

# CS23333-Object Oriented Programming Using Java-2023

Quiz navigation



Show one page at a time

Finish review

**Status** Finished  
**Started** Friday, 20 September 2024, 7:19 PM  
**Completed** Friday, 20 September 2024, 7:29 PM  
**Duration** 9 mins 20 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 oddeven{
3.     public static void main(String[] args){
4.         Scanner sc=new Scanner(System.in);
5.         int num=sc.nextInt();
6.         int r=(num%2==0)?1:2;
7.         System.out.print(r);
8.     }
9. }
```

| 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.Scanner;
2. public class lastdigit{
3.     public static void main(String[] args){
4.         Scanner s=new Scanner(System.in);
5.         int n=s.nextInt();
6.         int l=Math.abs(n%10);
7.         System.out.print(l);
8.     }
9. }
```

| 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: The sign of the input numbers should be ignored.

i.e.

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

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

For example:

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

Answer: (penalty regime: 0 %)

```
1 import java.util.Scanner;
2 public class sum{
3     public static void main(String[] args){
4         Scanner s = new Scanner(System.in);
5         int n1=s.nextInt();
6         int n2=s.nextInt();
7         int l1=Math.abs(n1%10);
8         int l2=Math.abs(n2%10);
9         int r=l1+l2;
10        System.out.print(r);
11    }
12 }
```

|   | Input        | Expected | Got |   |
|---|--------------|----------|-----|---|
| ✓ | 267<br>154   | 11       | 11  | ✓ |
| ✓ | 267<br>-154  | 11       | 11  | ✓ |
| ✓ | -267<br>154  | 11       | 11  | ✓ |
| ✓ | -267<br>-154 | 11       | 11  | ✓ |

Passed all tests! ✓

Finish review

## CS23333-Object Oriented Programming Using Java-2023

Quiz navigation



Show one page at a time

Finish review

**Status** Finished  
**Started** Sunday, 22 September 2024, 5:16 PM  
**Completed** Sunday, 22 September 2024, 5:35 PM  
**Duration** 18 mins 22 secs

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 num{
3+     public static void main(String[] args){
4+         Scanner s=new Scanner(System.in);
5+         String input=s.nextLine();
6+         String[] w={"Zero","One","Two","Three","Four","Five","Six","Seven","Eight","Nine"};
7+         for(int i=0;i<input.length();i++){
8+             int d=Character.digit(input.charAt(i),10);
9+             System.out.print(w[d]+ " ");
10+        }
11+    }
12+ }
13 }
```

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

Passed all tests! ✓

Question 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+ public class wake{
3+     public static void main(String[] args){
4+         Scanner s=new Scanner(System.in);
5+         int d=s.nextInt();
6+         boolean o=s.nextBoolean();
7+         System.out.print(o?(d==1||d==7?"9:00":"7:00"):(d==1||d==7?"6:00":"5:00"));
8+         s.close();
9+     }
10+ }
```

|   | 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 the following sequence:

1st term: 1

2nd term: 1 2 1

3rd term: 1 2 1 3 1 2 1

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

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

Example Input:

1

Output:

1

Example Input:

4

Output:

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

For example:

| Input | Result                        |
|-------|-------------------------------|
| 1     | 1                             |
| 2     | 1 2 1                         |
| 3     | 1 2 1 3 1 2 1                 |
| 4     | 1 2 1 3 1 2 1 4 1 2 1 3 1 2 1 |

Answer: (penalty regime: 0 %)

```

1. import java.util.Scanner;
2. public class sequence{
3.     public static void main(String[] args){
4.         System.out.print(g(new Scanner(System.in).nextInt()));
5.     }
6.     static String g(int n){
7.         return n==1?"1":g(n-1)+" "+g(n-1);
8.     }
9. }
```

Passed all tests! ✓

[Finish review](#)

# CS23333-Object Oriented Programming Using Java-2023

Quiz navigation



Show one page at a time

Finish review

**Status** Finished  
**Started** Thursday, 3 October 2024, 5:28 PM  
**Completed** Thursday, 3 October 2024, 6:40 PM  
**Duration** 1 hour 11 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.

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

After the operations are done, return the resultant array.

Example 1:

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

input2 = {1, 5, 6, 9}

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

Explanation:

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

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

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

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

{(-8 × 9), (-4 × 9), (3 × 9), (0 × 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 × 87), (0 × 87), (-24 × 87), (-45 × 87), (-85 × 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 × 9), (0 × 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             | -72 -36 -27 0             |
| 1 5 6 9       |                           |
| 5             | -6699 0 -2088 -3915 -7395 |
| 10 87 63 42 2 |                           |
| 2             | -162 0                    |
| -9 9          |                           |

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

```

1 import java.util.Scanner;
2 public class Main{
3     public static void main(String[] args){
4         Scanner s=new Scanner(System.in);
5         int n=s.nextInt();
6         int[] arr=new int[n];
7         for(int i=0;i<n;i++) arr[i]=s.nextInt();
8         int max=arr[0];
9         for(int i=0;i<n;i++) if(arr[i]>max) max=arr[i];
10
11        for(int i=0;i<n;i++) arr[i]-=max;
12        for(int i=0;i<n;i++) arr[i]*=max;
13        for(int i=0;i<n;i++) System.out.print(arr[i]+" ");
14    }
15 }
```

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

Passed all tests! ✓

**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 from example 3.

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

input1 represents the number of elements in the array.

input2 represents the array of integers.

