

# CS23333-Object Oriented Programming Using Java-2023

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

## Quiz navigation



Show one page at a time

Finish review

**Status** Finished

**Started** Thursday, 3 October 2024, 6:43 PM

**Completed** Thursday, 3 October 2024, 7:15 PM

**Duration** 31 mins 38 secs

### Question 1

Correct

Marked out of  
5.00

[Flag question](#)

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

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

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

#### For example:

Input	Result
123	2
456	1

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

```

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

Input	Expected	Got
123	2	2
456	1	1

Passed all tests!

### Question 2

Correct

Marked out of  
5.00

[Flag question](#)

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

The last digit should be returned as a positive number.

For example,

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

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

For example:

Input	Result
197	7
-197	7

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

```

1 import java.util.*;
2 public class last
3 {
4     public static void main(String args[])
5     {
6         Scanner scan=new Scanner(System.in);
7         int a=scan.nextInt();
8         int b=a%10;
9         System.out.println(""+Math.abs(b));
10    }
11 }
```

Input	Expected	Got	
197	7	7	
-197	7	7	

Passed all tests!

### Question 3

Correct

Marked out of  
5.00

[Flag question](#)

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

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

```

1 import java.util.Scanner;
2 public class LastDigitSum{
3     public static void main(String args[]){
4         Scanner scan=new Scanner(System.in);
5
6         int num1=scan.nextInt();
7         int num2=scan.nextInt();
8         int lastdigit1=Math.abs(num1)%10;
9         int lastdigit2=Math.abs(num2)%10;
10        int sum=lastdigit1+lastdigit2;
11        System.out.println(sum);
12    }
13 }
```

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

Passed all tests!

[Finish review](#)

[◀ Lab-01-MCQ](#)

Jump to...



Is Even? ▶

# CS23333-Object Oriented Programming Using Java-2023

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

## Quiz navigation



Show one page at a time

Finish review

Status	Finished
Started	Thursday, 3 October 2024, 7:19 PM
Completed	Sunday, 6 October 2024, 11:22 AM
Duration	2 days 16 hours

### Question 1

Correct

Marked out of  
5.00

[Flag question](#)

Write a program that takes as parameter an integer n.

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

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

Note:  $n! < 10^5$

Example Input:

3

Output:

0

Example Input:

60

Output:

14

Example Input:

100

Output:

24

Example Input:

1024

Output:

253

#### For example:

Input	Result
3	0
60	14
100	24
1024	253

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

[Reset answer](#)

```

1 // Java program to count trailing 0s in n!
2 import java.io.*;
3 import java.util.Scanner;
4 class prog {
5     // Function to return trailing
6     // 0s in factorial of n
7     static int findTrailingZeros(int n)
8     {
9         if (n < 0) // Negative Number Edge Case
10             return -1;
11
12         // Initialize result
13         int count=0;
14
15
16         // Keep dividing n by powers
17         // of 5 and update count
18         for (int i = 5; n / i >= 1; i*=5)
19             count += n / i;
20
21         return count;
22     }
23
24     // Driver Code
25     public static void main(String[] args)
26     {
27         int n ;
28         Scanner sc= new Scanner(System.in);
29         int n1=sc.nextInt();
30         System.out.println( findTrailingZeros(n1) );
31     }
32 }
33

```

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

Passed all tests!

**Question 2**

Correct

Marked out of  
5.00[Flag question](#)

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

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

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

**Input Format**

Input containing an integer and a boolean value.

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

You have to print the time you should get up.

**Example Input:**

1 false

**Output:**

6:00

**Example Input:**

5 false

**Output:**

5:00

**Example Input:**

1 true

**Output:**

9:00

**For example:**

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

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

```

1 import java.util.Scanner;
2
3 public class WakeUpTime {
4     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6
7         int day = scanner.nextInt();
8         boolean vacation = scanner.nextBoolean();
9
10
11         String wakeUpTime = "";
12
13         if (vacation) {
14
15             if (day == 1 || day == 7) {
16                 wakeUpTime = "9:00";
17             } else {
18                 wakeUpTime = "7:00";
19             }
20         } else {
21
22             if (day == 1 || day == 7) {
23                 wakeUpTime = "6:00";
24             } else {
25                 wakeUpTime = "5:00";
26             }
27         }
28         System.out.println(wakeUpTime);
29     }
30 }
31

```

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 3**

Correct

Marked out of  
5.00[Flag question](#)

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

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

**Example Input:**

5

**Output:**

4

**Example Input:**

8

**Output:**

24

**Example Input:**

11

Output:  
149**For example:**

Input	Result
5	4
8	24
11	149

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

```

1 import java.util.Scanner;
2 public class Main{
3     public static int term(int n){
4         if(n==1) return 0;
5         if(n==2) return 1;
6         if(n==3) return 1;
7         int a=0,b=1,c=1;
8         for(int i=4;i<=n;i++){
9             int next=a+b+c;
10            a=b;
11            b=c;
12            c=next;
13        }
14        return c;
15    }
16    public static void main(String args[]){
17        Scanner scan= new Scanner(System.in);
18        int n=scan.nextInt();
19        int result=term(n);
20        System.out.println(result);
21    }
22 }
23 }
```

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

Passed all tests!

[Finish review](#)[◀ Lab-02-MCQ](#)

Jump to...

[Lab-03-MCQ ▶](#)



```

38     msum+=csum;
39 }
40     return pos ? msum: -1;
41 }
42 public static void main(String[] args)
43 {
44     Scanner sc = new Scanner(System.in);
45     int a=sc.nextInt();
46     int[] b=new int[a];
47     for(int i=0;i<a;i++)
48     {
49         b[i]=sc.nextInt();
50     }
51     int r=sum(a,b);
52     System.out.println(""+r);

```

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

[Flag question](#)

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

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

This is explained below:

Example 1:

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

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

Step 1:

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

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

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

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

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

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

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

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

Step 2:

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

{1, 25, 16, 49, 16}

Step 3:

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

Note:

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

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

Example 2:

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

Step 1:

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

{1, 0, 4, 0, 6}

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

Step 2:

{1, 0, 16, 0, 36}

Step 3:

The final result = 53.

For example:

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

Answer: (penalty regime: 0 %)

```

1 * import java.util.*;
2 public class sum
3 {
4     public static int cal(int a,int b[])
5     {
6         int[] r=new int[a];
7         for(int i=0;i<a;i++)
8         {
9             String n=String.valueOf(b[i]);
10            int len=n.length();
11            if(len>i)
12            {
13                r[i]=Character.getNumericValue(n.charAt(len-i-1));
14            }
15            else
16            {
17                r[i]=0;
18            }

```

