

RAJALAKSHMI ENGINEERING COLLEGE

RAJALAKSHMI NAGAR, THANDALAM – 602 105



**RAJALAKSHMI
ENGINEERING COLLEGE**

CS23333 OBJECT ORIENTED PROGRAMMING USING JAVA

Laboratory Record Note Book

Name	Aravinthaa S
Year / Branch / Section	II / B.E.CSE / A
Register No	230701032
Semester	III
Academic Year	2024-2025

CS23333-Object Oriented Programming Using Java-2023

Week 1

Quiz navigation



Show all questions on one page

[Finish review](#)

Status Finished

Started Sunday, 22 September 2024, 12:10 AM

Completed Sunday, 22 September 2024, 12:45 AM

Duration 34 mins 43 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.Scanner;
2 public class OddCheck{
3     public static void main(String[] args){
4         Scanner scanner=new Scanner(System.in);
5         int number=scanner.nextInt();
6         if(number!=0&& number%2!=0){
7             System.out.println(2);
8         }else{
9             System.out.println(1);
10        }
11    }
12 }
13
14 }
```

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

Passed all tests! ✓

[Next page ►](#)

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

Finish review

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.Scanner;
2 public class LastDigit{
3     public static void main(String[] args){
4         Scanner scanner=new Scanner(System.in);
5         int number= scanner.nextInt();
6         int lastDigit=Math.abs(number)%10;
7         System.out.println(lastDigit);
8     }
9 }
```

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

Passed all tests! ✓

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

[Finish review](#)

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: Tie sign of the input numbers should be ignored.

i.e.

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

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

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

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

For example:

Input	Result
267	11
154	
267	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 scanner=new Scanner(System.in);
5 |         int num1=scanner.nextInt();
6 |         int num2=scanner.nextInt();
7 |         int lastDigit1=Math.abs(num1)%10;
8 |         int lastDigit2=Math.abs(num2)%10;
9 |         int sum=lastDigit1+lastDigit2;
10 |         System.out.println(sum);
11 |     }
12 |

```

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

Passed all tests! ✓

[◀ Previous page](#)

[Finish review](#)

CS23333-Object Oriented Programming Using Java-2023

Week 2

Quiz navigation



Show all questions on one page

[Finish review](#)

Status Finished
Started Monday, 30 September 2024, 6:43 PM
Completed Wednesday, 2 October 2024, 8:31 AM
Duration 1 day 13 hours

Question 1

Correct

Marked out of 5.00

[Flag question](#)

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

Logic to print number in words in Java programming.

Example**Input**

1234

Output

One Two Three Four

Input:

16

Output:

one six

For example:

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

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class NumberToWords{
3     static String[] words={"Zero","One","Two","Three","Four","Five","Six","Seven","Eight","Nine"};
4     public static void main(String[] args){
5         Scanner scanner=new Scanner(System.in);
6         while(scanner.hasNext()){
7             String number=scanner.next();
8             printNumberInWords(number);
9         }
10        scanner.close();
11    }
12    public static void printNumberInWords(String number){
13        for(int i=0;i<number.length();i++){
14            char digit=number.charAt(i);
15            if(Character.isDigit(digit)){
16                System.out.print(words[digit - '0']+ " ");
17            }
18        }
19        System.out.println();
20    }
21 }
```

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

Passed all tests! ✓

CS2333-Object Oriented Programming Using Java-2023

Quiz navigation

1 **2** **3**

Show all questions on one page

Finish review

Question **2**

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        Scanner sc=new Scanner(System.in);
28        int n = sc.nextInt();
29        sc.close();
30        int result=findTrailingzeros(n);
31        System.out.println(result);
32    }
33 }
34

```

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

Passed all tests! ✓

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

Finish review

Question 3

Correct

Marked out of
5.00

Flag question

You and your friend are movie fans and want to predict if the movie is going to be a hit!

The movie's success formula depends on 2 parameters:

the acting power of the actor (range 0 to 10)

the critic's rating of the movie (range 0 to 10)

The movie is a hit if the acting power is excellent (more than 8) or the rating is excellent (more than 8). This holds true except if either the acting power is poor (less than 2) or rating is poor (less than 2), then the movie is a flop. Otherwise the movie is average.

Write a program that takes 2 integers:

the first integer is the acting power

second integer is the critic's rating.

You have to print Yes if the movie is a hit, Maybe if the movie is average and No if the movie is flop.

Example input:

9 5

Output:

Yes

Example input:

1 9

Output:

No

Example input:

6 4

Output:

Maybe

For example:

Input	Result
9 5	Yes
1 9	No
6 4	Maybe

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class MoviePrediction{
3     public static void main(String[] args){
4         Scanner scanner=new Scanner(System.in);
5         int actingPower=scanner.nextInt();
6         int criticsRating=scanner.nextInt();
7         scanner.close();
8         if(actingPower<2|| criticsRating<2){
9             System.out.println("No");
10        }else if(actingPower>8 || criticsRating>8){
11            System.out.println("Yes");
12        }else{
13            System.out.println("Maybe");
14        }
15    }
16 }
```

