private int row2;

    public Winner(Piece winningPice, int col1,int row1, int col2,int row2){
        this.winningPice=winningPice;
        this.col1=col1;
        this.row1=row1;
        this.col2=col2;
        this.row2=row2;
    }
```

```
public Winner(Piece winningPice){
   this.winningPice=winningPice;
   this.col1=-1;
   this.col2=-1;
   this.row1=-1;
   this.row2=-1;
}
public Piece getWinningPiece() {
   return winningPice;
}
public int getCol1() {
   return col1;
}
public int getCol2() {
   return col2;
}
public int getRow1() {
   return row1;
}
public int getRow2() {
   return row2;
}
@Override
public String toString(){
   return "Winner{" +
         "winningPiece=" + winningPice +
         ", col1=" + col1 +
         ", row1=" + row1 +
         ", col2=" + col2 +
         ", row2=" + row2 +
                '}';
    }
}
```

3.3 DATABASE CONNECTIVITY

DBConnection.java:

```
package lk.ijse.dep.util;

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

public class DBConnection {
    private static final String URL = "jdbc:mysql://localhost:3306/connect4";
    private static final String USERNAME = "root";
    private static final String PASSWORD = "Karthikha*1011";

    private static Connection connection;

    public static Connection getInstance() throws SQLException {
        if (connection == null || connection.isClosed()) {
            connection = DriverManager.getConnection(URL, USERNAME, PASSWORD);
        }
        return connection;
    }

}
```

DEPAlert.java:

```
package lk.ijse.dep.util;
```

```
import javafx.scene.control.Alert;
import javafx.scene.control.ButtonType;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;

public class DEPAlert extends Alert {

    public DEPAlert(AlertType alertType, String title, String header, String message,
ButtonType... buttonTypes) {
        super(alertType, message, buttonTypes);
        setTitle(title);
        setHeaderText(header);

        String image = null;
        switch (alertType){
            case ERROR:
                image = "/asset/error.png";
                break;
            case INFORMATION:
                image = "/asset/info.png";
                break;
            case WARNING:
                image = "/asset/warning.png";
                break;
        }
        if (image !=null){
            ImageView imgView = new ImageView(new Image(image));
            imgView.setFitWidth(32);
            imgView.setFitHeight(32);
            setGraphic(imgView);}
    }
}
```

3.4 MAIN FUNCTION

AppInitializer.java:

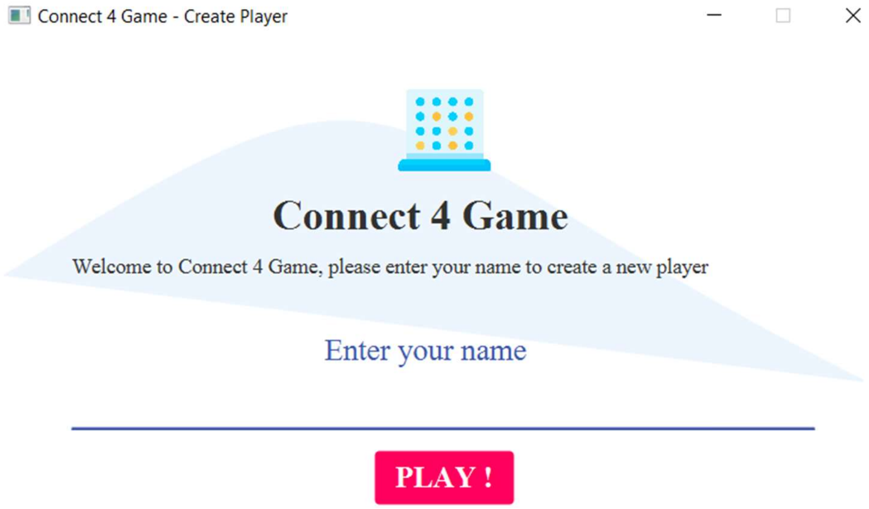
```
package lk.ijse.dep;
import java.io.IOException;
import javafx.application.Application;
import javafx.fxml.FXMLLoader;
import javafx.scene.Scene;
import javafx.stage.Stage;
public class AppInitializer extends Application {
    public static void main(String[] args) {
        launch(args);
    }
    @Override
    public void start(Stage primaryStage) throws IOException {
        primaryStage.setScene(new
Scene(FXMLLoader.load(getClass().getResource("/view/CreatePlayer.fxml"))));
        primaryStage.setResizable(false);
        primaryStage.setTitle("Connect 4 Game - Create Player");
        primaryStage.show();
        primaryStage.centerOnScreen();
    }
}
```

Launcher.java:

```
package lk.ijse.dep;
public class Launcher {
   public static void main(String[] args) {
        AppInitializer.main(args);
    }}
```

VISUAL REPRESENTATIONS

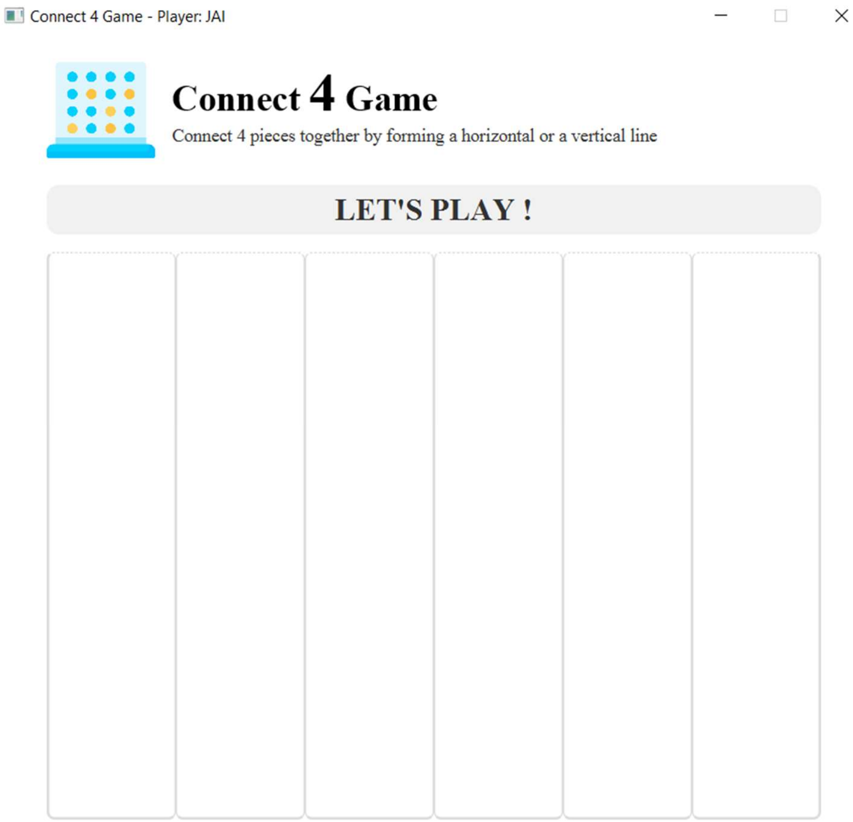
4.1 Game Board



Connect 4 Game - Create Player

Connect 4 Game

Welcome to Connect 4 Game, please enter your name to create a new player

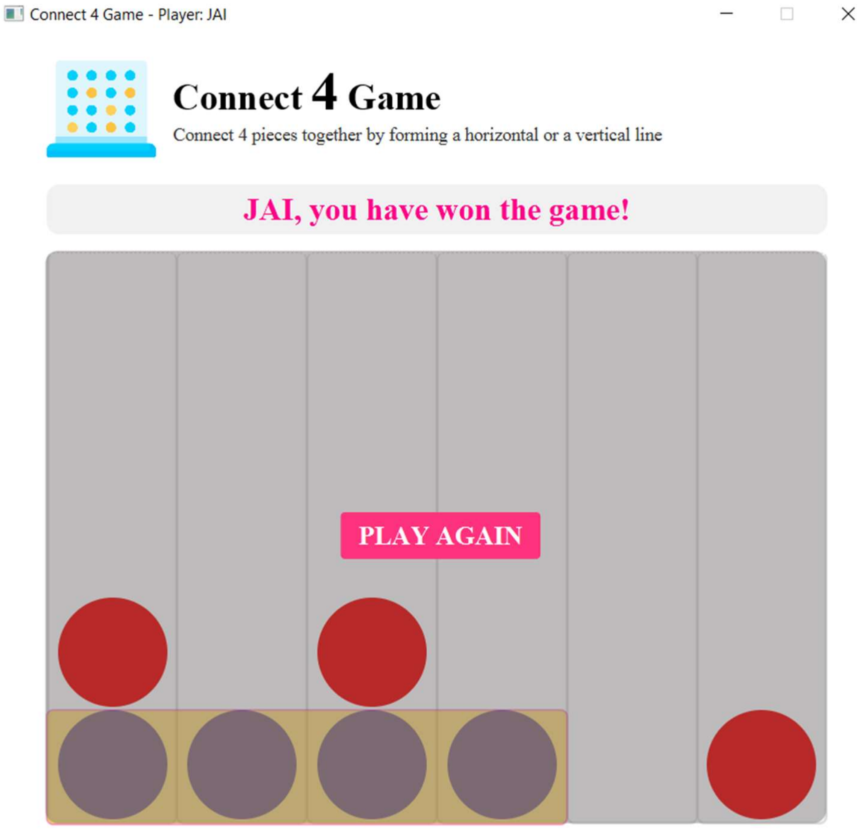


Connect 4 Game - Player: JAI

Connect 4 Game

Connect 4 pieces together by forming a horizontal or a vertical line

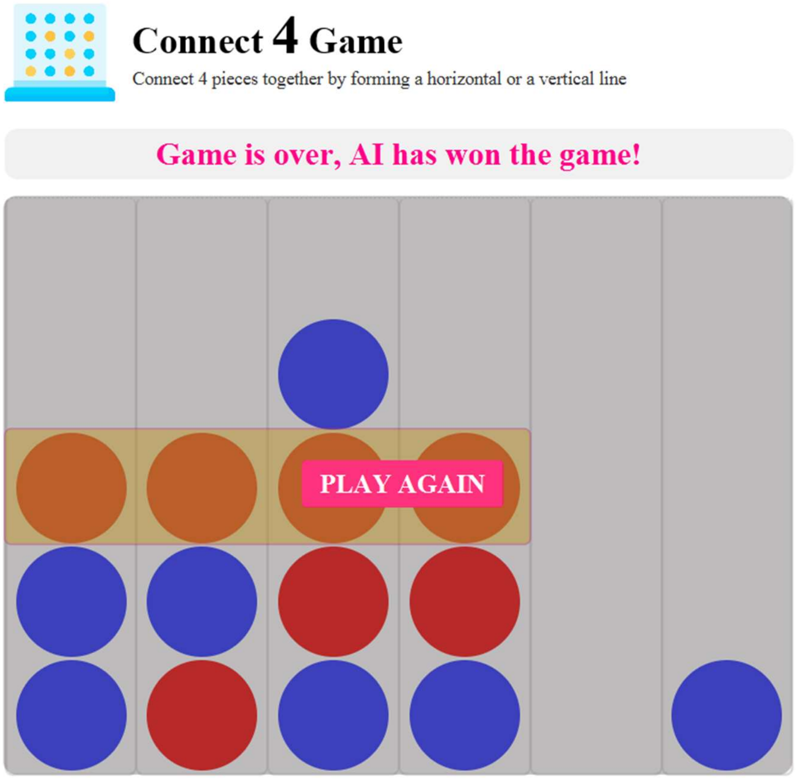
4.2 WIN/LOSS SCREEN


Connect 4 Game - Player: JAI

Connect 4 Game

Connect 4 pieces together by forming a horizontal or a vertical line

JAI, you have won the game!


Connect 4 Game

Connect 4 pieces together by forming a horizontal or a vertical line

Game is over, AI has won the game!

CONCLUSIONS

This project successfully implemented a Connect 4 game using JavaFX and MySQL. The game offers an engaging user experience with a visually appealing interface and challenging AI opponent. The integration of MySQL database allows for tracking player performance and game history.

REFERENCES

1. JavaFX Official Documentation - https://openjfx.io/
2. MySQL Documentation - https://dev.mysql.com/doc/
3. Java JDBC Tutorial - https://www.javatpoint.com/java-jdbc
4. JavaFX Tutorial - https://www.tutorialspoint.com/javafx/index.htm