```

19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
}
public static void main(String args[])
{
    Scanner sc=new Scanner(System.in);
    int a=sc.nextInt();
    int[] b=new int[a];
    for(int i=0;i<a;i++)
    {
        b[i]=sc.nextInt();
    }
    int res=cal(a,b);
    System.out.println(""+res);
}

```

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

Passed all tests!

### Question 3

Correct

Marked out of  
5.00

[Flag question](#)

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

- Find the maximum number in the array.
- Subtract the maximum number from each element of the array.
- 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	-162 0

Input	Result
-9 9	

Answer: (penalty regime: 0 %)

```

1 import java.util.*;
2 public class arr
3 {
4     public static int[] maxs(int a,int b[])
5     {
6         int max=Integer.MIN_VALUE;
7         for(int num:b)
8         {
9             if(num>max)
10            {
11                max=num;
12            }
13        }
14        for(int i=0;i<a;i++)
15        {
16            b[i]=(b[i]-max)*max;
17        }
18        return b;
19    }
20    public static void main(String[] args)
21    {
22        Scanner sc =new Scanner(System.in);
23        int[] a=sc.nextInt();
24        int[] b=new int[a];
25        for (int i=0;i<a;i++)
26        {
27            b[i]=sc.nextInt();
28        }
29        int[] r=maxs(a,b);
30        for(int num:r)
31        {
32            System.out.print(num+" ");
33        }
34        sc.close();
35    }
36 }
```

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!

[Finish review](#)

[◀ Lab-03-MCQ](#)

Jump to...



[Simple Encoded Array ▶](#)

# CS23333-Object Oriented Programming Using Java-2023

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

## Quiz navigation



Show one page at a time

Finish review

Status Finished

Started Sunday, 6 October 2024, 6:14 PM

Completed Sunday, 6 October 2024, 6:21 PM

Duration 6 mins 57 secs

Question 1

Correct

Marked out of  
5.00

[Flag question](#)

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

**Area of Circle =  $\pi r^2$**

**Circumference =  $2\pi r$**

**Input:**

2

**Output:**

**Area = 12.57**

**Circumference = 12.57**

**For example:**

Test	Input	Result
1	4	Area = 50.27 Circumference = 25.13

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

[Reset answer](#)

```

1 import java.io.*;
2 import java.util.Scanner;
3 class Circle
4 {
5     private double radius;
6     public Circle(double radius){
7         // set the instance variable radius
8         this.radius =radius;
9     }
10    public void setRadius(double radius){
11        // set the radius
12        this.radius=radius;
13    }
14    public double getRadius()    {
15        // return the radius
16        return radius;
17    }
18    public double calculateArea() { // complete the below statement
19        return Math.PI*radius*radius;
20    }
21    public double calculateCircumference()    {
22        // complete the statement
23        return 2*Math.PI*radius;
24    }
25 }
26 class prog{
27     public static void main(String[] args) {
28         int r;
29         Scanner sc= new Scanner(System.in);
30         r=sc.nextInt();
31         Circle c= new Circle(r);
32         System.out.println("Area = "+String.format("%.2f", c.calculateArea()));
33         // invoke the calculatecircumference method
34         System.out.println("Circumference = "+String.format("%.2f" , c.calculateCircumference()));
35         sc.close();
36     }
37 }
38
39
40
41
42
43
44

```

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!

Question 2

Correct

Marked out of  
5.00

[Flag question](#)

Create a Class Mobile with the attributes listed below,

```

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

```

Define a Parameterized constructor to initialize the above instance variables.

Define getter and setter methods for the attributes above.

for example : setter method for manufacturer is

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

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

**For example:**

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

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

```
1 public class mobile{  
2     private String man;  
3     private String os;  
4     public String clr;  
5     private int cost;  
6     public mobile(String man,String os,String clr,int cost){  
7         this.man=man;  
8         this.os=os;  
9         this.clr=clr;  
10        this.cost=cost;  
11    }  
12    public String toString(){  
13        return "manufacturer = "+man+"\n"+ "operating_system = "+os+"\n"+ "color = "+ clr+"\n"+ "cost = "+cost;  
14    }  
15    public static void main(String[]args){  
16        mobile mobile=new mobile("Redmi","Andriod","Blue",34000);  
17        System.out.println(mobile);  
18    }  
19 }  
20 }
```

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

Passed all tests!

Question 3  
Correct  
Marked out of 5.00  
[Flag question](#)

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

Student()

Student(String name)

Student(String name, int rollno)

**Input:**

No input

**Output:**

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

**For example:**

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

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

```
1 public class stud{  
2     private String name;  
3     private int roll;  
4     public stud(){  
5         System.out.println("No-arg constructor is invoked");  
6         name=null;  
7         roll=0;  
8     }  
9     public stud(String name){  
10        System.out.println("1 arg constructor is invoked");  
11        this.name=name;  
12        roll=0;  
13    }  
14    public stud(String name,int roll){  
15        System.out.println("2 arg constructor is invoked");  
16        this.name=name;  
17        this.roll=roll;  
18    }  
19 }  
20 }
```

```

22
23 +
24     public static void main (String[]args){
25         stud s1=new stud();
26         stud s2=new stud("Rajalakshmi");
27         stud s3=new stud("Lakshmi",101);
28         System.out.println("Name =" +s1.name+ " , Roll no = "+s2.roll);
29         System.out.println("Name =" +s2.name+ " , Roll no = "+s2.roll);
30         System.out.println("Name =" +s3.name+ " , Roll no = "+s3.roll);
31     }
32 }
```

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

Passed all tests!

[Finish review](#)

[◀ Lab-04-MCQ](#)

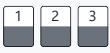
Jump to...

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

# CS23333-Object Oriented Programming Using Java-2023

Dashboard / My courses / CS23333-OOPUJ-2023 / Lab-05-Inheritance / Lab-05-Logic Building

## Quiz navigation



Show one page at a time

Finish review

Status	Finished
Started	Sunday, 6 October 2024, 6:24 PM
Completed	Sunday, 6 October 2024, 6:39 PM
Duration	15 mins 2 secs

### Question 1

Correct

Marked out of  
5.00

[Flag question](#)

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

College:

```
String collegeName;
public College() {}
```

```
public admitted() {}
```

Student:

```
String studentName;
```

```
String department;
```

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

```
public toString()
```

Expected Output:

A student admitted in REC

CollegeName : REC

StudentName : Venkatesh

Department : CSE

**For example:**

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

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

[Reset answer](#)