	Input	Expected	Got	
✓	9 5	Yes	Yes	✓
✓	1 9	No	No	✓
✓	6 4	Maybe	Maybe	✓

Passed all tests! ✓

◀ Previous page

Finish review

Quiz navigation



Show all questions on one page

[Finish review](#)

Status	Finished
Started	Wednesday, 2 October 2024, 8:32 AM
Completed	Wednesday, 2 October 2024, 9:42 AM
Duration	1 hour 9 mins

Question 1

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 -9 9	-162 0

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class ArrayOperations{
3     public static void main(String[] args){
4         Scanner scanner=new Scanner(System.in);
5         int n=scanner.nextInt();
6         int[] arr=new int[n];
7         for(int i=0;i<n;i++){
8             arr[i]=scanner.nextInt();
9         }
10        scanner.close();
11        int[] result=performOperations(arr);
12        for(int num:result){
13            System.out.print(num+" ");
14        }
15    }
16    public static int[] performOperations(int[] arr){
17        int n=arr.length;
18        int max=findMax(arr);
19        int[] subtractedArr=new int[n];
20        for(int i=0;i<n;i++){
21            subtractedArr[i]=arr[i]-max;
22        }
23        int[] finalArr=new int[n];
24        for(int i=0;i<n;i++){
25            finalArr[i]=subtractedArr[i]*max;
26        }
27        return finalArr;
28    }
29    public static int findMax(int[] arr){
30        int max=arr[0];
31        for(int i=1;i<arr.length;i++){
32            if(arr[i]>max){
33                max=arr[i];
34            }
35        }
36        return max;
37    }
38 }
```

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! ✓

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

[Finish review](#)

Question 2

Correct

Marked out of 5.00

[Flag question](#)

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, 14" is the longest of the four as it contains 4 elements. Therefore, the expected output = sum of the longest sequence of POSITIVE numbers = $12 + 18 + 18 + 14 = 62$.

Example 2:

input1 = 11

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

Expected output = -1

Explanation:

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

Example 3:

input1 = 16

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

Expected output = 174

Explanation:

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

For example:

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

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class LongestPositiveSequenceSum{
3     public static int findLongestPositiveSequenceSum(int[] input2,int input1){
4         int maxLength=0,maxSum=0;
5         int currentLength=0,currentSum = 0;
6         boolean hasPositive=false;
7         for(int i=0;i<input1;i++){
8             if(input2[i]>0){
9                 hasPositive=true;
10                currentLength++;
11                currentSum += input2[i];
12            }else{
13                if(currentLength>maxLength){
14                    maxLength=currentLength;
15                    maxSum=currentSum;
16                }else if(currentLength==maxLength){
17                    maxSum += currentSum;
18                }
19                currentLength=0;
20                currentSum=0;
21            }
22        }
23        if(currentLength>maxLength){
24            maxSum=currentSum;
25        }else if(currentLength==maxLength){
26            maxSum += currentSum;
27        }
28        if(!hasPositive){
29            return -1;
30        }
31        return maxSum;
32    }
33    public static void main(String[] args){
34        Scanner sc=new Scanner(System.in);
35        int input1=sc.nextInt();
36        int[] input2=new int[input1];
37        for(int i=0;i<input1;i++){
38            input2[i]=sc.nextInt();
39        }
40        int result=findLongestPositiveSequenceSum(input2,input1);
41        System.out.println(result);
42    }
43 }
```

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! ✓

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

[Finish review](#)

Question 3

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 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 input2: {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	

Input	Result
5	53
1 5 423 310 61540	

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class EncodedArraySum{
3     public static int calculateSum(int[] input1,int input2){
4         int sum=0;
5         for(int i=0;i<input2;i++){
6             int number = input1[i];
7             int digit=getDigitAtPosition(number,i);
8             sum +=(digit*digit);
9         }
10        return sum;
11    }
12    private static int getDigitAtPosition(int number,int position){
13        String numStr=Integer.toString(number);
14        if(numStr.length()<=position){
15            return 0;
16        }
17        return Character.getNumericValue(numStr.charAt(numStr.length()-1-position));
18    }
19    public static void main(String[] args){
20        Scanner sc=new Scanner(System.in);
21        int input2=sc.nextInt();
22        int[] input1=new int[input2];
23        for(int i=0;i<input2;i++){
24            input1[i]=sc.nextInt();
25        }
26        int result=calculateSum(input1,input2);
27        System.out.println(result);
28    }
29 }
```

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

Passed all tests! ✓

CS2333-Object Oriented Programming Using Java-2023

Week 4

Quiz navigation



Show all questions on one page

Finish review

Status Finished
Started Wednesday, 2 October 2024, 2:35 PM
Completed Wednesday, 2 October 2024, 3:09 PM
Duration 34 mins 27 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 = πr^2 **Circumference = $2\pi r$** **Input:**

2

Output:**Area = 12.57****Circumference = 12.57****For example:**

Test	Input	Result
1	4	Area = 50.27 Circumference = 25.13

Answer: (penalty regime: 0 %)

Reset answer