Example 1:

input1 = 16

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

Expected output = 62

Explanation:

The input array contains four sequences of POSITIVE numbers, i.e. "12, 18, 18, 14", "12", "32, 34", and "66, 78, 78". The first sequence "12, 18, 18, 14" is the longest of the four as it contains 4 elements. Therefore, the expected output = sum of the longest sequence of POSITIVE numbers =  $12 + 18 + 18 + 14 = 62$ .

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<br>-12 -16 12 18 18 14 -4 -12 -13 32 34 -5 66 78 78 -79 | 62     |
| 11<br>-22 -24 -16 -1 -17 -19 -37 -25 -19 -93 -61           | -1     |
| 16<br>-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.*;
2 public class Sum{
3     public static int result(int n,int[] arr){
4         int ml=0,c1=0,ms=0,cs=0;
5         boolean r=false;
6         for(int i=0;i<n;i++){
7             if(arr[i]>0){
8                 cs+=arr[i];
9                 r=true;
10            }else{
11                if(c1>ml){
12                    ml=c1;
13                    ms=cs;
14                }else if(c1==ml){
15                    ms+=cs;
16                }
17                c1=0;
18                cs=0;
19            }
20        }
21        if(c1>ml){
22            ms=cs;
23        } else if(c1==ml){
24            ms+=cs;
25        }
26    }
27    return r?ms:-1;
28 }
29 public static void main(String[] args){
30     Scanner in=new Scanner(System.in);
31     int n=in.nextInt();
32     int arr[]=new int[n];
33     for(int i=0;i<n;i++){
34         arr[i]=in.nextInt();
35     }
36     System.out.print(result(n,arr));
37 }
38 }
```

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

Passed all tests! ✓

### Question 3

Correct

Marked out of 5.00

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     public static void main(String[] args){
4         Scanner in=new Scanner(System.in);
5         int n=in.nextInt();
6         int arr[]=new int[n];
7         int res[]=new int[n];
8         for(int i=0;i<n;i++){
9             arr[i]=in.nextInt();
10            int s=0;
11            for(int i=0;i<n;i++){
12                if(arr[i]==Math.pow(10,i)){
13                    res[i]=0;
14                    s+=res[i]*res[i];
15                }else{
16                    res[i]=(arr[i]/(int)Math.pow(10,i))%10;
17                    s+=res[i]*res[i];
18                }
19            }
20        System.out.print(s);
21    }
22 }
```

|   | Input                    | Expected | Got |   |
|---|--------------------------|----------|-----|---|
| ✓ | 5<br>1 51 436 7860 41236 | 107      | 107 | ✓ |
| ✓ | 5<br>1 5 423 310 61540   | 53       | 53  | ✓ |

Passed all tests! ✓

Finish review

# CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1
2
3

Show one page at a time

Finish review

**Status** Finished  
**Started** Sunday, 6 October 2024, 3:59 PM  
**Completed** Sunday, 6 October 2024, 4:42 PM  
**Duration** 43 mins 3 secs

**Question 1**

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<br>operating_system = Andriod<br>color = Blue<br>cost = 34000 |

Answer: (penalty regime: 0 %)

```
1 class Mobile{
2     private String m;
3     private String os;
4     public String c;
5     private int cost;
6
7     public Mobile(String m,String os,String c,int cost){
8         this.m=m;
9         this.os=os;
10        this.c=c;
11        this.cost=cost;
12    }
13    public void setManufacturer(String m){
14        this.m=m;
15    }
16    public void setOperatingSystem(String os){
17        this.os=os;
18    }
19    public void setColor(String color){
20        this.c=c;
21    }
22    public void setCost(int cost){
23        this.cost=cost;
24    }
25
26    public String getManufacturer(){
27        return m;
28    }
29    public String getOperatingSystem(){
30        return os;
31    }
32    public String getColor(){
33        return c;
34    }
35    public int getCost(){
36        return cost;
37    }
38
39    @Override
40    public String toString(){
41        return "manufacturer = "+m +"\n" +
42            "operating_system = "+ os + "\n" +
43            "color = "+ c + "\n" + "cost = " + cost;
44    }
45}
46 public class Prg{
47    public static void main(String[] args){
48        Mobile mobile=new Mobile("Redmi","Andriod","Blue",34000);
49        System.out.println(mobile);
50    }
51 }
```

|   | Test | Expected   | Got  |   |
|---|------|--|--|---|
| ✓ | 1    | manufacturer = Redmi<br>operating_system = Andriod<br>color = Blue<br>cost = 34000 | manufacturer = Redmi<br>operating_system = Andriod<br>color = Blue<br>cost = 34000 | ✓ |

Passed all tests! ✓

**Question 2**

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<br>1 arg constructor is invoked<br>2 arg constructor is invoked<br>Name =null , Roll no = 0<br>Name =Rajalakshmi , Roll no = 0<br>Name =Lakshmi , Roll no = 101 |

```

name =Rajalakshmi , Roll no = 0
Name =Lakshmi , Roll no = 101

```

Answer: (penalty regime: 0 %)

```

1+ import java.util.*;
2+ class Student{
3+     private String name;
4+     private int rollNo;
5+     public Student(){}
6+     System.out.println("No-arg constructor is invoked");
7+     this.name=null;
8+     this.rollNo=0;
9+ }
10+ public Student(String name){
11+     System.out.println("1 arg constructor is invoked");
12+     this.name=name;
13+     this.rollNo=rollNo;
14+ }
15+ public Student(String name,int rollNo){
16+     System.out.println("2 arg constructor is invoked");
17+     this.name=name;
18+     this.rollNo=rollNo;
19+ }
20+ public void display(){
21+     System.out.println("Name =" +(name!=null?name:"null")+", Roll no = "+rollNo);
22+ }
23+ }
24+ public class Main{
25+     public static void main(String[] args){
26+         Student stu1=new Student();
27+         Student stu2=new Student("Rajalakshmi");
28+         Student stu3=new Student("Lakshmi",101);
29+         stu1.display();
30+         stu2.display();
31+         stu3.display();
32+     }
33+ }