```
1 class College
2 {
3     public String collegeName;
4
5     public College(String collegeName) {
6         // initialize the instance variables
7         this.collegeName=collegeName;
8     }
9
10    public void admitted() {
11        System.out.println("A student admitted in "+collegeName);
12    }
13 }
14 class Student extends College{
15
16     String studentName;
17     String department;
18
19     public Student(String collegeName, String studentName, String department) {
20         // initialize the instance variables
21         super(collegeName);
22         this.studentName=studentName;
23         this.department=department;
24     }
25
26     public String toString(){
27         // return the details of the student
28         return "CollegeName : "+collegeName+"\n"+ "StudentName : "+studentName+"\n"+ "Department : "+department;
29     }
30 }
31
32 public class Main {
33     public static void main (String[] args) {
34         Student s1 = new Student("REC", "Venkatesh", "CSE");
35         s1.admitted();                                // invoke the admitted() method
36         System.out.println(s1.toString());
37     }
38 }
```

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 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().

[Flag question](#)

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

```

class Mobile{

}
class CameraMobile extends Mobile {
}
class AndroidMobile extends CameraMobile {
}
expected output:
Basic Mobile is Manufactured
Camera Mobile is Manufactured
Android Mobile is Manufactured
Camera Mobile with 5MG px
Touch Screen Mobile is Manufactured

```

**For example:**

**Result**

```

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

```

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

```

1 v class mob{
2 v     mob(){
3 v         System.out.println("Basic Mobile is Manufactured");
4 v     }
5 v     void basmob(){
6 v         System.out.println("Basic Mobile is Manufactured");
7 v     }
8 v }
9 v class cam extends mob{
10 v    cam(){
11 v        super();
12 v        System.out.println("Camera Mobile is Manufactured");
13 v    }
14 v    void newm(){
15 v        System.out.println("Camera Mobile with 5MG px");
16 v    }
17 v }
18 v class and extends cam{
19 v    and(){
20 v        super();
21 v        System.out.println("Android Mobile is Manufactured");
22 v    }
23 v    void andmob(){
24 v        System.out.println("Touch Screen Mobile is Manufactured");
25 v    }
26 v }
27 v
28 v public class Main{
29 v     public static void main(String[]args){
30 v         and andmob=new and();
31 v         andmob.newm();
32 v         andmob.andmob();
33 v     }
34 v }
35 v
36 v

```

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

Passed all tests!

Question 3

Correct

Marked out of  
5.00

[Flag question](#)

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

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

**For example:**

**Result**

```

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

```

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

[Reset answer](#)

```

1 v class BankAccount {
2 v     // Private field to store the account number
3 v     private String accountNumber;
4 v
5 v     // Private field to store the balance
6 v     private double balance;
7 v
8 v     // Constructor to initialize account number and balance
9 v     public BankAccount(String accountNumber,double balance){

```

```

10     this.accountNumber=accountNumber;
11     this.balance=balance;
12 }
13
14
15
16
17 // Method to deposit an amount into the account
18 public void deposit(double amount) {
19     // Increase the balance by the deposit amount
20     balance+=amount;
21 }
22
23 // Method to withdraw an amount from the account
24 public void withdraw(double amount) {
25     // Check if the balance is sufficient for the withdrawal
26     if (balance >= amount) {
27         // Decrease the balance by the withdrawal amount
28         balance -= amount;
29     } else {
30         // Print a message if the balance is insufficient
31         System.out.println("Insufficient balance");
32     }
33 }
34
35 // Method to get the current balance
36 public double getBalance() {
37     // Return the current balance
38     return balance;
39 }
40 public String getAccountNumber(){
41     return accountNumber;
42 }
43 }
44 class SavingsAccount extends BankAccount {
45     // Constructor to initialize account number and balance
46     public SavingsAccount(String accountNumber, double balance) {
47         // Call the parent class constructor
48         super(accountNumber,balance);
49     }
50
51     // Override the withdraw method from the parent class
52     @Override

```

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 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 initia Try to withdraw \$250 from SA1000! Minimum balance of \$100 required! Balance after trying to withdraw \$250: \$300.0

Passed all tests!

Finish review

◀ Lab-05-MCQ

Jump to...

Is Palindrome Number? ▶

# CS23333-Object Oriented Programming Using Java-2023

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

## Quiz navigation



Show one page at a time

Finish review

Status	Finished
Started	Sunday, 6 October 2024, 6:44 PM
Completed	Sunday, 6 October 2024, 7:20 PM
Duration	35 mins 12 secs

### Question 1

Correct

Marked out of  
5.00

[Flag question](#)

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

For example:

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

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

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

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

If the word to be processed is "Nice":

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

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

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

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

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

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

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

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

Expected output:

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

Example 1:

```
input1 = "Today is a Nice Day"
```

```
input2 = 41
```

```
output = "iNce doTday"
```

Example 2:

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

```
input2 = 39
```

```
output = "naMngO arGpes"
```

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

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

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

For example:

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

Answer: (penalty regime: 0 %)

```

1 import java.util.*;
2 public class mix{
3     public static void main(String[] args){
4         Scanner scan = new Scanner(System.in);
5         String g = scan.nextLine();
6         int n = scan.nextInt(),ones,flag = 0;
7         StringBuffer temp = new StringBuffer();
8         StringBuffer temp1 = new StringBuffer();
9         int space = 0;
10        while (n > 0){
11            ones = (n % 10) - 1;
12            for(int i = 0; i < g.length();i++){
13                if (g.charAt(i) == ' '){
14                    space = space + 1;
15                }
16                else if(space == ones && flag == 0){
17                    temp.append(Character.toString(g.charAt(i)));
18                }
19                else if(space == ones && flag == 1){
20                    temp1.append(Character.toString(g.charAt(i)));
21                }
22            }
23            space = 0 ;
24            flag = 1;
25            n = n /10;
26        }
27        rew m = new rew();
28        System.out.println(m.r(temp1.toString()) + " " + m.r(temp.toString()));
29    }
30 }
31 class rew{
32     String r(String a){
33         int le = a.length(),n,q;
34         StringBuffer temp3 = new StringBuffer();
35         if(le % 2 == 1){
36             n = ((int)(le/2));
37             q = ((int)(le/2));
38         }
39         else{

```

```

40         n = ((int)(le/2)) - 1;
41         q = ((int)(le/2));
42     }
43     for(int i = n;i >= 0;i--){
44         temp3.append(Character.toString(a.charAt(i)));
45     }
46     for(int i = q;i < le;i++){
47         temp3.append(Character.toString(a.charAt(i)));
48     }
49     return temp3.toString();
50 }
51 }
52

```

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

Passed all tests!

### Question 2

Correct

Marked out of  
5.00

[Flag question](#)

Given a String input1, which contains many number of words separated by : and each word contains exactly two lower case alphabets, generate an output based upon the below 2 cases.

Note:

1. All the characters in input 1 are lowercase alphabets.
2. input 1 will always contain more than one word separated by :
3. Output should be returned in uppercase.

Case 1:

Check whether the two alphabets are same.

If yes, then take one alphabet from it and add it to the output.

Example 1:

```
input1 = ww:ii:pp:rr:oo
output = WIPRO
```

Explanation:

word1 is ww, both are same hence take w

word2 is ii, both are same hence take i

word3 is pp, both are same hence take p

word4 is rr, both are same hence take r

word5 is oo, both are same hence take o

Hence the output is WIPRO

Case 2:

If the two alphabets are not same, then find the position value of them and find maximum value – minimum value.

Take the alphabet which comes at this (maximum value - minimum value) position in the alphabet series.

Example 2"

```
input1 = zx:za:ee
output = BYE
```

Explanation

word1 is zx, both are not same alphabets

position value of z is 26

position value of x is 24

max – min will be 26 – 24 = 2

Alphabet which comes in 2<sup>nd</sup> position is b

Word2 is za, both are not same alphabets

position value of z is 26

position value of a is 1

max – min will be 26 – 1 = 25

Alphabet which comes in 25<sup>th</sup> position is y

word3 is ee, both are same hence take e

Hence the output is BYE

For example:

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

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

```

1 * import java.util.*;
2 * class diff{
3 *     char different(char a, char b){
4 *         if ((int)a != (int)b)
5 *             return (char)((int)'a' + ((int)a-(int)b) - 1);
6 *         return a;
7 *     }
8 *
9 * public class Main{
10 *     public static void main(String[] args){
11 *         Scanner scan = new Scanner(System.in);
12 *         diff z = new diff();
13 *         String q = scan.nextLine();
14 *         StringBuffer ans = new StringBuffer();
15 *         StringBuffer temp = new StringBuffer();
16 *         for(int i = 0;i < q.length();i++){
17 *             if(q.charAt(i) == ':'){

```

```

18         temp.append(" ");
19     }
20     else{
21         temp.append(Character.toString(q.charAt(i)));
22     }
23 }
24 String h = temp.toString();
25 for(int i = 0;i < temp.length();i++){
26     if(i%3 == 0){
27         ans.append(Character.toString(z.different(h.charAt(i),h.charAt(i+1))));
28     }
29 }
30 System.out.print(ans.toString().toUpperCase());
31 }
32 }
33 }
34 }

```

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

[Flag question](#)

Given 2 strings input1 & input2.

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

Assumption 1:

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

Assumption 2:

Both inputs will be in lower case.

Example 1:

Input 1: apple

Input 2: orange

Output: rponlgea

Example 2:

Input 1: fruits

Input 2: are good

Output: utsroigfeda

Example 3:

Input 1: ""

Input 2: ""

Output: null

**For example:**

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

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

```

1 import java.util.*;
2
3
4 public class HelloWorld {
5     public static void main(String[] args) {
6         Scanner scan = new Scanner(System.in);
7         String a = scan.nextLine();
8         String b = scan.nextLine();
9         StringBuffer ab = new StringBuffer();
10        if(a.trim().isEmpty() && b.trim().isEmpty()){
11            System.out.print("null");
12        }
13        else{
14            for(int i = 0;i < a.length();i++){
15                if (a.charAt(i) != ' '){
16                    ab.append(Character.toString(a.charAt(i)));
17                }
18            }
19            for(int i = 0;i < b.length();i++){
20                if (b.charAt(i) != ' '){
21                    ab.append(Character.toString(b.charAt(i)));
22                }
23            }
24            char[] d = ab.toString().toCharArray();
25            Arrays.sort(d);
26            for(int i = d.length - 1;i >= 1;i--){
27                if(d[i] != d[i-1])
28                    System.out.print(d[i]);
29            }
30            System.out.print(d[0]);
31        }
32    }
33 }
34 }
35

```

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

Passed all tests!

[Finish review](#)

[◀ Lab-06-MCQ](#)

Jump to...



[Return second word in Uppercase ▶](#)

# CS23333-Object Oriented Programming Using Java-2023

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

## Quiz navigation



Show one page at a time

Finish review

Status	Finished
Started	Sunday, 6 October 2024, 7:20 PM
Completed	Monday, 7 October 2024, 10:30 AM
Duration	15 hours 10 mins

### Question 1

Correct

Marked out of  
5.00

[Flag question](#)

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

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

RBI interface has two more methods default and static method.

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

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

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

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

#### Sample Input/Output:

**RBI has a new Policy issued in 2023**

**RBI has updated new regulations in 2024.**

**SBI rate of interest: 7.6 per annum.**

**Karur rate of interest: 7.4 per annum.**

#### For example:

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

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

```
1  interface RBI {
2      String parentBank = "RBI";
3
4      double rateOfInterest();
5
6      default void policyNote() {
7          System.out.println("RBI has a new Policy issued in 2023");
8      }
9
10     static void regulations() {
11         System.out.println("RBI has updated new regulations in 2024.");
12     }
13
14 }
15
16 class SBI implements RBI {
17
18     public double rateOfInterest() {
19         return 7.6;
20     }
21
22 }
23
24 class Karur implements RBI {
25
26     public double rateOfInterest() {
27         return 7.4;
28     }
29
30 }
31
32 public class Main {
33     public static void main(String[] args) {
34
35         RBI rbi = new SBI();
36         rbi.policyNote();
37         rbi.regulations();
38
39         SBI sbi = new SBI();
40         System.out.println("SBI rate of interest: " + sbi.rateOfInterest() + " per annum.");
41
42         Karur karur = new Karur();
43         System.out.println("Karur rate of interest: " + karur.rateOfInterest() + " per annum.");
44     }
45
46 }
47
48
49
50
51
52 }
```

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

Question 2  
Correct  
Marked out of 5.00  
[Flag question](#)

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:**

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

**For example:**

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

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

```
1 v import java.util.Scanner;
2
3
4 v interface Playable {
5     // Abstract method to play the respective sport
6     void play();
7 }
8
9
10 v class Football implements Playable {
11     String name;
12
13
14 v     public Football(String name) {
15         this.name = name;
16     }
17
18 v     public void play() {
19         System.out.println(name + " is Playing football");
20     }
21 }
22
23
24 v class Volleyball implements Playable {
25     String name;
26
27
28 v     public Volleyball(String name) {
29         this.name = name;
30     }
31
32
33 v     public void play() {
34         System.out.println(name + " is Playing volleyball");
35     }
36 }
37
38 // Basketball class implementing Playable interface
39 v class Basketball implements Playable {
40     String name;
41
42
43 v     public Basketball(String name) {
44         this.name = name;
45     }
46
47 v     public void play() {
48         System.out.println(name + " is Playing basketball");
49     }
50 }
51
52
```

Test	Input	Expected	Got
1	Sadhwini Sanjay Sruthi	Sadhwini is Playing football Sanjay is Playing volleyball Sruthi is Playing basketball	Sadhwini 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

Question 3  
Correct

Create interfaces shown below.