```

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

```

	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! ✓

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

Finish review

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 manufacturer;
3     private String operating_system;
4     public String color;
5     private int cost;
6     public Mobile(String manufacturer,String operating_system,String color,int cost){
7         this.manufacturer=manufacturer;
8         this.operating_system=operating_system;
9         this.color=color;
10        this.cost=cost;
11    }
12    public void setManufacturer(String manufacturer){
13        this.manufacturer=manufacturer;
14    }
15    public String getmanufacturer(){
16        return manufacturer;
17    }
18    public void setOperatingSystem(String operating_system){
19        this.operating_system=operating_system;
20    }
21    public String getOperatingSystem(){
22        return operating_system;
23    }
24    public void setColor(String color){
25        this.color=color;
26    }
27    public String getColor(){
28        return color;
29    }
30    public void setCost(int cost){
31        this.cost=cost;
32    }
33    public int getCost(){
34        return cost;
35    }
36    public String toString(){
37        return "manufacturer = " + manufacturer +
38            "\noperating_system = " + operating_system +
39            "\ncolor = " + color +
40            "\ncost = " + cost;
41    }
42    public static void main(String[] args){
43        Mobile mobile=new Mobile("Redmi","Andriod","Blue",34000);
44        System.out.println(mobile.toString());
45    }
46 }
47 }
```

	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! ✓

CS2333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

[Finish review](#)

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 Student{
2 |     private String name;
3 |     private int rollno;
4 |     public Student(){
5 |         this.name= null ;
6 |         this.rollno=0;
7 |         System.out.println("No-arg constructor is invoked");
8 |     }
9 |     public Student(String name){
10 |         this.name=name;
11 |         this.rollno=0;
12 |         System.out.println("1 arg constructor is invoked");
13 |     }
14 |     public Student(String name,int rollno){
15 |         this.name=name;
16 |         this.rollno=rollno;
17 |         System.out.println("2 arg constructor is invoked");
18 |     }
19 |     public void display(){
20 |         System.out.println("Name =" + name + " , Roll no = " + rollno);
21 |     }
22 |     public static void main(String[] args){
23 |         Student s1= new Student();
24 |         Student s2 = new Student("Rajalakshmi");
25 |         Student s3 = new Student("Lakshmi",101);
26 |         s1.display();
27 |         s2.display();
28 |         s3.display();
29 |     }
30 | }
```

	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! ✓

[◀ Previous page](#)

[Finish review](#)

CS2333-Object Oriented Programming Using Java-2023

Week 5

Quiz navigation

[1](#) [2](#) [3](#)[Show all questions on one page](#)[Finish review](#)

Status Finished
Started Wednesday, 2 October 2024, 3:11 PM
Completed Wednesday, 2 October 2024, 4:01 PM
Duration 50 mins 48 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     protected String collegeName;
3
4     public College(String collegeName) {
5         // initialize the instance variables
6         this.collegeName = collegeName;
7     }
8
9     public void admitted() {
10        System.out.println("A student admitted in " + collegeName);
11    }
12 }
13 }
14 class Student extends College{
15     String studentName;
16     String department;
17
18     public Student(String collegeName, String studentName, String department) {
19         // initialize the instance variables
20         super(collegeName);
21         this.studentName = studentName;
22         this.department = department;
23     }
24
25     public String toString(){
26         // return the details of the student
27         return "CollegeName : " + collegeName + "\nStudentName : " + studentName + "\nDepartment : " + department;
28     }
29 }
30 }
31 }
32 }
33 public class Main {
34     public static void main (String[] args) {
35         Student s1 = new Student("REC", "Venkatesh", "CSE");
36         s1.admitted();                                // invoke the admitted() method
37         System.out.println(s1.toString());
38     }
39 }
40 }
```

	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! ✓

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

Finish review

Question 2

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. class BankAccount {  
2.     // Private field to store the account number  
3.     private String accountNumber;  
4.  
5.     // Private field to store the balance  
6.     private double balance;  
7.  
8.     // Constructor to initialize account number and balance  
9.     public BankAccount(String accountNumber, double balance){  
10.         this.accountNumber=accountNumber;  
11.         this.balance=balance;  
12.     }  
13.  
14.  
15.  
16.  
17.     // Method to deposit an amount into the account  
18.     public void deposit(double amount) {  
19.         // Increase the balance by the deposit amount  
20.         balance += amount;  
21.         System.out.println("New balance after depositing $" + (int)amount + ": $" + balance);  
22.     }  
23.  
24.     // Method to withdraw an amount from the account  
25.     public void withdraw(double amount) {  
26.         // Check if the balance is sufficient for the withdrawal  
27.         if (balance >= amount) {  
28.             // Decrease the balance by the withdrawal amount  
29.             balance -= amount;  
30.         } else {  
31.             // Print a message if the balance is insufficient  
32.             System.out.println("Insufficient balance");  
33.         }  
34.     }  
35.  
36.     // Method to get the current balance  
37.     public double getBalance() {  
38.         // Return the current balance  
39.         return balance;  
40.     }  
41. }  
42.  
43. class SavingsAccount extends BankAccount {  
44.     // Constructor to initialize account number and balance  
45.     public SavingsAccount(String accountNumber, double balance) {  
46.         // Call the parent class constructor  
47.         super(accountNumber, balance);  
48.     }  
49.  
50.     // Override the withdraw method from the parent class  
51.     @Override  
52.     public void withdraw(double amount) {
```

Expected

- ✓ 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

Got

- Create a Bank Account object (A/c No. BA1234)
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)
Try to withdraw \$250 from SA1000!
Minimum balance of \$100 required!
Balance after trying to withdraw \$250: \$300.0

Passed all tests! ✓

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

Finish review

Question 3

Correct

Marked out of
5.00

Flag question

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

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

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

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

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

For example:

Result

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

Answer: (penalty regime: 0 %)