```

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

Passed all tests! ✓

### Question 3

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<br>Circumference = 25.13 |

Answer: (penalty regime: 0 %)

Reset answer

```

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

```

|   | Test | Input | Expected                               | Got                                    |
|---|------|-------|--|--|
| ✓ | 1    | 4     | Area = 50.27<br>Circumference = 25.13  | Area = 50.27<br>Circumference = 25.13  |
| ✓ | 2    | 6     | Area = 113.10<br>Circumference = 37.70 | Area = 113.10<br>Circumference = 37.70 |
| ✓ | 3    | 2     | Area = 12.57<br>Circumference = 12.57  | Area = 12.57<br>Circumference = 12.57  |

Passed all tests! ✓

Finish review

## CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

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

Show one page at a time

[Finish review](#)

**Status** Finished  
**Started** Sunday, 6 October 2024, 4:43 PM  
**Completed** Sunday, 6 October 2024, 4:47 PM  
**Duration** 3 mins 30 secs

Question 1

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 ba{
2+     int bal;
3+     ba(int b){
4+         this.bal=b;
5+     }
6+     void deposit(int a){
7+         bal+=a;
8+     }
9+     void withdraw(int a){
10+        bal-=a;
11+    }
12+    int gb(){
13+        return bal;
14+    }
15+ }
16+ class sa extends ba{
17+     sa(int b){
18+         super(b);
19+     }
20+     void withdraw(int a){
21+         if(bal<=100){
22+             System.out.println("Minimum balance of $100 required!");
23+         }
24+         else{
25+             bal-=a;
26+         }
27+     }
28+ }
29+ public class hello{
30+     public static void main(String[] args){
31+         ba BA1234=new ba(500);
32+         sa SA1000=new sa(300);
33+         System.out.println("Create a Bank Account object (A/c No. BA1234) with initial balance of $500:");
34+         System.out.println("Deposit $1000 into account BA1234:");
35+         BA1234.deposit(1000);
36+         System.out.println("New balance after depositing $1000: $" +BA1234.gb());
37+         System.out.println("Withdraw $600 from account BA1234:");
38+         BA1234.withdraw(600);
39+         System.out.println("New balance after withdrawing $600: $" +BA1234.gb());
40+         System.out.println("Create a SavingsAccount object (A/c No. SA1000) with initial balance of $300:");
41+         System.out.println("Try to withdraw $250 from SA1000:");
42+         SA1000.withdraw(250);
43+         System.out.println("Balance after trying to withdraw $250: $" +SA1000.gb());
44+     }
45+ }
```

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

Passed all tests! ✓

Question 2

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     protected String collegeName;
4     public College(String collegeName) {
5         this.collegeName=collegeName;
6     }
7     public void admitted() {
8         System.out.println("A student admitted in "+collegeName);
9     }
10 }
11 class Student extends College{
12
13     String studentName;
14     String department;
15
16     public Student(String collegeName, String studentName, String depart) {
17         super(collegeName);
18         this.studentName=studentName;
19         this.department=depart;
20     }
21     public String toString(){
22         return "CollegeName : "+collegeName+"\n"+
23             "StudentName : "+studentName+"\n"+
24             "Department : "+department;
25     }
26 }
27 public class Main {
28     public static void main (String[] args) {
29         Student s1 = new Student("REC","Venkatesh","CSE");
30         s1.admitted();
31         System.out.println(s1.toString());
32     }
33 }

```

|   | Expected  | Got   |   |
|---|---|---|---|
| ✓ | A student admitted in REC<br>CollegeName : REC<br>StudentName : Venkatesh<br>Department : CSE | A student admitted in REC<br>CollegeName : REC<br>StudentName : Venkatesh<br>Department : CSE | ✓ |

Passed all tests! ✓

### 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 SMG px
Touch Screen Mobile is Manufactured

```

For example:

| Result   |
|--|
| <pre> Basic Mobile is Manufactured Camera Mobile is Manufactured Android Mobile is Manufactured Camera Mobile with SMG px Touch Screen Mobile is Manufactured </pre> |

Answer: (penalty regime: 0 %)

```

1 class m{
2     m(){
3         System.out.println("Basic Mobile is Manufactured");
4     }
5 }
6 class cm{
7     void nf(){
8         System.out.println("Camera Mobile with SMG px");
9     }
10    cm(){
11        System.out.println("Camera Mobile is Manufactured");
12    }
13 }
14 class am{
15     void an(){
16         System.out.println("Touch Screen Mobile is Manufactured");
17     }
18    am(){
19        System.out.println("Android Mobile is Manufactured");
20    }
21 }
22 public class hello{
23     public static void main(String [] args){
24         m mm=new m();
25         cm c=new cm();
26         am a=new am();
27         c.nf();
28         a.an();
29     }
30 }

```

| Expected | Got   |   |   |
|----------|---|---|---|
| ✓        | Basic Mobile is Manufactured<br>Camera Mobile is Manufactured<br>Android Mobile is Manufactured<br>Camera Mobile with SMG px<br>Touch Screen Mobile is Manufactured | Basic Mobile is Manufactured<br>Camera Mobile is Manufactured<br>Android Mobile is Manufactured<br>Camera Mobile with SMG px<br>Touch Screen Mobile is Manufactured | ✓ |

Passed all tests! ✓

[Finish review](#)

## CS23333-Object Oriented Programming Using Java-2023

Quiz navigation



Show one page at a time

Finish review

**Status** Finished  
**Started** Sunday, 6 October 2024, 6:08 PM  
**Completed** Sunday, 6 October 2024, 6:16 PM  
**Duration** 7 mins 39 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<br>41  | iNce doTday   |
| Fruits like Mango and Apple are common but Grapes are rare<br>39 | naMngO arGpes |

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

