

JAVA MANUAL  
AND  
MINI PROJECT

P.MADHUMITHA  
230701169  
CSE - C

# CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

**Finish review**

**Status** Finished  
**Started** Thursday, 19 September 2024, 10:40 PM  
**Completed** Thursday, 19 September 2024, 10:54 PM  
**Duration** 14 mins 15 secs

**Question 1**  
 Correct  
 Marked out of 5.00  
[Flag question](#)

**For example:**

Input	Result
123	2
456	1

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

```

1 * import java.util.Scanner;
2 * public class sample{
3 *     public static void main(String args[]){
4 *         Scanner scn = new Scanner(System.in);
5 *         int n = scn.nextInt();
6 *         if(n%2 != 0){
7 *             System.out.println('2');
8 *         }
9 *         else{
10 *             System.out.println('1');
11 *         }
12 *     }
13 * }
```

	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 Digit{
3 *     public static void main(String args[]){
4 *         Scanner scn = new Scanner(System.in);
5 *         int n = scn.nextInt();
6 *         int lastdigit = Math.abs(n%10);
7 *
8 *         System.out.println(lastdigit);
9 *     }
10 * }
```

	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: Tle 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 sample{
3 ✓     public static void main(String args[]){
4 ✓         Scanner scn = new Scanner(System.in);
5 ✓         int n = scn.nextInt();
6 ✓         int m = scn.nextInt();
7 ✓         int x = Math.abs(n%10);
8 ✓         int y= Math.abs(m%10);
9 ✓         int sum = x + y;
10    }
11   }
12 }
```

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

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, 22 September 2024, 10:55 PM  
**Completed** Sunday, 22 September 2024, 11:20 PM  
**Duration** 24 mins 31 secs

Question 1

Correct

Marked out of  
5.00

Flag question

Write a program that takes as parameter an integer n.

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

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

Note:  $n! < 10^5$

Example Input:

3

Output:

0

Example Input:

60

Output:

14

Example Input:

100

Output:

24

Example Input:

1024

Output:

253

**For example:**

Input	Result
3	0
60	14
100	24
1024	253

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

Reset answer

```

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

```

	Input	Expected	Got	
✓	3	0	0	✓
✓	60	14	14	✓

			14	
✓	100	24	24	✓
✓	1024	253	253	✓

Passed all tests! ✓

### Question 2

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.*;
2 ✓ public class prog{
3 ✓     public static void main(String[] args){
4 ✓         Scanner scn = new Scanner(System.in);
5 ✓         String name= "";
6 ✓         String value = scn.nextLine();
7 ✓         for(int i=0;i<value.length();i++){
8 ✓             switch((value.charAt(i))){
9 ✓                 case '1':
10 ✓                     name="One";
11 ✓                     break;
12 ✓                 case '2':
13 ✓                     name="Two";
14 ✓                     break;
15 ✓                 case '3':
16 ✓                     name="Three";
17 ✓                     break;
18 ✓                 case '4':
19 ✓                     name="Four";
20 ✓                     break;
21 ✓                 case '5':
22 ✓                     name="Five";
23 ✓                     break;
24 ✓                 case '6':
25 ✓                     name="Six";
26 ✓                     break;
27 ✓                 case '7':
28 ✓                     name="Seven";
29 ✓                     break;
30 ✓                 case '8':
31 ✓                     name="Eight";
32 ✓                     break;
33 ✓                 case '9':
34 ✓                     name="Nine";
35 ✓                     break;
36 ✓                 case '0':
37 ✓                     name="Zero";
38 ✓                     break;
39 ✓             }
40 ✓         }
41 ✓     }
42 ✓ }
43 }
```

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

Passed all tests! ✓

### Question 3

Correct

Marked out of 5.00

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

[Flag question](#)

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.*;
2+ public class Prog{
3+     public static void main(String[] args){
4+         Scanner scn= new Scanner(System.in);
5+         int day = scn.nextInt();
6+
7+         boolean vacation = scn.nextBoolean();
8+
9+         if(vacation){
10+             if(day == 1 || day == 7){
11+                 System.out.println("9:00");
12+             }
13+             else {
14+                 System.out.println("7:00");
15+             }
16+         }
17+         else{
18+             if( day == 1 || day == 7){
19+                 System.out.println("6:00");
20+             }
21+             else{
22+                 System.out.println("5:00");
23+             }
24+         }
25+     }
26+ }
```