```
1. class Mobile{  
2.     public Mobile(){  
3.         System.out.println("Basic Mobile is Manufactured");  
4.     }  
5.     public void basicMobile(){  
6.     }  
7. }  
8. class CameraMobile extends Mobile{  
9.     public CameraMobile(){  
10.        super();  
11.        System.out.println("Camera Mobile is Manufactured");  
12.    }  
13.    public void newFeature(){  
14.        System.out.println("Camera Mobile with 5MG px");  
15.    }  
16. }  
17. class AndroidMobile extends CameraMobile{  
18.     public AndroidMobile(){  
19.        super();  
20.        System.out.println("Android Mobile is Manufactured");  
21.    }  
22.    public void androidMobile(){  
23.        System.out.println("Touch Screen Mobile is Manufactured");  
24.    }  
25. }  
26. public class main{  
27.     public static void main(String[] args){  
28.         AndroidMobile androidMobile=new AndroidMobile();  
29.         androidMobile.newFeature();  
30.         androidMobile.androidMobile();  
31.     }  
32. }  
33. }
```

	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! ✓

◀ Previous page

Finish review

CS23333-Object Oriented Programming Using Java-2023

Week 6

Quiz navigation



Show all questions on one page

[Finish review](#)

Status	Finished
Started	Sunday, 6 October 2024, 8:53 AM
Completed	Sunday, 6 October 2024, 9:50 AM
Duration	57 mins 6 secs

Question 1

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 2nd position is b

Word2 is za, both are not same alphabets

position value of z is 26

position value of a is 1

max – min will be 26 – 1 = 25

Alphabet which comes in 25th position is y

word3 is ee, both are same hence take e

Hence the output is BYE

For example:

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

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2 public class StringProcessor{
3     public static String generateOutput(String input){
4         String[] words=input.split(":");
5         StringBuilder output=new StringBuilder();
6         for(String word : words){
7             if(word.length()>2){
8                 char firstChar=word.charAt(0);
9                 char secondChar=word.charAt(1);
10                if(firstChar==secondChar){
11                    output.append(Character.toUpperCase(firstchar));
12                }else{
13                    int positionFirst=firstChar - 'a' + 1;
14                    int positionSecond=secondChar - 'a' + 1;
15                    int maxMin=Math.abs(positionFirst - positionSecond);
16                    char resultChar=(char) ('A' + maxMin - 1);
17                    output.append(resultchar);
18                }
19            }
20        }
21        return output.toString();
22    }
23    public static void main(String[] args){
24        Scanner scanner=new Scanner(System.in);
25        String userInput=scanner.nextLine();
26        String result=generateOutput(userInput);
27        System.out.println(result);
28        scanner.close();
29    }
30 }
```

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

Passed all tests! ✓

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

Finish review

Question 2

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 "dO".

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

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

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

Expected output:

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

Example 1:

input1 = "Today is a Nice Day"

input2 = 41

output = "iNce doTday"

Example 2:

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

input2 = 39

output = "naMngO arGpes"

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

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

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

For example:

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

Answer: (penalty regime: 0 %)

```

1+ import java.util.Scanner;
2+ public class WordProcessor{
3+     public static void main(String[] args){
4+         Scanner scanner = new Scanner(System.in);
5+         String input1=scanner.nextLine();
6+         int input2=scanner.nextInt();
7+         System.out.println(processWords(input1,input2));
8+     }
9+     public static String processWords(String input1, int input2){
10+        String[] words=input1.split(" ");
11+        int firstWordIndex=input2 % 10 - 1;
12+        int secondWordIndex=input2 / 10 - 1;
13+        return processWord(words[secondWordIndex]) + " " + processWord(words[firstWordIndex]);
14+    }
15+    public static String processWord(String word){
16+        int length=word.length();
17+        int middleIndex=length / 2;
18+        String middleToBegin;
19+        String middleToEnd;
20+        if(length % 2 == 0){
21+            middleToBegin=new StringBuilder(word.substring(0, middleIndex)).reverse().toString();
22+            middleToEnd=word.substring(middleIndex);
23+        }else{
24+            middleToBegin=new StringBuilder(word.substring(0, middleIndex + 1)).reverse().toString();
25+            middleToEnd=word.substring(middleIndex);
26+        }
27+        return middleToBegin + middleToEnd;
28+    }
29+ }
30+