```
interface Sports {
```

```
public void setHomeTeam(String name);
public void setVisitingTeam(String name);
}

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

create a class College that implements the Football interface and provides the necessary functionality to the abstract methods.
```

sample Input:

```
Rajalakshmi
Saveetha
22
21
```

Output:

```
Rajalakshmi 22 scored
Saveetha 21 scored
Rajalakshmi is the Winner!
```

**For example:**

Test	Input	Result
1	Rajalakshmi Saveetha 22 21	Rajalakshmi 22 scored Saveetha 21 scored Rajalakshmi is the winner!

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

[Reset answer](#)

```
1 * import java.util.Scanner;
2
3 * interface Sports {
4     void setHomeTeam(String name);
5     void setVisitingTeam(String name);
6 }
7
8 * interface Football extends Sports {
9     void homeTeamScored(int points);
10    void visitingTeamScored(int points);
11 }
12
13 * class College implements Football {
14     private String homeTeam;
15     private String visitingTeam;
16     private int homeTeamPoints = 0;
17     private int visitingTeamPoints = 0;
18
19     public void setHomeTeam(String name) {
20         this.homeTeam = name;
21     }
22
23     public void setVisitingTeam(String name) {
24         this.visitingTeam = name;
25     }
26
27     public void homeTeamScored(int points) {
28         homeTeamPoints += points;
29         System.out.println(homeTeam + " " + points + " scored");
30     }
31
32     public void visitingTeamScored(int points) {
33         visitingTeamPoints += points;
34         System.out.println(visitingTeam + " " + points + " scored");
35     }
36
37     public void winningTeam() {
38         if (homeTeamPoints > visitingTeamPoints) {
39             System.out.println(homeTeam + " is the winner!");
40         } else if (homeTeamPoints < visitingTeamPoints) {
41             System.out.println(visitingTeam + " is the winner!");
42         } else {
43             System.out.println("It's a tie match.");
44         }
45     }
46 }
47
48 * public class Main {
49     public static void main(String[] args) {
50         Scanner sc = new Scanner(System.in);
51
52     }
53 }
```

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!

[Finish review](#)

# CS23333-Object Oriented Programming Using Java-2023

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

## Quiz navigation



Show one page at a time

Finish review

Status Finished

Started Monday, 21 October 2024, 9:25 PM

Completed Tuesday, 22 October 2024, 6:35 PM

Duration 21 hours 10 mins

### Question 1

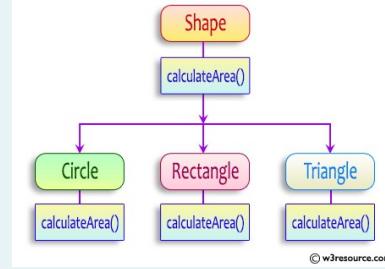
Correct

Marked out of  
5.00

Flag question

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

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



```

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

```

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

sample Input :

```

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

```

#### OUTPUT:

```

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

```

#### For example:

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

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

```

1 import java.util.Scanner;
2
3 // Abstract class Shape
4 abstract class Shape {
5     public abstract double calculateArea();
6 }
7
8 // Circle class
9 class Circle extends Shape {
10     private double radius;
11
12     public Circle(double radius) {
13         this.radius = radius;
14     }
15
16     @Override
17     public double calculateArea() {
18         return Math.PI * radius * radius; // Area of circle: πr²
19     }
20 }
21
22 // Rectangle class
23 class Rectangle extends Shape {
24     private double length;
25     private double breadth;
26
27     public Rectangle(double length, double breadth) {
28         this.length = length;
29         this.breadth = breadth;
30     }
31
32     @Override
33     public double calculateArea() {
34         return length * breadth; // Area of rectangle: length * breadth
35     }
36 }
37
38 // Triangle class
39 class Triangle extends Shape {

```

```

40     private double base;
41     private double height;
42
43     public Triangle(double base, double height) {
44         this.base = base;
45         this.height = height;
46     }
47
48     @Override
49     public double calculateArea() {
50         return 0.5 * base * height; // Area of triangle: 0.5 * base * height
51     }
52 }

```

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

Passed all tests!

### Question 2

Correct

Marked out of  
5.00

[Flag question](#)

## 1. Final Variable:

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

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

## 2. Final Method:

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

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

## 3. Final Class:

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

Given a Java Program that contains the bug in it, your task is to clear the bug to the output.  
you should delete any piece of code.

For example:

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

Answer: (penalty regime: 0 %)

[Reset answer](#)

```

1 v final class FinalExample {
2     // Final variable
3     final int MAX_SPEED = 120; // Constant value
4
5     // Final method
6     public final void display() {
7         System.out.println("The maximum speed is: " + MAX_SPEED + " km/h");
8     }
9 }
10
11 // Main class to test the final class
12 public class Test {
13     public static void main(String[] args) {
14         // Create an instance of FinalExample
15         FinalExample example = new FinalExample();
16         example.display();
17
18         // Uncommenting the following line will result in a compile-time error
19         // because FinalExample is a final class and cannot be subclassed.
20         // class SubclassExample extends FinalExample { }
21
22         System.out.println("This is a subclass of FinalExample.");
23     }
24 }
25
26

```

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

Passed all tests!

**Question 3**

Correct

Marked out of  
5.00[Flag question](#)

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

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

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

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

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

input2: String array.

Example 1:

input1: 3

input2: ("oreo", "sirish", "apple")

output: oreoapple

Example 2:

input1: 2

input2: ("Mango", "banana")

output: no matches found

Explanation:

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

Hence the output is no matches found.

Example 3:

input1: 3

input2: ("Ate", "Ace", "Girl")

output: ateace

**For example:**

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

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

```

1 import java.util.Scanner;
2
3 public class VowelStringExtractor {
4
5     // Method to extract strings with vowels as first and last characters
6     public static String extractVowelStrings(String[] stringArray) {
7         StringBuilder result = new StringBuilder();
8         String vowels = "aeiouAEIOU"; // String containing all vowels
9
10        // Iterate through the array of strings
11        for (String s : stringArray) {
12            // Check if the string is not empty and if both the first and last characters are vowels
13            if (s.length() > 0 && vowels.indexOf(s.charAt(0)) != -1 && vowels.indexOf(s.charAt(s.length() - 1)) != -1)
14                result.append(s); // Append matching string to the result
15        }
16
17        // Return the concatenated string in lowercase or "no matches found"
18        return result.length() > 0 ? result.toString().toLowerCase() : "no matches found";
19    }
20
21
22    public static void main(String[] args) {
23        Scanner scanner = new Scanner(System.in);
24
25        // Input for the number of strings
26
27        int n = scanner.nextInt();
28        scanner.nextLine(); // Consume the newline character
29
30        // Input for the strings in one line
31
32        String input = scanner.nextLine();
33        String[] strings = input.split(" "); // Split input into an array
34
35        // Process and output the result
36        String result = extractVowelStrings(strings);
37        System.out.println(result);
38
39        scanner.close(); // Close the scanner
40    }
41 }
42

```

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!

[Finish review](#)



# CS23333-Object Oriented Programming Using Java-2023

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

## Quiz navigation



Show one page at a time

Finish review

Status	Finished
Started	Monday, 18 November 2024, 6:42 PM
Completed	Monday, 18 November 2024, 7:07 PM
Duration	25 mins 28 secs

### Question 1

Correct

Marked out of  
5.00

Flag question

Write a Java program to handle `ArithmaticException` and `ArrayIndexOutOfBoundsException`.

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

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

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

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

#### Input:

```
5
10 0 20 30 40
```

#### Output:

```
java.lang.ArithmaticException: / by zero
```

I am always executed

#### Input:

```
3
10 20 30
```

#### Output:

```
java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3
```

I am always executed

#### For example:

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

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

```
1 * import java.util.*;
2 * public class prog{
3 *     public static void main(String args[])
4 *     {
5 *         Scanner obj=new Scanner(System.in);
6 *         int n=obj.nextInt();
7 *         int[]arr=new int[n];
8 *         try{
9 *             for(int i=0;i<n;i++)
10 *             {
11 *                 arr[i]=obj.nextInt();
12 *             }
13 *             int temp=arr[0]/arr[1];
14 *             int t=arr[n];
15 *         }
16 *         catch(Exception e)
17 *         {
18 *             System.out.println(e);
19 *         }
20 *         finally{
21 *             System.out.println("I am always executed");
22 *         }
23 *     }
24 * }
```

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

Passed all tests!

### Question 2

Correct

Marked out of  
5.00

Flag question

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**

```

1 * import java.util.Scanner;
2 * import java.util.InputMismatchException;
3 * class prog {
4 *     public static void main(String[] args) {
5 *         Scanner sc = new Scanner(System.in);
6 *         int length = sc.nextInt();
7 *         // create an array to save user input
8 *         int[] name = new int[length];
9 *         int sum=0;//save the total sum of the array.
10 *
11 *         /* Define try-catch block to save user input in the array "name"
12 *          If there is an exception then catch the exception otherwise print
13 *          the total sum of the array. */
14 *         try
15 *         {
16 *             for(int i=0;i<length;i++)
17 *             {
18 *                 name[i]=sc.nextInt();
19 *             }
20 *             for(int i=0;i<length;i++)
21 *             {
22 *                 sum+=name[i];
23 *             }
24 *             System.out.print(sum);
25 *         }
26 *         catch(Exception e)
27 *         {
28 *             System.out.print("You entered bad data.");
29 *         }
30 *         finally{
31 *             sc.close();
32 *         }
33 *
34 *
35 *
36 *
37 *
38 *
39 *
40 *
41 *     }
42 * }
```

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

Passed all tests!

**Question 3**

Correct

Marked out of  
5.00[Flag question](#)

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

**Sample input and Output:**

82 is even.

Error: 37 is odd.

Fill the preloaded answer to get the expected output.

**For example:**

Result
82 is even. Error: 37 is odd.

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

```

1 * class prog {
2 *     public static void main(String[] args) {
3 *         int n = 82;
4 *         trynumber(n);
5 *         n = 37;
6 *         // call the trynumber(n);
7 *         trynumber(n);
8 *
9 *     }
10 *
11 *     public static void trynumber(int n) {
12 *         try {
13 *             //call the checkEvenNumber()
14 *             checkEvenNumber(n);
15 *             System.out.println(n + " is even.");
16 *         } catch (Exception e) {
17 *             System.out.println("Error: " + e.getMessage());
18 *         }
19 *     }
}
```

```
20
21+ public static void checkEvenNumber(int number) {
22+     if (number % 2 != 0) {
23+         throw new RuntimeException (number + " is odd.");
24+     }
25+
26}
27
28|
```

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

Passed all tests!

[Finish review](#)

[◀ Lab-09-MCQ](#)

Jump to...



[The “Nambiar Number” Generator ►](#)

# CS23333-Object Oriented Programming Using Java-2023

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

## Quiz navigation



Show one page at a time

Finish review

**Status** Finished  
**Started** Monday, 18 November 2024, 6:44 PM  
**Completed** Monday, 18 November 2024, 6:49 PM  
**Duration** 4 mins 32 secs

### Question 1

Correct  
Marked out of 1.00  
[Flag question](#)

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

Input: ArrayList = [1, 2, 3, 4]

Output: First = 1, Last = 4

Input: ArrayList = [12, 23, 34, 45, 57, 67, 89]

Output: First = 12, Last = 89

#### Approach:

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

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

```

1 import java.util.ArrayList;
2 import java.util.Scanner;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7         ArrayList<Integer> arrayList = new ArrayList<>();
8         int n = scanner.nextInt();
9         for (int i = 0; i < n; i++) {
10             arrayList.add(scanner.nextInt());
11         }
12         System.out.println("ArrayList: "+arrayList);
13         printFirstAndLast(arrayList);
14         scanner.close();
15     }
16
17     public static void printFirstAndLast(ArrayList<Integer> arrayList) {
18         if (arrayList.isEmpty()) {
19             System.out.println("The ArrayList is empty.");
20         } else {
21             int first = arrayList.get(0);
22             int last = arrayList.get(arrayList.size() - 1);
23             System.out.println("First : " + first + ", Last : " + last);
24         }
25     }
26 }
27 
```

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

Passed all tests!

### Question 2

Correct  
Marked out of 1.00  
[Flag question](#)

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

```

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

```

The above methods are used for the below Java program.

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

[Reset answer](#)

```

1 import java.util.ArrayList;
2 import java.util.Scanner;
3
4 public class Prog {
5
6     public static void main(String[] args) {
7         Scanner sc = new Scanner(System.in);
8         int n = sc.nextInt();
9
10        ArrayList<Integer> list = new ArrayList<Integer>();

```

```

11     for (int i = 0; i < n; i++)
12         list.add(sc.nextInt());
13
14
15     // Printing initial value of ArrayList
16     System.out.println("ArrayList: " + list);
17
18     // Replacing the element at index 1 with 100
19     list.set(1, 100); // Fill in here
20
21     // Getting the index of first occurrence of 100
22     System.out.println("Index of 100 = " + list.indexOf(100)); // Fill in here
23
24     // Getting the index of last occurrence of 100
25     System.out.println("LastIndex of 100 = " + list.lastIndexOf(100)); // Fill in here
26
27     // Check whether 200 is in the list or not
28     System.out.println("+" + list.contains(200)); // Fill in here; Output: false
29
30     // Print ArrayList size
31     System.out.println("Size Of ArrayList = " + list.size()); // Fill in here
32
33     // Inserting 500 at index 1
34     list.add(1, 500); // Fill in here
35
36     // Removing an element from position 3
37     list.remove(3); // Fill in here
38
39     // Print final ArrayList
40     System.out.print("ArrayList: " + list);
41
42     sc.close();
43 }

```

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

[Flag question](#)

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

□

Sample input and Output:

Red

Green

Orange

White

Black

Sample output :

List before reversing :

[Red, Green, Orange, White, Black]

List after reversing :

[Black, White, Orange, Green, Red]

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