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

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, 11:20 PM  
**Completed** Monday, 23 September 2024, 12:27 AM  
**Duration** 1 hour 6 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)} = {-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 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.*;
2 ✓ public class Array{
3 ✓     public static void main(String[] args){
4 ✓         Scanner scn = new Scanner(System.in);
5 ✓         int n = scn.nextInt();
6 ✓         int[] a = new int[n];
7 ✓         for(int i=0;i<n;i++){
8 ✓             a[i] = scn.nextInt();
9 ✓         }
10 ✓        int max=0;
11 ✓        for(int i=0;i<n;i++){
12 ✓            if(max<a[i]){

```

```

13     max=a[i];
14   }
15 }
16 for(int i=0;i<n;i++){
17   a[i] = a[i]-max;
18 }
19 for(int i=0;i<n;i++){
20   a[i]=a[i]*max;
21   System.out.print(a[i]+" ");
22 }
23 }
24 }
25 }
```

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

Passed all tests! ✓

## Question 2

Correct

Marked out of 5.00

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:**

<b>Input</b>	<b>Result</b>
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 Prog{
3   public static void main(String[] args){
4     Scanner scn = new Scanner(System.in);
5     int n= scn.nextInt();
6     int[] a = new int[n];
7     int i=0;
8     for( i=0;i<n;i++){
9       a[i]=scn.nextInt();
```

```

10 }
11 int r=0;
12 for( i=0;i<n;i++){
13     int d=g(a[i],i);
14     r+=d*d;
15 }
16 System.out.println(r);
17 }
18 public static int g(int num,int pos){
19     for( int i=0;i<pos;i++){
20         num/=10;
21     }
22     return num%10;
23 }
24 }

```

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

Passed all tests! ✓

### Question 3

Correct

Marked out of  
5.00

[Flag question](#)

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

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

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

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

input1 represents the number of elements in the array.

input2 represents the array of integers.

Example 1:

input1 = 16

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

Expected output = 62

Explanation:

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

Example 2:

input1 = 11

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

Expected output = -1

Explanation:

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

Example 3:

input1 = 16

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

Expected output = 174

Explanation:

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

#### For example:

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

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

```

1 import java.util.*;
2 public class Main{
3     public static int long_seq(int n,int[] seq){
4         int curr_len=0,curr_sum=0,max_len=0,max_sum=0;
5         for(int i=0;i<n;i++){
6             if(seq[i]>=0){
7                 curr_len++;
8                 curr_sum+=seq[i];
9             }
10            else{
11                if(curr_len>max_len){
12                    max_sum=curr_sum;
13                    max_len=curr_len;
14                }
15            }
16        }
17        return max_sum;
18    }
19 }

```

```

16     max_len=curr_len;
17     max_sum=curr_sum;
18   }
19   curr_sum=0;
20   curr_len=0;
21 }
22 if(curr_len>max_len){
23   max_len=curr_len;
24   max_sum=curr_sum;
25 }
26 else if(curr_len>max_len){
27   max_sum+=curr_sum;
28 }
29 if(max_len<=0){
30   return -1;
31 }
32 else
33   return max_sum;
34 }
35 public static void main(String[] args){
36   Scanner scn = new Scanner(System.in);
37   int n=scn.nextInt();
38   int[] Array=new int[n];
39   for(int i=0;i<n;i++){
40     Array[i]=scn.nextInt();
41   }
42   System.out.println(long_seq(n,Array));
43 }
44 }
```

	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! ✓

Finish review

# CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1
2
3

Show one page at a time

Finish review

<b>Status</b>	Finished
<b>Started</b>	Sunday, 29 September 2024, 2:41 PM
<b>Completed</b>	Sunday, 29 September 2024, 3:14 PM
<b>Duration</b>	32 mins 28 secs