```

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! ✓

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

Finish review

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.Arrays;
2 import java.util.Collections;
3 import java.util.Scanner;
4 import java.util.Set;
5 import java.util.stream.Collectors;
6 public class StringConcatenation{
7     public static void main(String[] args){
8         Scanner scanner = new Scanner(System.in);
9         String input1=scanner.nextLine();
10        String input2=scanner.nextLine();
11        String concatenatedString=concatenateAndRemoveDuplicates(input1, input2);
12        if(concatenatedString==null){
13            System.out.println("null");
14        }else{
15            String result=arrangeAlphabetsDescending(concatenatedString);
16            System.out.println(result);
17        }
18        scanner.close();
19    }
20    public static String concatenateAndRemoveDuplicates(String input1, String input2){
21        if(input1==null || input2==null || input1.trim().isEmpty() || input2.trim().isEmpty()){
22            return null;
23        }
24        String concatenatedString = input1 + input2;
25        Set<Character> uniqueChars = concatenatedString.chars()
26                .mapToObj(c -> (char) c)
27                .filter(c -> Character.isLetter(c))
28                .collect(Collectors.toSet());
29        return uniqueChars.stream()
30                .map(String::valueOf)
31                .collect(Collectors.joining());
32    }
33    public static String arrangeAlphabetsDescending(String str){
34        char[] charArray=str.toCharArray();
35        Arrays.sort(charArray);
36        StringBuilder descendingOrder =new StringBuilder (new String(charArray));
37        return descendingOrder.reverse().toString();
38    }
39 }

```

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

Passed all tests! ✓

◀ Previous page

Finish review

CS23333-Object Oriented Programming Using Java-2023

Week 7

Quiz navigation

1 2 3

Show all questions on one page

Finish review

Status Finished
Started Thursday, 17 October 2024, 12:09 PM
Completed Thursday, 17 October 2024, 1:10 PM
Duration 1 hour 1 min

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

	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! ✓

Next page ►

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation



Show all questions on one page

Finish review

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+ import java.util.Scanner;
2+ interface Playable{
3+     void play();
4+ }
5+ class Football implements Playable{
6+     String name;
7+     public Football(String name){
8+         this.name=name;
9+     }
10+    public void play(){
11+        System.out.println(name + " is Playing football");
12+    }
13+ }
14+ class Volleyball implements Playable{
15+     String name;
16+     public Volleyball(String name){
17+         this.name=name;
18+     }
19+    public void play(){
20+        System.out.println(name + " is Playing volleyball");
21+    }
22+ }
23+ class Basketball implements Playable{
24+     String name;
25+     public Basketball(String name){
26+         this.name=name;
27+     }
28+    public void play(){
29+        System.out.println(name + " is Playing basketball");
30+    }
31+ }
32+ public class Main{
33+     public static void main(String[] args){
34+         Scanner scanner=new Scanner(System.in);
35+         String footballPlayer=scanner.nextLine().trim();
36+         String volleyballPlayer=scanner.nextLine().trim();
37+         String basketballPlayer=scanner.nextLine().trim();
38+         Football football=new Football(footballPlayer);
39+         Volleyball volleyball=new Volleyball(volleyballPlayer);
40+         Basketball basketball=new Basketball(basketballPlayer);
41+         football.play();
42+         volleyball.play();
43+         basketball.play();
44+         scanner.close();
45+     }
46+ }
47+ }
```

	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 ✓

Passed all tests! ✓

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation



Show all questions on one page

[Finish review](#)

Question 3

Correct

Marked out of
5.00

[Flag question](#)

Create interfaces shown below.

```
interface Sports {
    public void setHomeTeam(String name);
    public void setVisitingTeam(String name);
}
```

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

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

sample Input:

```
Rajalakshmi
Saveetha
22
21
```

Output:

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

For example:

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

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1+ import java.util.Scanner;
2+ interface Sports {
3+     public void setHomeTeam(String name);
4+     public void setVisitingTeam(String name);
5+
6+ }
7+ interface Football extends Sports {
8+     public void homeTeamScored(int points);
9+     public void visitingTeamScored(int points);
10+
11+ }
12+ class College implements Football {
13+     String homeTeam;
14+     String visitingTeam;
15+
16+     public void setHomeTeam(String name){
17+         homeTeam=name;
18+     }
19+
20+     public void setVisitingTeam(String name){
21+         visitingTeam=name;
22+     }
23+
24+     public void homeTeamScored(int points){
25+         System.out.println(homeTeam+" "+points+" scored");
26+     }
27+     public void visitingTeamScored(int points){
28+         System.out.println(visitingTeam+" "+points+" scored");
29+     }
30+     public void winningTeam(int p1, int p2){
31+         if(p1>p2){
32+             System.out.println(homeTeam + " is the winner!");
33+         }
34+
35+         else if(p1<p2){
36+             System.out.println(visitingTeam + " is the winner!");
37+         }else{
38+             System.out.println("It's a tie match.");
39+         }
40+
41+     }
42+
43+ public class Main{
44+     public static void main(String[] args){
45+         //String hname;
46+         Scanner sc= new Scanner(System.in);
47+         String hname=sc.nextLine();
48+         String vteam=sc.nextLine();
49+         int hpoints=sc.nextInt();
50+         int vpoints=sc.nextInt();
51+         College s= new College();
52+         s.setHomeTeam(hname);
```

	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! ✓

[Previous page](#)

[Finish review](#)

Quiz navigation



Show all questions on one page

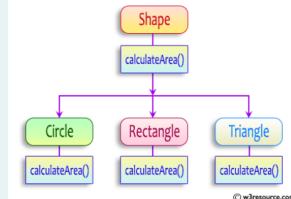
Finish review

Status Finished
Started Saturday, 5 October 2024, 1:33 PM
Completed Saturday, 5 October 2024, 6:43 PM
Duration 5 hours 9 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 %)

```

4
5 * class Circle extends Shape{
6   private double radius;
7   public Circle(double radius){
8     this.radius=radius;
9   }
10  public double calculateArea(){
11    return Math.PI * radius * radius;
12  }
13 }
14 * class Rectangle extends Shape {
15   private double length;
16   private double breadth;
17   public Rectangle(double length,double breadth){
18     this.length=length;
19     this.breadth=breadth;
20   }
21   public double calculateArea(){
22    return length * breadth;
23  }
24 }
25 * class Triangle extends Shape{
26   private double base;
27   private double height;
28   public Triangle(double base,double height){
29     this.base=base;
30     this.height=height;
31   }
32   public double calculateArea(){
33    return 0.5 * base * height;
34  }
35 }
36 public class Main{
37   public static void main(String[] args){
38     Scanner sc=new Scanner(System.in);
39     double radius=sc.nextDouble();
40     Shape circle=new Circle(radius);
41     double length=sc.nextDouble();
42     double breadth=sc.nextDouble();
43     Shape rectangle=new Rectangle(length,breadth);
44     double base=sc.nextDouble();
45     double height=sc.nextDouble();
46     Shape triangle=new Triangle(base,height);
47     System.out.printf("Area of a circle: %.2f\n", circle.calculateArea());
48     System.out.printf("Area of a Rectangle: %.2f\n",rectangle.calculateArea());
49     System.out.printf("Area of a Triangle: %.2f\n",triangle.calculateArea());
50     sc.close();
51   }
52 }
53 }
54 }
55 }
  
```

	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! ✓

CS2333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

Finish review

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. class FinalExample {
2.
3.     // Final variable
4.     final int maxSpeed = 120;
5.
6.     // Final method
7.     public final void displayMaxSpeed() {
8.         System.out.println ("The maximum speed is: " + maxSpeed + " km/h");
9.     }
10.
11.
12. class SubClass extends FinalExample {
13.
14.     // public final void displayMaxSpeed() {
15.     //     System.out.println("Cannot override a final method");
16.     // }
17.
18.     // You can create new methods here
19.     public final void showDetails() {
20.         System.out.println("This is a subclass of FinalExample.");
21.     }
22.
23.
24. public class prog {
25.     public static void main(String[] args) {
26.         FinalExample obj = new FinalExample();
27.         obj.displayMaxSpeed();
28.
29.         SubClass subObj = new SubClass();
30.         subObj.showDetails();
31.     }
32. }
```

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

Passed all tests! ✓

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

Finish review

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+ public class VowelExtraction{
3+     public static String extractVowelStrings(String[] strings){
4+         StringBuilder result=new StringBuilder();
5+         for (String str:strings){
6+             if(isVowel(str.charAt(0)) && isVowel(str.charAt(str.length()-1))){
7+                 result.append(str);
8+             }
9+         }
10+        return result.toString().toLowerCase();
11+    }
12+    public static boolean isVowel(char c){
13+        return "aeiouAEIOU".indexOf(c) != -1;
14+    }
15+    public static void main(String[] args){
16+        Scanner scanner = new Scanner(System.in);
17+        int n=scanner.nextInt();
18+        String[] strings=new String[n];
19+        for(int i=0;i<n;i++){
20+            strings[i]=scanner.next();
21+        }
22+        String result=extractVowelStrings(strings);
23+        if(result.isEmpty()){
24+            System.out.println("no matches found");
25+        }else{
26+            System.out.println(result);
27+        }
28+        scanner.close();
29+    }
30+ }
```

	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! ✓

◀ Previous page

Finish review

CS2333-Object Oriented Programming Using Java-2023

Week 9

Quiz navigation



Show all questions on one page

Finish review

Status Finished
Started Thursday, 10 October 2024, 4:48 PM
Completed Thursday, 10 October 2024, 4:55 PM
Duration 7 mins 2 secs

Question 1

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             name[i] = sc.nextInt(); // Attempt to read integer input
18         }
19
20         // calculate the sum of the array
21         for (int value : name) {
22             sum += value;
23         }
24
25         // Print the total sum of the array
26         System.out.println(sum);
27
28     }
29     catch(InputMismatchException e )
30     {
31
32         System.out.println("You entered bad data.");
33     }finally{
34         sc.close();
35     }
36 }
37 }
38 }
```