```

1 v import java.util.ArrayList;
2 import java.util.Collections;
3 import java.util.Scanner;
4
5 v public class ReverseArrayList {
6 v     public static void main(String[] args) {
7 v         Scanner scanner = new Scanner(System.in);
8 v         ArrayList<String> colors = new ArrayList<>();
9
10    int n = scanner.nextInt();
11    scanner.nextLine();
12
13    for (int i = 0; i < n; i++) {
14        colors.add(scanner.nextLine());
15    }
16
17    System.out.println("List before reversing :");
18    System.out.println(colors);
19
20    Collections.reverse(colors);
21
22    System.out.println("List after reversing :");
23    System.out.println(colors);
24
25    scanner.close();
26
27 }
28

```

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!

Finish review

◀ Lab-10-MCQ

Jump to...

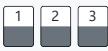


Lab-11-MCQ ▶

# CS23333-Object Oriented Programming Using Java-2023

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

## Quiz navigation



Show one page at a time

Finish review

Status Finished

Started Tuesday, 19 November 2024, 6:40 PM

Completed Tuesday, 19 November 2024, 6:43 PM

Duration 2 mins 48 secs

Question 1

Correct

Marked out of  
1.00

Flag question

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

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

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

## Java HashSet Features

A few important features of HashSet are mentioned below:

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

`public class HashSet<E> extends AbstractSet<E> implements Set<E>, Cloneable, Serializable`

Sample Input and Output:

5

90

56

45

78

25

78

Sample Output:

78 was found in the set.

Sample Input and output:

3

2

7

9

5

Sample Input and output:

5 was not found in the set.

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

[Reset answer](#)

```

1 import java.util.HashSet;
2 import java.util.Scanner;
3
4 class prog {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         int n = sc.nextInt();
8
9         HashSet<Integer> numbers = new HashSet<>();
10        for (int i = 0; i < n; i++) {
11            numbers.add(sc.nextInt());
12        }
13
14        int skey = sc.nextInt();
15        if (numbers.contains(skey)) {
16            System.out.println(skey + " was found in the set.");
17        } else {
18            System.out.println(skey + " was not found in the set.");
19        }
20    }
21 }
```

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

Flag question

Write a Java program to compare two sets and retain elements that are the same.

**Sample Input and Output:**

5

Football

Hockey

```

Cricket
Volleyball
Basketball
7 // HashSet 2:
Golf
Cricket
Badminton
Football
Hockey
Volleyball
Handball
SAMPLE OUTPUT:
Football
Hockey
Cricket
Volleyball
Basketball

```

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

```

1 import java.util.HashSet;
2 import java.util.Scanner;
3
4 public class CompareSets {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         int n1 = sc.nextInt();
9         sc.nextLine();
10        HashSet<String> set1 = new HashSet<>();
11        for (int i = 0; i < n1; i++) {
12            set1.add(sc.nextLine());
13        }
14
15
16
17        int n2 = sc.nextInt();
18        sc.nextLine();
19        HashSet<String> set2 = new HashSet<>();
20        for (int i = 0; i < n2; i++) {
21            set2.add(sc.nextLine());
22        }
23
24
25        set1.retainAll(set2);
26
27
28        for (String element : set1) {
29            System.out.println(element);
30        }
31
32    }
33
34 }
35

```

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

Passed all tests!

### Question 3

Correct

Marked out of  
1.00

[Flag question](#)

### Java HashMap Methods

[containsKey\(\)](#) Indicate if an entry with the specified key exists in the map

[containsValue\(\)](#) Indicate if an entry with the specified value exists in the map

[putIfAbsent\(\)](#) Write an entry into the map but only if an entry with the same key does not already exist

[remove\(\)](#) Remove an entry from the map

[replace\(\)](#) Write to an entry in the map only if it exists

[size\(\)](#) Return the number of entries in the map

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

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

Reset answer

```
1 import java.util.HashMap;
2 import java.util.Map.Entry;
3 import java.util.Set;
4 import java.util.Scanner;
5
6 class prog {
7     public static void main(String[] args) {
8         HashMap<String, Integer> map = new HashMap<>();
9
10        String name;
11        int num;
12        Scanner sc = new Scanner(System.in);
13
14        int n = sc.nextInt();
15
16        for (int i = 0; i < n; i++) {
17            name = sc.next();
18            num = sc.nextInt();
19            map.put(name, num);
20        }
21
22        Set<Entry<String, Integer>> entrySet = map.entrySet();
23        for (Entry<String, Integer> entry : entrySet) {
24            System.out.println(entry.getKey() + " : " + entry.getValue());
25        }
26
27        System.out.println("-----");
28
29        HashMap<String, Integer> anotherMap = new HashMap<>();
30
31        anotherMap.put("SIX", 6);
32        anotherMap.put("SEVEN", 7);
33
34        anotherMap.putAll(map);
35
36        entrySet = anotherMap.entrySet();
37        for (Entry<String, Integer> entry : entrySet) {
38            System.out.println(entry.getKey() + " : " + entry.getValue());
39        }
40
41        map.putIfAbsent("FIVE", 5);
42
43        Integer value = map.get("TWO");
44
45        System.out.println(value);
46
47
48        System.out.println( map.containsKey("ONE"));
49
50        System.out.println(map.containsValue(3));
51
52        System.out.println(map.size());
```

Test	Input	Expected	Got
1	3 ONE TWO 1 THREE TWO ----- 2 SIX : 6 THREE ONE : 1 TWO : 2 SEVEN : 7 THREE : 3 2 true true 4	ONE : 1 TWO : 2 THREE : 3 TWO : 2 SIX : 6 ONE : 1 TWO : 2 SEVEN : 7 THREE : 3 2 true true 4	ONE : 1 TWO : 2 THREE : 3 TWO : 2 SIX : 6 ONE : 1 TWO : 2 SEVEN : 7 THREE : 3 2 true true 4
2	ONE TWO THREE FOUR FIVE SIX SEVEN EIGHT NINE TEN	ONE : 1 TWO : 2 THREE : 3 FOUR : 4 FIVE : 5 SIX : 6 SEVEN : 7 EIGHT : 8 NINE : 9 TEN : 10	ONE : 1 TWO : 2 THREE : 3 FOUR : 4 FIVE : 5 SIX : 6 SEVEN : 7 EIGHT : 8 NINE : 9 TEN : 10
3	ONE TWO THREE FOUR FIVE SIX SEVEN EIGHT NINE TEN	ONE : 1 TWO : 2 THREE : 3 FOUR : 4 FIVE : 5 SIX : 6 SEVEN : 7 EIGHT : 8 NINE : 9 TEN : 10	ONE : 1 TWO : 2 THREE : 3 FOUR : 4 FIVE : 5 SIX : 6 SEVEN : 7 EIGHT : 8 NINE : 9 TEN : 10
4	ONE TWO THREE FOUR FIVE SIX SEVEN EIGHT NINE TEN	ONE : 1 TWO : 2 THREE : 3 FOUR : 4 FIVE : 5 SIX : 6 SEVEN : 7 EIGHT : 8 NINE : 9 TEN : 10	ONE : 1 TWO : 2 THREE : 3 FOUR : 4 FIVE : 5 SIX : 6 SEVEN : 7 EIGHT : 8 NINE : 9 TEN : 10

Passed all tests!

Finish review

◀ Lab-11-MCQ

Jump to...

TreeSet example ▶

# CS23333-Object Oriented Programming Using Java-2023

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

## Quiz navigation



Show one page at a time

Finish review

Status	Finished
Started	Monday, 18 November 2024, 6:59 PM
Completed	Monday, 18 November 2024, 7:00 PM
Duration	1 min 36 secs

### Question 1

Correct

Marked out of  
5.00

[Flag question](#)

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

Get the ASCII values of all the extracted alphabets.

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

Return that single digit as output.

Note:

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

Example 1:

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

input2: {'b', 'c'}

output: 8

Explanation:

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

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

$98 + 99 = 197$

$1 + 9 + 7 = 17$

$1 + 7 = 8$

For example:

Input	Result
a b c	8
b c	

Answer: (penalty regime: 0 %)