**Question 1**

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

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

Passed all tests! ✓

**Question 2**

Correct

Marked out of  
5.00

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     String mf;
3     String os;
4     String color;
5     int cost;
6
7     public Mobile(String mf , String os, String color, int cost){
8         this.mf = mf;
9         this.os = os;
10        this.color = color;
11        this.cost = cost;
12    }
13    public void setmf(String mf){
14        this.mf = mf;
15    }
16    public void setos(String os){
17        this.os = os;
18    }
19    public void setcost( int cost){
20        this.cost = cost;
21    }
22    public String getmf(){
23        return mf;
24    }
25    public String getos(){
26        return os;
27    }
28    public String getcolor(){
29        return color;
30    }
31    public int getcost(){
32        return cost;
33    }
34    public static void main(String args[]){
35        Mobile product = new Mobile("Redmi","Andriod","Blue",34000);
36        System.out.println("manufacturer = "+ product.getmf());
37        System.out.println("operating system = "+ product.getos());
38        System.out.println("color = "+ product.getcolor());
39        System.out.println("cost = "+ product.getcost());
40    }
41 }

```

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

Passed all tests! ✓

### Question 3

Correct

Marked out of  
5.00

[Flag question](#)

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

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

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

**Input:**

2

**Output:**

**Area = 12.57**

**Circumference = 12.57**

**For example:**

Test	Input	Result
1	4	Area = 50.27 Circumference = 25.13

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

[Reset answer](#)

```

1 import java.io.*;
2 import java.util.*;
3 class Circle

```

```

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

```

	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! ✓

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	Saturday, 5 October 2024, 10:01 PM
Completed	Saturday, 5 October 2024, 10:34 PM
Duration	33 mins 25 secs

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

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

Passed all tests! ✓

Question 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 acc,double bal){  
10         this.accountNumber=acc;  
11         this.balance=bal;  
12     }  
13  
14  
15  
16  
17     // Method to deposit an amount into the account  
18     public void deposit(double amount) {  
19         // Increase the balance by the deposit amount  
20         balance+=amount;  
21     }  
22  
23     // Method to withdraw an amount from the account  
24     public void withdraw(double amount) {  
25         // Check if the balance is sufficient for the withdrawal  
26         if (balance >= amount) {  
27             // Decrease the balance by the withdrawal amount  
28             balance -= amount;  
29         } else {  
30             // Print a message if the balance is insufficient  
31             System.out.println("Insufficient balance");  
32         }  
33  
34  
35     // Method to get the current balance  
36     public double getBalance() {  
37         // Return the current balance  
38         return balance;  
39     }  
40 }  
41  
42 class SavingsAccount extends BankAccount {  
43     // Constructor to initialize account number and balance  
44     public SavingsAccount(String accountNumber, double balance) {  
45         // Call the parent class constructor  
46         super(accountNumber,balance);  
47     }  
48  
49     // Override the withdraw method from the parent class  
50     @Override  
51     public void withdraw(double amount) {  
52         // Check if the withdrawal would cause the balance to drop below $100
```

Expected	Got
<p>✓ 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</p>	<p>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</p>

Passed all tests! ✓

### Question 3

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 Admitted0. Create a subclass called CSE that extends Student class,with department attribute , Course0 method to sub class. Print the details of the Student.

College:

```
String collegeName;
```

```
public College()
```

```
public admitted0()
```

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

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

Passed all tests! ✓

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, 5:50 PM  
**Completed** Sunday, 6 October 2024, 6:19 PM  
**Duration** 29 mins 6 secs

Question 1

Correct

Marked out of  
5.00

[Flag question](#)

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

For example:

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

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

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

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

If the word to be processed is "Nice":

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

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

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

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

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

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

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

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

Expected output:

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

Example 1:

input1 = "Today is a Nice Day"

input2 = 41

output = "iNce doTday"

Example 2:

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

input2 = 39

output = "naMngo arGpes"

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

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

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

For example:

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

Answer: (penalty regime: 0 %)

```

1+ import java.util.*;
2+ public class hello{
3+   public static void main(String[] args){
4+     Scanner scn = new Scanner(System.in);
5+     String stc=scn.nextLine();
6+     String[] str=stc.split(" ");
7+     int a =scn.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+      for(int i=mid-1;i>0;i--){
15+        st+=str[c-1].charAt(i);
16+      }
17+    }else{
18+      for(int i=mid;i>0;i--){
19+        st+=str[c-1].charAt(i);
20+      }
21+    }
22+    for(int i=mid;i<str[c-1].length();i++){
23+      st+=str[c-1].charAt(i);
24+    }
25+    int len1=st[b-1].length();
26+    int m1=len1/2;
27+    st+=" ";
28+    if(len1%2==0){
29+      for(int i=m1-1;i>0;i--){
30+        st+=str[b-1].charAt(i);
31+      }

```

```

32 }
33 }
34 else
35 {
36     for(int i=m1;i>=0;i--){
37         st+=str[b-1].charAt(i);
38     }
39     for(int i =m1;i<str[b-1].length();i++){
40         st+=str[b-1].charAt(i);
41     }
42 }
43 System.out.println(st);
44 }
45 }
46 }

```

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

Passed all tests! ✓

## Question 2

Correct

Marked out of  
5.00

Flag question

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

Note:

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

Case 1:

Check whether the two alphabets are same.

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

Example 1:

input1 = ww:ii:pp:rr:oo

output = WIPRO

Explanation:

word1 is ww, both are same hence take w

word2 is ii, both are same hence take i

word3 is pp, both are same hence take p

word4 is rr, both are same hence take r

word5 is oo, both are same hence take o

Hence the output is WIPRO

Case 2:

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

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

Example 2"

input1 = zx:za:ee

output = BYE

Explanation

word1 is zx, both are not same alphabets

position value of z is 26

position value of x is 24

max – min will be 26 – 24 = 2

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

Word2 is za, both are not same alphabets

position value of z is 26

position value of a is 1

max – min will be 26 – 1 = 25

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

word3 is ee, both are same hence take e

Hence the output is BYE

**For example:**

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

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

```

1 import java.util.*;
2 public class sample{
3     public static void main(String[] args){

```

```

4 Scanner scn = new Scanner(System.in);
5 String s=scn.next();
6 String st="";
7 String stri="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 =stri.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: rponligea

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	rponligea
2	fruits are good	utsroigfeda

Answer: (penalty regime: 0 %)

```

1 import java.util.*;
2 import java.util.Set;
3 import java.util.TreeSet;
4 public class Concatenate{
5     public static void main(String[] args){
6         Scanner scn = new Scanner(System.in);
7         String inp1 = scn.nextLine();
8         String inp2 = scn.nextLine();
9
10        if(inp1.trim().isEmpty() && inp2.trim().isEmpty()){
11            System.out.println("null");
12        }
13        else{
14            String combined = inp1+inp2;
15            Set<Character> charset = new TreeSet<>((a,b) -> Character.compare(b,a));
16            for(char c : combined.toCharArray()){
17                if( c!= ' '){
18                    charset.add(c);
19                }
20            }
21            StringBuilder result = new StringBuilder();
22            for(char c : charset){
23                result.append(c);
24            }
25            System.out.println(result.toString());
26        }
}

```

27  
28 }

	Test	Input	Expected	Got	
✓	1	apple orange	rponlgea	rponlgea	✓
✓	2	fruits 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, 6:19 PM  
**Completed** Sunday, 6 October 2024, 6:53 PM  
**Duration** 33 mins 17 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 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 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 sample{
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 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.	RBI has a new Policy issued in 2023 RBI has updated new regulations in 2024. SBI rate of interest: 7.6 per annum. Karur rate of interest: 7.4 per annum.	✓

Passed all tests! ✓

**Question 2**

Correct

create an interface Playable with a method play0 that takes no arguments and returns void. Create three classes Football, Volleyball, and Basketball that implement the Playable interface and override the play0 method to play the respective sports.

Marked out of

5.00

[Flag question](#)