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

Passed all tests! ✓

CS2333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

Finish review

Question 2

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	<code>java.lang.ArithmaticException: / by zero</code> I am always executed

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2
3 public class ExceptionHandlingDemo {
4
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7
8         int n = sc.nextInt();
9         int[] arr = new int[n];
10
11        for (int i = 0; i < n; i++) {
12            arr[i] = sc.nextInt();
13        }
14
15        try {
16            int result = arr[0] / arr[1];
17
18            System.out.println(arr[3]);
19
20        } catch (ArithmaticException e) {
21            System.out.println("java.lang.ArithmaticException: / by zero");
22        } catch (ArrayIndexOutOfBoundsException e) {
23            System.out.println("java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length " + n);
24        } finally {
25            System.out.println("I am always executed");
26        }
27
28        sc.close();
29    }
30}

```

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

Passed all tests! ✓

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

Finish review

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

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

Passed all tests!

Previous page

Finish review

CS2333-Object Oriented Programming Using Java-2023

Week 10

Quiz navigation



Show all questions on one page

Finish review

Status Finished
Started Wednesday, 30 October 2024, 5:42 PM
Completed Wednesday, 30 October 2024, 6:55 PM
Duration 1 hour 13 mins

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 = 4Input: 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 FirstAndLastElement {
5 *     public static void main(String[] args) {
6 *         Scanner scanner = new Scanner(System.in);
7 *
8 *         int n = scanner.nextInt();
9 *         ArrayList<Integer> list = new ArrayList<>();
10 *
11 *
12 *         for (int i = 0; i < n; i++) {
13 *             list.add(scanner.nextInt());
14 *         }
15 *
16 *         printFirstAndLast(list);
17 *     }
18 *
19 *
20 *     public static void printFirstAndLast(ArrayList<Integer> list) {
21 *         if (list.isEmpty()) {
22 *             System.out.println("The list is empty.");
23 *             return;
24 *         }
25 *         int first = list.get(0);
26 *         int last = list.get(list.size() - 1);
27 *         System.out.println("ArrayList: " + list);
28 *         System.out.println("First : " + first + ", Last : " + last);
29 *     }
30 * }
31 */

```

	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! ✓

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

[Finish review](#)

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

	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! ✓

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

Finish review

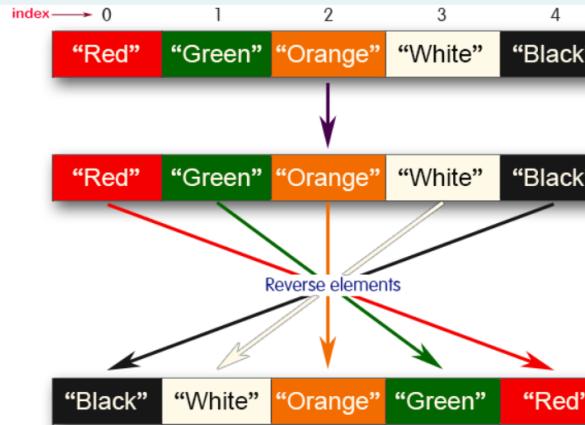
Question 3

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

	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! ✓

◀ Previous page

Finish review

CS2333-Object Oriented Programming Using Java-2023

Week 11

Quiz navigation



Show all questions on one page

Finish review

Status Finished
Started Saturday, 9 November 2024, 5:48 PM
Completed Saturday, 9 November 2024, 6:46 PM
Duration 57 mins 52 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 class prog {
4     public static void main(String[] args) {
5         Scanner sc= new Scanner(System.in);
6         int n = sc.nextInt();
7         // Create a HashSet object called numbers
8         HashSet<Integer> numbers = new HashSet<>();
9
10        // Add values to the set
11        for(int i=0;i<n;i++){
12            numbers.add(sc.nextInt());
13        }
14
15        int skey=sc.nextInt();
16
17        // Show which numbers between 1 and 10 are in the set
18
19        if (numbers.contains(skey)){
20            System.out.println(skey + " was found in the set.");
21        } else {
22            System.out.println(skey + " was not found in the set.");
23        }
24    }
25 }
```

	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! ✓

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

Finish review

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        int n2 = sc.nextInt();
16        sc.nextLine();
17        HashSet<String> set2 = new HashSet<>();
18        for (int i = 0; i < n2; i++) {
19            set2.add(sc.nextLine());
20        }
21
22        set1.retainAll(set2);
23
24        for (String element : set1) {
25            System.out.println(element);
26        }
27    }
28 }
29

```

	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! ✓

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

[Finish review](#)