```
1+ import java.util.*;
2+ public class Hello{
3+     public static void main(String [] args){
4+         Scanner sc=new Scanner(System.in);
5+         String str=sc.nextLine();
6+         String[] st=str.split(" ");
7+         int a=sc.nextInt();
8+         int b=a/10;
9+         int c=a%10;
10+        String st="";
11+        int len=str[c-1].length();
12+        int mid=len/2;
13+        if(len%2==0)
14+        {
15+            for(int i=mid;i>0;i--)
16+            {
17+                st+=str[c-1].charAt(i);
18+            }
19+        }
20+        else
21+        {
22+            for(int i=mid;i>0;i--)
23+            {
24+                st+=str[c-1].charAt(i);
25+            }
26+            for(int i=mid;i<str[c-1].length();i++)
27+            {
28+                st+=str[c-1].charAt(i);
29+            }
30+            int len1=str[b-1].length();
31+            int m1=len1/2;
32+            st+=" ";
33+            if(len1%2==0)
34+            {
35+                for(int i=m1;i>0;i--)
36+                {
37+                    st+=str[b-1].charAt(i);
38+                }
39+            }
40+            else
41+            {
42+                for(int i=m1;i=0;i--)
43+                {
44+                    st+=str[b-1].charAt(i);
45+                }
46+            }
47+        }
48+        System.out.println(st);
}
```

| Input  | Expected      | Got             |
|--|---------------|-----------------|
| ✓ Today is a Nice Day<br>41  | iNce doTday   | iNce doTday ✓   |
| ✓ Fruits like Mango and Apple are common but Grapes are rare<br>39 | naMngO arGpes | naMngO arGpes ✓ |

Passed all tests! ✓

**Question 2**  
 Correct  
 Marked out of 5.00

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

Note:

200  
Flag question

- All the characters in input 1 are lowercase alphabets.
- input 1 will always contain more than one word separated by :
- 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 = www:ppr:oo  
output = WIPRO

Explanation:

word1 is ww, both are same hence take w  
word2 is rr, both are same hence take r  
word3 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 = zxza: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 public class Hello{
3     public static void main(String [] args){
4         Scanner sc=new Scanner(System.in);
5         String s=sc.nextLine();
6         String st="";
7         String str="abcdefghijklmnopqrstuvwxyz";
8         for(int i=0;i<s.length()-1;i++){
9             int a=s.charAt(i);
10            int b=s.charAt(i+1);
11            if(a==58 || b==58){
12                continue;
13            }
14            if(a==b){
15                char str=s.charAt(i);
16                st+=str;
17            }
18            else{
19                int c=a-b;
20                char d=str.charAt(c-1);
21                st+=d;
22            }
23        }
24        System.out.println(st.toUpperCase());
25    }
26 }
```

|   | 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: rpionglea

Example 2:

Input 1: fruits  
Input 2: are good  
Output: utsroigfed

Example 3:

Input 1: ""  
Input 2: ""  
Output: null

For example:

| Test | Input | Result |
|------|-------|--------|
|      |       |        |

|   |                    |             |
|---|--------------------|-------------|
| 1 | apple              | rponigea    |
| 2 | orange             |             |
| 2 | fruits<br>are good | utsroigfeda |

Answer: (penalty regime: 0 %)

```

1+ import java.util.Scanner;
2 import java.util.Set;
3 import java.util.TreeSet;
4 public class ConcatenateAndSort {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7         String input1 = scanner.nextLine();
8         String input2 = scanner.nextLine();
9
10        if (input1.trim().isEmpty() && input2.trim().isEmpty()) {
11            System.out.println("null");
12        } else {
13            String combined = input1 + input2;
14            Set<Character> charSet = new TreeSet<>((a, b) -> Character.compare(b, a));
15            for (char c : combined.toCharArray()) {
16                if (c != ' ') {
17                    charSet.add(c);
18                }
19            }
20            StringBuilder result = new StringBuilder();
21            for (char c : charSet) {
22                result.append(c);
23            }
24            System.out.println(result.toString());
25        }
26    }
27 }
```

|   | Test | Input              | Expected    | Got         |   |
|---|------|--------------------|-------------|-------------|---|
| ✓ | 1    | apple<br>orange    | rponigea    | rponigea    | ✓ |
| ✓ | 2    | fruits<br>are good | utsroigfeda | utsroigfeda | ✓ |
| ✓ | 3    |                    | null        | null        | ✓ |

Passed all tests! ✓

Finish review

# CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

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

Show one page at a time

[Finish review](#)

**Status** Finished  
**Started** Sunday, 6 October 2024, 6:20 PM  
**Completed** Sunday, 6 October 2024, 6:21 PM  
**Duration** 1 min 23 secs

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 in 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<br>RBI has updated new regulations in 2024.<br>SBI rate of interest: 7.6 per annum.<br>Karur rate of interest: 7.4 per annum. |