```
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.*;
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 sample{
33     public static void main(String[] args){
34         Scanner scn = new Scanner(System.in);
35         f f1=new f(scn.next());
36         v v1=new v(scn.next());
37         b b1=new b(scn.next());
38         f1.pl();
39         v1.pl();
40         b1.pl();
41     }
42 }
```

	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! ✓

**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);
}
```

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

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, 13 October 2024, 10:30 AM  
**Completed** Sunday, 13 October 2024, 10:58 AM  
**Duration** 28 mins 51 secs

Question 1

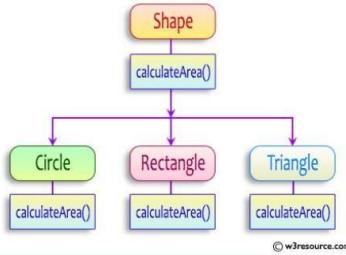
Correct

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:



```

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

System.out.printf("Area of a Triangle :%.2f%n",((0.5)*base*height)); // use this statement
sample Input :
4 // radius of the circle to calculate area PI*r*r
5 // length of the rectangle
6 // breadth of the rectangle to calculate the area of a rectangle
4 // base of the triangle
3 // height of the triangle
  
```

OUTPUT:

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

For example:

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

Answer: (penalty regime: 0 %)

```

1 import java.util.*;
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)/2;
38        System.out.printf("Area of a Triangle: %.2f\n",a);
39        return a;
40    }
41 }
  
```

```

22
23     this.h=h;
24 }
25
26     public double calculateArea(){
27         double a=b*h*0.5;
28         System.out.printf("Area of a Triangle: %.2f\n",a);
29         return a;
30     }
31 }
32
33     public class sample{
34         public static void main(String[] args){
35             Scanner scn=new Scanner(System.in);
36             double r1=scn.nextDouble();
37             c c1=new c(r1);
38             double l1=scn.nextDouble();
39             double b1=scn.nextDouble();
40             r r2=new r(l1,b1);
41             double b2=scn.nextDouble();
42             double h2=scn.nextDouble();
43             t t1=new t(b2,h2);
44         }
45     }

```

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

Passed all tests! ✓

### Question 2

Correct

Marked out of  
5.00

Flag question

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: oreapple

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	oreapple
2 Mango banana	no matches found
3 Ate Ace Girl	ateace

Answer: (penalty regime: 0 %)

```

1 import java.util.*;
2 public class sample{
3     public static void main(String[] args){
4         Scanner scn = new Scanner(System.in);
5         int n = scn.nextInt();
6         int k=0;
7         String arr[] = new String[n];
8         for(int i=0;i<n;i++){
9             arr[i] = scn.next();
10            arr[i] = arr[i].toLowerCase();
11            char ch=arr[i].charAt(0);
12            if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u'){
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
3
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
38
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
42
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
45
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
47
480
481
482
483
484
485
486
487
488
489
48
490
491
492
493
494
495
496
497
498
499
49
4
500
4
501
4
502
4
503
4
504
4
505
4
506
4
507
4
508
4
509
4
50
510
511
512
513
514
515
516
517
518
519
51
520
521
522
523
524
525
526
527
528
529
52
530
531
532
533
534
535
536
537
538
539
53
540
541
542
543
544
545
546
547
548
549
54
550
551
552
553
554
555
556
557
558
559
55
560
561
562
563
564
565
566
567
568
569
56
570
571
572
573
574
575
576
577
578
579
57
580
581
582
583
584
585
586
587
588
589
58
590
591
592
593
594
595
596
597
598
599
59
5
600
5
601
5
602
5
603
5
604
5
605
5
606
5
607
5
608
5
609
5
60
610
611
612
613
614
615
616
617
618
619
61
620
621
622
623
624
625
626
627
628
629
62
630
631
632
633
634
635
636
637
638
639
63
640
641
642
643
644
645
646
647
648
649
64
650
651
652
653
654
655
656
657
658
659
65
660
661
662
663
664
665
666
667
668
669
66
670
671
672
673
674
675
676
677
678
679
67
680
681
682
683
684
685
686
687
688
689
68
690
691
692
693
694
695
696
697
698
699
69
6
700
6
701
6
702
6
703
6
704
6
705
6
706
6
707
6
708
6
709
6
70
710
711
712
713
714
715
716
717
718
719
71
720
721
722
723
724
725
726
727
728
729
72
730
731
732
733
734
735
736
737
738
739
73
740
741
742
743
744
745
746
747
748
749
74
750
751
752
753
754
755
756
757
758
759
75
760
761
762
763
764
765
766
767
768
769
76
770
771
772
773
774
775
776
777
778
779
77
780
781
782
783
784
785
786
787
788
789
78
790
791
792
793
794
795
796
797
798
799
79
7
800
7
801
7
802
7
803
7
804
7
805
7
806
7
807
7
808
7
809
7
80
810
811
812
813
814
815
816
817
818
819
81
820
821
822
823
824
825
826
827
828
829
82
830
831
832
833
834
835
836
837
838
839
83
840
841
842
843
844
845
846
847
848
849
84
850
851
852
853
854
855
856
857
858
859
85
860
861
862
863
864
865
866
867
868
869
86
870
871
872
873
874
875
876
877
878
879
87
880
881
882
883
884
885
886
887
888
889
88
890
891
892
893
894
895
896
897
898
899
89
8
900
8
901
8
902
8
903
8
904
8
905
8
906
8
907
8
908
8
909
8
90
910
911
912
913
914
915
916
917
918
919
91
920
921
922
923
924
925
926
927
928
929
92
930
931
932
933
934
935
936
937
938
939
93
940
941
942
943
944
945
946
947
948
949
94
950
951
952
953
954
955
956
957
958
959
95
960
961
962
963
964
965
966
967
968
969
96
970
971
972
973
974
975
976
977
978
979
97
980
981
982
983
984
985
986
987
988
989
98
990
991
992
993
994
995
996
997
998
999
99
9