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 class prog
6 {
7     public static void main(String[] args)
8     {
9         //Creating HashMap with default initial capacity and load factor
10        HashMap<String, Integer> map = new HashMap<String, Integer>();
11
12        String name;
13        int num;
14        Scanner sc= new Scanner(System.in);
15        int n=sc.nextInt();
16        for(int i =0;i<n;i++)
17        {
18            name=sc.next();
19            num= sc.nextInt();
20            map.put(name,num);
21        }
22
23        //Printing key-value pairs
24
25        Set<Entry<String, Integer>> entrySet = map.entrySet();
26
27        for (Entry<String, Integer> entry : entrySet)
28        {
29            System.out.println(entry.getKey()+" : "+entry.getValue());
30        }
31        System.out.println("-----");
32        //Creating another HashMap
33
34        HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();
35
36        //Inserting key-value pairs to anotherMap using put() method
37
38        anotherMap.put("SIX", 6);
39
40        anotherMap.put("SEVEN", 7);
41
42        //Inserting key-value pairs of map to anotherMap using putAll() method
43
44        anotherMap.putAll(map); // code here
45
46        //Printing key-value pairs of anotherMap
47
48        entrySet = anotherMap.entrySet();
49
50        for (Entry<String, Integer> entry : entrySet)
51        {
52            System.out.println(entry.getKey()+" : "+entry.getValue());
53        }

```

	Test	Input	Expected	Got	
✓	1	3 ONE TWO 1 THREE : 3 TWO ----- 2 SIX : 6 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! ✓

[◀ Previous page](#)

[Finish review](#)

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

[1](#) [2](#) [3](#)

Show all questions on one page

[Finish review](#)

Question 2

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 import java.util.HashSet;
2
3 public class CommonAlphabetsASCII {
4     public static void main(String[] args) {
5         String input1 = "abc";
6         String input2 = "bc";
7
8         HashSet<Character> set1 = new HashSet<>();
9         for (char ch : input1.toCharArray()) {
10             set1.add(ch);
11         }
12
13         HashSet<Character> set2 = new HashSet<>();
14         for (char ch : input2.toCharArray()) {
15             set2.add(ch);
16         }
17
18         int sum1 = 0;
19         for (char ch : set1) {
20             if (set2.contains(ch)) {
21                 sum1 += (int) ch;
22             }
23         }
24
25         int singleDigitSum = getSingleDigitSum(sum1);
26         System.out.println(singleDigitSum);
27     }
28
29     private static int getSingleDigitSum(int num) {
30         while (num > 9) {
31             int sum = 0;
32             while (num > 0) {
33                 sum += num % 10;
34                 num /= 10;
35             }
36             num = sum;
37         }
38         return num;
39     }
40 }
41

```

	Input	Expected	Got
✓	a b c b c	8	8 ✓

Passed all tests! ✓

CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show all questions on one page

Finish review

Question 3

Correct

Marked out
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 SeigOlOnhceT Erolagnab".

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

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

NOTE:

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

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

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

Examples:

S. No.	input1	input2	output
1	Wipro Technologies Bangalore	0	orpiW seigolOnhceT eroLagnab
2	Wipro Technologies, Bangalore	0	orpiW ,seigolOnhceT eroLagnab
3	Wipro Technologies Bangalore	1	Orpiw SeigolOnhceT Erolagnab
4	Wipro Technologies, Bangalore	1	Orpiw ,seigolOnhceT Erolagnab

For example:

Input	Result
Wipro Technologies Bangalore 0	orpiW seigolOnhceT eroLagnab
Wipro Technologies, Bangalore 0	orpiW ,seigolOnhceT eroLagnab
Wipro Technologies Bangalore 1	Orpiw SeigolOnhceT Erolagnab
Wipro Technologies, Bangalore 1	Orpiw ,seigolOnhceT Erolagnab

Answer: (penalty regime: 0 %)

```

1 import java.util.Scanner;
2
3 public class SentenceReverser {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6
7         String s = sc.nextLine();
8
9         int c = 0;
10        try {
11            c = sc.nextInt();
12        } catch (Exception e) {
13            System.out.println("ERROR: Please enter a valid integer for the mode (1 or 0).");
14            sc.close();
15            return;
16        }
17
18        String[] w = s.split(" ");
19        StringBuilder res = new StringBuilder();
20
21        for (String wd : w) {
22            StringBuilder rw = new StringBuilder(wd);
23            rw.reverse();
24
25            if (c == 1) {
26                StringBuilder fw = new StringBuilder();
27                for (int i = 0; i < wd.length(); i++) {
28                    char r = rw.charAt(i);
29                    char o = wd.charAt(i);
30
31                    if (Character.isAlphabetic(o)) {
32                        if (Character.isUpperCase(o)) {
33                            fw.append(Character.toUpperCase(r));
34                        } else {
35                            fw.append(Character.toLowerCase(r));
36                        }
37                    } else {
38                        fw.append(r);
39                    }
40                }
41                res.append(fw).append(" ");
42            } else {
43                res.append(rw).append(" ");
44            }
45        }
46
47        System.out.println(res.toString().trim());
48        sc.close();
49    }
50 }
```

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! ✓