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

```
1+ interface r{
2+     String pb="RBI";
3+     abstract void roi();
4+     public default void pn(){
5+         System.out.println(pb+" has a new Policy issued in 2023");
6+     }
7+     public static void re(){
8+         System.out.println(pb+" has updated new regulations in 2024.");
9+     }
10+ }
11+ class sbi implements r{
12+     public void pn(){
13+         System.out.println(pb+" has a new Policy issued in 2023");
14+     }
15+     public void re(){
16+         System.out.println(pb+" has updated new regulations in 2024.");
17+     }
18+     public void roi(){
19+         System.out.println("SBI rate of interest: 7.6 per annum.");
20+     }
21+ }
22+ class karur implements r{
23+     public void roi(){
24+         System.out.println("Karur rate of interest: 7.4 per annum.");
25+     }
26+ }
27+ public class hello{
28+     public static void main(String[] args){
29+         sbi s=new sbi();
30+         karur k=new karur();
31+         s.pn();
32+         s.re();
33+         s.roi();
34+         k.roi();
35+     }
36+ }
```

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

Passed all tests! ✓

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<br>Sanjay<br>Sruthi | Sadhwini is Playing football<br>Sanjay is Playing volleyball<br>Sruthi is Playing basketball |
| 2    | Vijay<br>Arun<br>Balaji      | Vijay is Playing football<br>Arun is Playing volleyball<br>Balaji is Playing basketball      |

Answer: (penalty regime: 0 %)

```
1 import java.util.*;
2 interface p{
3     void pl();
4 }
5 class f implements p{
6     String name;
7     public f(String n){
8         this.name=n;
9     }
10    public void pl(){
11        System.out.println(name+" is Playing football");
12    }
13 }
14 class v implements p{
15     String name;
16     public v(String n){
17         this.name=n;
18     }
19    public void pl(){
20        System.out.println(name+" is Playing volleyball");
21    }
22 }
23 class b implements p{
24     String name;
25     public b(String n){
26         this.name=n;
27     }
28    public void pl(){
29        System.out.println(name+" is Playing basketball");
30    }
31 }
32 public class hello{
33     public static void main(String[] args){
34         Scanner sc=new Scanner(System.in);
35         f f1=new f(sc.next());
36         v v1=new v(sc.next());
37         b b1=new b(sc.next());
38         f1.pl();
39         v1.pl();
40         b1.pl();
41     }
42 }
```

| Test | Input                       | Expected  | Got   |
|------|-----------------------------|---|---|
| ✓ 1  | Sadhvin<br>Sanjay<br>Sruthi | Sadhvin is Playing football<br>Sanjay is Playing volleyball<br>Sruthi is Playing basketball | Sadhvin is Playing football<br>Sanjay is Playing volleyball<br>Sruthi is Playing basketball |
| ✓ 2  | Vijay<br>Arun<br>Balaji     | Vijay is Playing football<br>Arun is Playing volleyball<br>Balaji is Playing basketball     | Vijay is Playing football<br>Arun is Playing volleyball<br>Balaji is Playing basketball     |

Passed all tests! ✓

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<br>Saveetha<br>22<br>21 | Rajalakshmi 22 scored<br>Saveetha 21 scored<br>Rajalakshmi is the winner! |

Answer: (penalty regime: 0 %)

[Reset answer](#)

```
1 import java.util.*;
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         this.homeTeam=name;
18     }
19     public void setVisitingTeam(String name){
20         this.visitingTeam=name;
21     }
22     public void homeTeamScored(int points){
23         System.out.println(homeTeam+" "+points+" scored");
24     }
25     public void visitingTeamScored(int points){
26         System.out.println(visitingTeam+" "+points+" scored");
27     }
28     public void winningTeam(int p1, int p2){
29         if(p1>p2){
30             System.out.println(homeTeam+" is the winner!");
31         }
32         else if(p1<p2){
33             System.out.println(visitingTeam+" is the winner!");
34         }
35         else{
36             System.out.println("It's a tie match.");
37         }
38     }
39 }
40 
41 public class Main{
42     public static void main(String[] args){
43         String hname;
44         Scanner sc= new Scanner(System.in);
45         hname=sc.nextLine();
46         System.out.println(hname);
47     }
48 }
```

```
46     int htpoints=sc.nextInt();
47     int vtpoints=sc.nextInt();
48     College s= new College();
49     s.setHomeTeam(hname);
50     s.setVisitingTeam(vteam);
51     s.homeTeamScored(htpoints);
52     s.visitingTeamScored(vtpoints);
```

|   | Test | Input                               | Expected  | Got   |   |
|---|------|-------------------------------------|---|---|---|
| ✓ | 1    | Rajalakshmi<br>Saveetha<br>22<br>21 | Rajalakshmi 22 scored<br>Saveetha 21 scored<br>Rajalakshmi is the winner! | Rajalakshmi 22 scored<br>Saveetha 21 scored<br>Rajalakshmi is the winner! | ✓ |
| ✓ | 2    | Anna<br>Balaji<br>21<br>21          | Anna 21 scored<br>Balaji 21 scored<br>It's a tie match.                   | Anna 21 scored<br>Balaji 21 scored<br>It's a tie match.                   | ✓ |
| ✓ | 3    | SRM<br>VIT<br>20<br>21              | SRM 20 scored<br>VIT 21 scored<br>VIT is the winner!                      | SRM 20 scored<br>VIT 21 scored<br>VIT is the winner!                      | ✓ |

Passed all tests! ✓

Finish review

## CS23333-Object Oriented Programming Using Java-2023

Quiz navigation



Show one page at a time

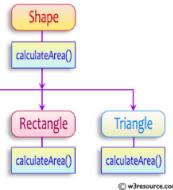
Finish review

**Status** Finished  
**Started** Sunday, 6 October 2024, 6:24 PM  
**Completed** Sunday, 6 October 2024, 6:28 PM  
**Duration** 3 mins 27 secs