```

```

13     |         k=1;
14     |         System.out.print(arr[i]);
15     |
16     |     }
17     |     if(k==0){
18     |         System.out.println("no matches found");
19     |
20     |
21     }
22 }
```

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

Passed all tests! ✓

### Question 3

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 display0 {
    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     |         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
15     // You can create new methods here
16     public void showDetails() {
17         System.out.println("This is a subclass of FinalExample.");
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 }
30 }
```

	Test	Expected	Got	
✓	1	The maximum speed is: 120 km/h	The maximum speed is: 120 km/h	✓

This is a subclass of FinalExample. This is a subclass of FinalExample.

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, 13 October 2024, 2:08 PM  
**Completed** Sunday, 13 October 2024, 2:51 PM  
**Duration** 42 mins 16 secs

**Question 1**  
 Correct  
 Marked out of 5.00  
[Flag question](#)

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

**Sample input and Output:**

```
82 is even.  
Error: 37 is odd.
```

Fill the preloaded answer to get the expected output.

**For example:**

Result
82 is even. Error: 37 is odd.

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

[Reset answer](#)

```

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

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

Passed all tests! ✓

**Question 2**  
 Correct  
 Marked out of 5.00  
[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.ArithmetricException: / by zero**  
**I am always executed****Input:**3  
10 20 30**Output:****java.lang.ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3**  
**I am always executed****For example:**

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

Answer: (penalty regime: 0 %)

```

1+ import java.util.*;
2
3 class prog{
4+   public static void main(String[] args){
5     Scanner scn = new Scanner(System.in);
6     int size = scn.nextInt();
7     int[] arr = new int[size];
8+   for(int i=0 ;i<size;i++){
9       arr[i] = scn.nextInt();
10    }
11+ try{
12     int result = arr[0] / arr[1];
13
14     System.out.println(arr[3]);
15+ }catch(ArithmaticException e){
16     System.out.println("java.lang.ArithmaticException: "+ e.getMessage());
17+ }catch(ArrayIndexOutOfBoundsException e){
18     System.out.println("java.lang.ArrayIndexOutOfBoundsException: "+e.getMessage());
19+ }
20+ finally{
21     System.out.println("I am always executed");
22+ }
23
24  }
25 }
```