```

1 public class CommonCharSum {
2     public static int digitalRoot(int sum) {
3
4         while (sum >= 10) {
5             int tempSum = 0;
6             while (sum > 0) {
7                 tempSum += sum % 10;
8                 sum /= 10;
9             }
10            sum = tempSum;
11        }
12        return sum;
13    }
14
15    public static void main(String[] args) {
16
17        char[] input1 = {'a', 'b', 'c'};
18        char[] input2 = {'b', 'c'};
19
20        int sum1 = 0;
21
22        for (int i = 0; i < input1.length; i++) {
23            for (int j = 0; j < input2.length; j++) {
24                if (input1[i] == input2[j]) {
25                    sum1 += (int) input1[i];
26                    break;
27                }
28            }
29        }
30        int result = digitalRoot(sum1);
31        System.out.println(result);
32    }
33 }
```

Input	Expected	Got
a b c	8	8
b c		

Passed all tests!

### Question 2

Correct

Marked out of  
5.00

[Flag question](#)

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 (00000000000000000000000000000000).

```

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: 0000100000000000000000001000000000000100000000000010000000000001

The decoded string (original word) will be: WIPRO

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

**For example:**

Input	Result
010010001	ZYX
0000100000000000000000001000000000000100000000000010000000000001	WIPRO

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

```

1 import java.util.Scanner;
2
3 public class Main {
4     public static String decode(String str) {
5         StringBuilder decodedWord = new StringBuilder();
6         String[] groups = str.split("1");
7         for (String group : groups)
8         {
9             int zeroCount = group.length();
10            if (zeroCount > 0) {
11                char letter = (char) ('Z' - (zeroCount - 1));
12                decodedWord.append(letter);
13            }
14        }
15
16        return decodedWord.toString();
17    }
18
19    public static void main(String[] args) {
20        Scanner sc = new Scanner(System.in);
21        String input = sc.nextLine();
22
23        String result = decode(input);
24        System.out.println(result);
25
26        sc.close();
27    }
28 }

```

Input	Expected	Got
010010001	ZYX	ZYX
0000100000000000000000001000000000000100000000000010000000000001	WIPRO	WIPRO

Passed all tests!

### Question 3

Correct

Marked out of  
5.00

[Flag question](#)

Write a function that takes an input String (sentence) and generates a new String (modified sentence) by reversing the words in the original String, maintaining the words position.

In addition, the function should be able to control the reversing of the case (upper or lowercase) based on a case\_option parameter, as follows:

If case\_option = 0, normal reversal of words i.e., if the original sentence is "Wipro TechNologies BangaLore", the new reversed sentence should be "orpiW seigoloNhceT erolagnab".

If case\_option = 1, reversal of words with retaining position's case i.e., if the original sentence is "Wipro TechNologies BangaLore", the new reversed sentence should be "Orpiw SeigOlhoncet Erolagnab".

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

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

NOTE:

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

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

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

Examples:

S. No.	input1	input2	output
1	Wipro Technologies Bangalore	0	orpiW seigolonhceT erolagnab
2	Wipro Technologies, Bangalore	0	orpiW,seiGolonhceT Erolagnab
3	Wipro Technologies Bangalore	1	Orpiw SeigolonhceT Erolagnab
4	Wipro Technologies, Bangalore	1	Orpiw,seiGolonhceT Erolagnab

For example:

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

Answer: (penalty regime: 0 %)

```
1 import java.util.Scanner;
2 public class Main {
3     public static String rev(String str1, int caseOption) {
4         String[] eachWords = str1.split(" ");
5         StringBuilder revString = new StringBuilder();
6         for (String word : eachWords) {
7             StringBuilder r = new StringBuilder(word).reverse();
8             if (r.charAt(r.length() - 1) == ',') {
9                 char lastChar = r.charAt(r.length() - 1);
10                r.deleteCharAt(r.length() - 1);
11                r.reverse();
12                r.append(lastChar);
13            }
14            if (caseOption == 1) {
15                for (int i = 0; i < word.length(); i++) {
16                    if (Character.isUpperCase(word.charAt(i))) {
17                        r.setCharAt(i, Character.toUpperCase(r.charAt(i)));
18                    } else if (Character.isLowerCase(word.charAt(i))) {
19                        r.setCharAt(i, Character.toLowerCase(r.charAt(i)));
20                    }
21                }
22            }
23            revString.append(r).append(" ");
24        }
25        return revString.toString().trim();
26    }
27    public static void main(String[] args) {
28        Scanner sc = new Scanner(System.in);
29        String strGiven = sc.nextLine();
30        int caseOption = sc.nextInt();
31        String result = rev(strGiven, caseOption);
32        System.out.println(result);
33        sc.close();
34    }
35 }
```

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!

Finish review

◀ Lab-12-MCQ

Jump to...

Identify possible words ▶