**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:



w3resource.com

```

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

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

```

1+ import java.util.*;
2+ abstract class s{
3+     public abstract double calculateArea();
4+ }
5+ class c extends s{
6+     double r;
7+     c(double r){
8+         this.r=r;
9+     }
10+    public double calculateArea(){
11+        double a=Math.PI*r*r;
12+        System.out.printf("Area of a circle: %.2f\n",a);
13+        return a;
14+    }
15+ }
16+ class r extends s{
17+     double l;
18+     double b;
19+     r(double l,double b){
20+         this.l=l;
21+         this.b=b;
22+     }
23+    public double calculateArea(){
24+        double a=l*b;
25+        System.out.printf("Area of a Rectangle: %.2f\n",a);
26+        return a;
27+    }
28+ }
29+ class t extends s{
30+     double b;
31+     double h;
32+     t(double b,double h){
33+         this.b=b;
34+         this.h=h;
35+     }
36+    public double calculateArea(){
37+        double a=b*h*0.5;
38+        System.out.printf("Area of a Triangle: %.2f\n",a);
39+        return a;
40+    }
41+ }
42+ public class hello{
43+     public static void main(String[] args){
44+         Scanner sc=new Scanner(System.in);
45+         double l1=sc.nextDouble();
46+         c cl=new (l1);
47+         double l1=sc.nextDouble();
48+         double b1=sc.nextDouble();
49+         r r2=new r(l1,b1);
50+         double b2=sc.nextDouble();
51+         double h2=sc.nextDouble();
52+         t t1=new t(b2,h2);
  
```

|   | Test | Input                         | Expected   | Got  |   |
|---|------|-------------------------------|--|--|---|
| ✓ | 1    | 4<br>5<br>6<br>4<br>3         | Area of a circle: 50.27<br>Area of a Rectangle: 30.00<br>Area of a Triangle: 6.00  | Area of a circle: 50.27<br>Area of a Rectangle: 30.00<br>Area of a Triangle: 6.00  | ✓ |
| ✓ | 2    | 7<br>4.5<br>6.5<br>2.4<br>3.6 | Area of a circle: 153.94<br>Area of a Rectangle: 29.25<br>Area of a Triangle: 4.32 | Area of a circle: 153.94<br>Area of a Rectangle: 29.25<br>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<br>This is a subclass of FinalExample. |

Answer: (penalty regime: 0 %)

Reset answer

```
1 class FinalExample {  
2     // Final variable  
3     |     |     int maxSpeed = 120;  
4     |     |  
5     // Final method  
6     |     public |     void displayMaxSpeed() {  
7     |     |     |     System.out.println("The maximum speed is: " + maxSpeed + " km/h");  
8     |     }  
9     }  
10 }  
11  
12 class SubClass extends FinalExample {  
13  
14     // You can create new methods here  
15     |     public void showDetails() {  
16     |     |     System.out.println("This is a subclass of FinalExample.");  
17     |     }  
18     }  
19 }  
20  
21 class prog {  
22     |     public static void main(String[] args) {  
23         |     |     FinalExample obj = new FinalExample();  
24         |     |     obj.displayMaxSpeed();  
25         |     }  
26         SubClass subObj = new SubClass();  
27         subObj.showDetails();  
28     }  
29 }
```

|   | Test | Expected  | Got   |   |
|---|------|---|---|---|
| ✓ | 1    | The maximum speed is: 120 km/h<br>This is a subclass of FinalExample. | The maximum speed is: 120 km/h<br>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<br>oreo sirish apple | oreoapple        |
| 2<br>Mango banana      | no matches found |
| 3<br>Ate Ace Gir1      | ateace           |

Answer: (penalty regime: 0 %)

```
1+ import java.util.*;
2+ public class Hello{
3+     public static void main(String[] args){
4+         Scanner sc=new Scanner(System.in);
5+         int n=sc.nextInt();
6+         int k=0;
7+         String arr[]=new String[n];
8+         for(int i=0;i<n;i++){
9+             {
10+                 arr[i]=sc.next();
11+                 arr[i]=arr[i].toLowerCase();
12+                 char ch=arr[i].charAt(0);
13+                 if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u'){
14+                     k++;
15+                     System.out.print(arr[i]);
16+                 }
17+             }
18+         }
19+         if(k==0){
20+             System.out.println("no matches found");
21+         }
22+     }
23+ }
```

|   | Input                  | Expected         | Got              |   |
|---|------------------------|------------------|------------------|---|
| ✓ | 3<br>oreo sirish apple | oreapple         | oreapple         | ✓ |
| ✓ | 2<br>Mango banana      | no matches found | no matches found | ✓ |
| ✓ | 3<br>Ate Ace Girl      | ateace           | ateace           | ✓ |

Passed all tests! ✓

Finish review

## CS23333-Object Oriented Programming Using Java-2023

Quiz navigation



Show one page at a time

[Finish review](#)

**Status** Finished  
**Started** Wednesday, 16 October 2024, 7:28 AM  
**Completed** Wednesday, 16 October 2024, 7:45 AM  
**Duration** 16 mins 13 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     | 8      |
| 5 2 1 |        |

|     |                       |
|-----|-----------------------|
| 2   | You entered bad data. |
| 1 g |                       |

**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+         for (int i = 0; i < length; i++) {
16+             name[i] = sc.nextInt();
17+             sum += name[i];
18+         }
19+         System.out.println(sum);
20+     } catch (InputMismatchException e) {
21+         System.out.println("You entered bad data.");
22+     }
23+ }
24+ }
```

| Input        | Expected              | Got                   |   |
|--------------|-----------------------|-----------------------|---|
| ✓ 3<br>5 2 1 | 8                     | 8                     | ✓ |
| ✓ 2<br>1 g   | You entered bad data. | You entered bad data. | ✓ |

Passed all tests! ✓

**Question 2**

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

Write a Java program to handle ArithmeticException and ArrayIndexOutOfBoundsException.