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

Passed all tests! ✓

### Question 3

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
7
8+   /* Define try-catch block to save user input in the array "name"
9 If there is an exception then catch the exception otherwise print
10 the total sum of the array. */
11
12  {
13     int length = sc.nextInt();
14     int[] name = new int[length];
15     int sum =0 ;
16+   for (int i=0;i<length;i++){
17         name[i] = sc.nextInt();
18         sum += name[i];
19     }
20     System.out.println(sum);
21 }
```

```
21 }  
22 }  
23 catch(InputMismatchException e )  
24 {  
25     System.out.println("You entered bad data.");  
26  
27 }finally{  
28     sc.close();  
29 }  
30 }  
31 }  
32 }
```

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

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, 3 November 2024, 7:34 PM  
**Completed** Sunday, 3 November 2024, 7:56 PM  
**Duration** 21 mins 59 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.*;
2 * class prog{
3 *     public static void main(String[] args){
4 *         ArrayList<Integer> array = new ArrayList<Integer>();
5 *         Scanner scn = new Scanner(System.in);
6 *         int n = scn.nextInt();
7 *         for(int i=0 ;i<n;i++){
8 *             array.add(scn.nextInt());
9 *         }
10 *        System.out.println("ArrayList: "+array);
11 *        System.out.print("First : "+array.get(0)+" , ");
12 *        System.out.println("Last : "+array.get(n-1));
13 *    }
14 * }
```

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

Passed all tests! ✓

**Question 2**  
 Correct  
 Marked out of 1.00  
[Flag question](#)

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

```

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

The above methods are used for the below Java program.

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

[Reset answer](#)

```

1 * import java.util.ArrayList;
2 * import java.util.Scanner;
3 *
4 * public class Prog{
5 *
6 *     public static void main(String[] args)
7 *     {
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;i++){
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 // Check whether 200 is in the list or not
28 System.out.println(list.contains(200)); //output : false
29 // Print ArrayList size
30 System.out.println("Size of ArrayList = "+ list.size());
31 //Inserting 500 at index 1
32 list.add(1,500); // code here
33 //Removing an element from position 3
34 list.remove(3); // code here
35 System.out.print("ArrayList: " + list);
36 }
37 }

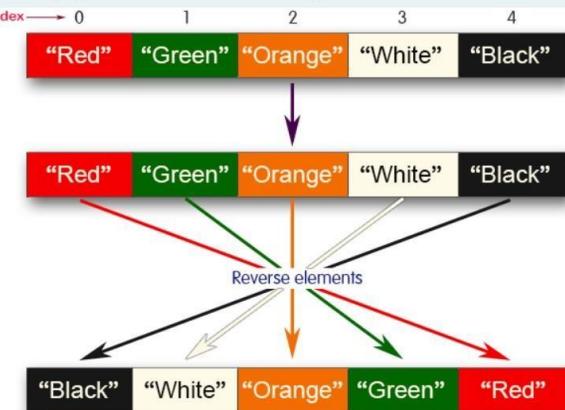
```

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

Passed all tests! ✓

Question 3  
Correct  
Marked out of 1.00  
Flag question

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



Sample input and Output:

```

Red
Green
Orange
White
Black
Sample output
List before reversing :
[Red, Green, Orange, White, Black]
List after reversing :
[Black, White, Orange, Green, Red]

```

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