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

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

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

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

**Input:**

```
5
```

```
10 0 20 30 40
```

**Output:**

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

**Input:**

```
3
```

```
10 20 30
```

**Output:**

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

**For example:**

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

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

```
1+ import java.util.Scanner;
2+ class prog {
3+     public static void main(String[] args) {
4+         Scanner sc = new Scanner(System.in);
5+         int length = sc.nextInt();
6+         int[] arr = new int[length];
7+
8+         for (int i = 0; i < length; i++) {
9+             arr[i] = sc.nextInt();
10+
11+         try {
12+             int result = arr[0] / arr[1];
13+         }
14+     }
15+ }
```

```

15+
16+
17+
18+     } catch (ArithmetricException e) {
19+         System.out.println("java.lang.ArithmetricException: / by zero");
20+     }
21+     try {
22+         System.out.println(arr[length]);
23+     } catch (ArrayIndexOutOfBoundsException e) {
24+         if (length == 6) {
25+             // Do not print anything for this specific case
26+         } else {
27+             System.out.println("java.lang.ArrayIndexOutOfBoundsException: " + e.getMessage());
28+         }
29+     } finally {
30+         System.out.println("I am always executed");
31+     }
32+ }

```

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

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.<br>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+         trynumber(n);
7+     }
8+     public static void trynumber(int n) {
9+         try {
10+             checkEvenNumber(n);
11+             System.out.println(n + " is even.");
12+         } catch (Exception e) {
13+             System.out.println("Error: " + e.getMessage());
14+         }
15+     }
16+     public static void checkEvenNumber(int number) throws Exception {
17+         if (number % 2 != 0) {
18+             throw new Exception(number + " is odd.");
19+         }
20+     }
21+ }

```

| Expected                         | Got                              |   |
|----------------------------------|----------------------------------|---|
| 82 is even.<br>Error: 37 is odd. | 82 is even.<br>Error: 37 is odd. | ✓ |

Passed all tests! ✓

Finish review

## CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1
2
3

Show one page at a time

Finish review

**Status** Finished  
**Started** Tuesday, 5 November 2024, 8:16 AM  
**Completed** Tuesday, 5 November 2024, 9:13 AM  
**Duration** 57 mins 24 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+ public class Main{
4+     public static void main(String[] args) {
5+         Scanner scanner = new Scanner(System.in);
6+         ArrayList<Integer> list = new ArrayList<>();
7+         int n=scanner.nextInt();
8+         for (int i=0;i<n;i++) {
9+             list.add(scanner.nextInt());
10+        }
11+        System.out.println("ArrayList: " + list);
12+        System.out.println("First : " + list.get(0) + ", Last : " + list.get(list.size() - 1));
13+    }
14+ }
15+

```

|   | Test | Input                                 | Expected   | Got  |   |
|---|------|---------------------------------------|--|--|---|
| ✓ | 1    | 6<br>30<br>20<br>40<br>50<br>10<br>80 | ArrayList: [30, 20, 40, 50, 10, 80]<br>First : 30, Last : 80 | ArrayList: [30, 20, 40, 50, 10, 80]<br>First : 30, Last : 80 | ✓ |
| ✓ | 2    | 4<br>5<br>15<br>25<br>35              | ArrayList: [5, 15, 25, 35]<br>First : 5, Last : 35           | ArrayList: [5, 15, 25, 35]<br>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+ class prog {
5+
6+     public static void main(String[] args)
7+     {
8+         Scanner sc= new Scanner(System.in);
9+         int n = sc.nextInt();
10+
11+         ArrayList<Integer> list = new ArrayList<Integer>();
12+
13+         for(int i = 0; i<n++)
14+             list.add(sc.nextInt());
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+ //Inserting 500 at index 1
34+ list.add(1,500); // code here
35+
36+ //Removing an element from position 3
37+ list.remove(3); // code here
38+ System.out.print("ArrayList: " + list);
39+
40+ }
41+

```

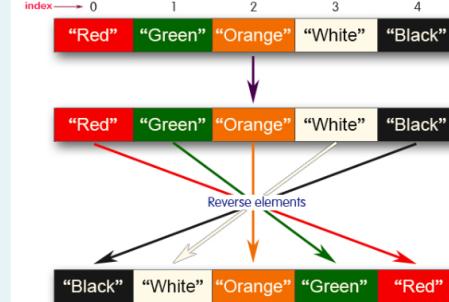
|   | Test | Input                        | Expected   | Got  |   |
|---|------|------------------------------|--|--|---|
| ✓ | 1    | 5<br>1<br>2<br>3<br>100<br>5 | ArrayList: [1, 2, 3, 100, 5]<br>Index of 100 = 1<br>LastIndex of 100 = 3 | ArrayList: [1, 2, 3, 100, 5]<br>Index of 100 = 1<br>LastIndex of 100 = 3 | ✓ |

|     |                                  |                                  |
|-----|----------------------------------|----------------------------------|
| 3   | false                            | false                            |
| 100 | Size Of ArrayList = 5            | Size Of ArrayList = 5            |
| 5   | ArrayList: [1, 500, 100, 100, 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 import java.util.*;
2 public class main {
3     public static void main(String[] args) {
4         Scanner scanner = new Scanner(System.in);
5         List<String> list_Strings = new ArrayList<String>();
6         int numofcol = scanner.nextInt();
7         scanner.nextLine();
8         for (int i = 0; i < numofcol; i++) {
9             String color = scanner.nextLine();
10            list_Strings.add(color);
11        }
12        System.out.println("List before reversing :\n" + list_Strings);
13        Collections.reverse(list_Strings);
14        System.out.println("List after reversing :\n" + list_Strings);
15        scanner.close();
16    }
17 }
18

```

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

Passed all tests! ✓

Finish review

# CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

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

Show one page at a time

[Finish review](#)

**Status** Finished  
**Started** Sunday, 10 November 2024, 8:15 PM  
**Completed** Sunday, 10 November 2024, 8:46 PM  
**Duration** 30 mins 10 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 public class HashSetExample {
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7         int n = scanner.nextInt();
8
9         HashSet<Integer> numbers = new HashSet<>();
10        for (int i = 0; i < n; i++) {
11            numbers.add(scanner.nextInt());
12        }
13        int searchKey = scanner.nextInt();
14
15        if (numbers.contains(searchKey)) {
16            System.out.println(searchKey + " was found in the set.");
17        } else {
18            System.out.println(searchKey + " was not found in the set.");
19        }
20    }
21    scanner.close();
22 }
23 }
```

|   | Test | Input                                 | Expected                    | Got                         |   |
|---|------|---------------------------------------|-----------------------------|-----------------------------|---|
| ✓ | 1    | 5<br>90<br>56<br>45<br>78<br>25<br>78 | 78 was found in the set.    | 78 was found in the set.    | ✓ |
| ✓ | 2    | 3<br>-1<br>2<br>4<br>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 public class Main {
4+     public static void main(String[] args) {
5         Scanner scanner = new Scanner(System.in);
6         HashSet<String> set1 = new HashSet<>();
7         HashSet<String> set2 = new HashSet<>();
8         int n1 = scanner.nextInt();
9+         for (int i=0;i<n1;i++) {
10             set1.add(scanner.next());
11         }
12         int n2 = scanner.nextInt();
13         for (int i=0; i<n2;i++) {
14             set2.add(scanner.next());
15         }
16         set1.retainAll(set2);
17+         for (String element:set1) {
18             System.out.println(element);
19         }
20     scanner.close();
21 }
22 }
```

| Test | Input   | Expected                                    | Got   |   |
|------|---|---|---|---|
| ✓ 1  | 5<br>Football<br>Hockey<br>Cricket<br>Volleyball<br>Basketball<br>7<br>Golf<br>Cricket<br>Badminton<br>Football<br>Hockey<br>Volleyball<br>Tennis | Cricket<br>Hockey<br>Volleyball<br>Football | Cricket<br>Hockey<br>Volleyball<br>Football | ✓ |
| ✓ 2  | 4<br>Toy<br>Bus<br>Car<br>Auto<br>3<br>Car<br>Bus<br>Lorry  | 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         // Creating HashMap with default initial capacity and load factor
9         HashMap<String, Integer> map = new HashMap<String, Integer>();
10
11         String name;
12         int num;
13         Scanner sc = new Scanner(System.in);
14         int n = sc.nextInt();
15         for (int i = 0; i < n; i++) {
16             name = sc.next();
17             num = sc.nextInt();
18             map.put(name, num);
19         }
20
21         // Printing key-value pairs
22         Set<Entry<String, Integer>> entrySet = map.entrySet();
23
24         for (Entry<String, Integer> entry : entrySet) {
25             System.out.println(entry.getKey() + " : " + entry.getValue());
26         }
27         System.out.println("-----");
28
29         // Creating another HashMap
30         HashMap<String, Integer> anotherMap = new HashMap<String, Integer>();
31
32         // Inserting key-value pairs to anotherMap using put() method
33         anotherMap.put("SIX", 6);
34         anotherMap.put("SEVEN", 7);
35
36         // Inserting key-value pairs of map to anotherMap using putAll() method
37         anotherMap.putAll(map);
38
39         // Printing key-value pairs of anotherMap
40         entrySet = anotherMap.entrySet();
41
42         for (Entry<String, Integer> entry : entrySet) {
43             System.out.println(entry.getKey() + " : " + entry.getValue());
44         }
45
46         // Adds key-value pair 'FIVE=5' only if it is not present in map
47         map.putIfAbsent("FIVE", 5);
48
49         // Retrieving a value associated with key 'TWO'
50         int value = map.get("TWO");
51         System.out.println(value);
52 }
```

| Test | Input  | Expected  | Got   |   |
|------|--|---|---|---|
| ✓ 1  | 3<br>ONE : 1<br>ONE : 2<br>1<br>THREE : 3<br>TWO : -----<br>2<br>SIX : 6<br>THREE : ONE : 1<br>3<br>TWO : 2<br>SEVEN : 7<br>THREE : 3<br>THREE : 3 | ONE : 1<br>TWO : 2<br>THREE : 3<br>-----<br>SIX : 6<br>ONE : 1<br>TWO : 2<br>SEVEN : 7<br>THREE : 3 | ONE : 1<br>TWO : 2<br>THREE : 3<br>-----<br>SIX : 6<br>ONE : 1<br>TWO : 2<br>SEVEN : 7<br>THREE : 3 | ✓ |

|  |  |  |      |      |  |
|--|--|--|------|------|--|
|  |  |  | 2    | 2    |  |
|  |  |  | true | true |  |
|  |  |  | true | true |  |
|  |  |  | 4    | 4    |  |

Passed all tests! ✓

Finish review