```

1 ✓ import java.util.*;
2 ✓ class prog{
3 ✓     public static void main(String[] args){
4 ✓         Scanner scn = new Scanner(System.in);
5 ✓         ArrayList<String> list = new ArrayList<String>();
6 ✓         int n = scn.nextInt();
7
8 ✓         for(int i = 0 ;i<n+1 ;i++){
9 ✓             list.add(scn.nextLine());
10
11             list.remove(0);
12             System.out.println("List before reversing :");
13             System.out.println(list);
14             Collections.reverse(list);
15             System.out.println("List after reversing :");
16             System.out.println(list);
17         }
18     }

```

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

Passed all tests! ✓

Finish review

# CS23333-Object Oriented Programming Using Java-2023

Quiz navigation

1 2 3

Show one page at a time

Finish review

**Status** Finished  
**Started** Monday, 11 November 2024, 1:24 PM  
**Completed** Monday, 11 November 2024, 1:54 PM  
**Duration** 29 mins 29 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+ public 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        int skey=sc.nextInt();
15
16        // Show which numbers between 1 and 10 are in the set
17
18+        if(numbers.contains(skey)){
19+            System.out.println( skey + " was found in the set.");
20+        } else {
21+            System.out.println(skey + " was not found in the set.");
22+        }
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! ✓

**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.*;
2 * import java.util.HashSet;
3 * import java.util.Set;
4 * public class prog{
5 *     public static void main(String[] args){
6 *         Scanner scn = new Scanner(System.in);
7 *         int n1 = scn.nextInt();
8 *         scn.nextLine();
9 *         Set<String> set1 = new HashSet<>();
10 *        for(int i =0; i<n1;i++){
11 *            set1.add(scn.nextLine());
12 *        }
13 *        int n2 = scn.nextInt();
14 *        scn.nextLine();
15 *        Set<String> set2 = new HashSet<>();
16 *        for(int i = 0;i<n2;i++){
17 *            set2.add(scn.nextLine());
18 *        }
19 *        set1.retainAll(set2);
20 *        for(String item : set1){
21 *            System.out.println(item);
22 *        }
23 *    }
24 * }
```

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

Passed all tests! ✓

**Question 3**

Correct

Marked out of  
1.00

Flag question

**Java HashMap Methods****containsKey()** Indicate if an entry with the specified key exists in the map**containsValue()** Indicate if an entry with the specified value exists in the map**putIfAbsent()** Write an entry into the map but only if an entry with the same key does not already exist

`remove()` Remove an entry from the map  
`replace()` Write to an entry in the map only if it exists  
`size()` Return the number of entries in the map

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

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

[Reset answer](#)

```
1+ import java.util.HashMap;
2 import java.util.Map.Entry;
3 import java.util.Set;
4 import java.util.Scanner;
5 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        }
54    }
55}
```

	Test	Input	Expected	Got	
✓	1	3 ONE TWO 1 THREE TWO ----- 2 SIX THREE ONE : 1 3 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	ONE : 1 TWO : 2 THREE : 3 ----- SIX : 6 ONE : 1 TWO : 2 SEVEN : 7 THREE : 3 2 true true 4	✓

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** Monday, 11 November 2024, 1:54 PM  
**Completed** Monday, 11 November 2024, 2:37 PM  
**Duration** 43 mins 14 secs

Question 1

Correct

Marked out of 5.00

[Flag question](#)

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

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

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

If case\_option = 1, reversal of words with retaining position's case i.e., if the original sentence is "Wipro TechNologies BangaLore", the new reversed sentence should be "Orpiw 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 ReverseWordsWithCaseControl {
4+   public static String reverseWordsWithCase(String sentence, int case_option) {
5+     String[] words = sentence.split(" ");
6+     StringBuilder result = new StringBuilder();
7
8+     for (String word : words) {
9+       String reversedWord = reverseWord(word);
10+      if (case_option == 1) {
11+        reversedWord = reverseCaseWithOriginalPosition(reversedWord, word);
12+      }
13+      result.append(reversedWord).append(" ");
14+    }
15+    return result.toString().trim();
16+
17+  private static String reverseWord(String word) {
18+    StringBuilder reversed = new StringBuilder(word);
19+    return reversed.reverse().toString();
20+  }
21
22+  private static String reverseCaseWithOriginalPosition(String reversedWord, String originalWord) {
23+    StringBuilder result = new StringBuilder(reversedWord);
24+    for (int i = 0; i < originalWord.length(); i++) {
25+      char originalChar = originalWord.charAt(i);
26+      char reversedChar = reversedWord.charAt(i);
27+      if (Character.isUpperCase(originalChar)) {
28+        result.setCharAt(i, Character.toUpperCase(reversedChar));
29+      } else if (Character.isLowerCase(originalChar)) {
30+        result.setCharAt(i, Character.toLowerCase(reversedChar));
31+      }
32+    }
33+    return result.toString();
34+  }
35
36+  public static void printResult(String input, int case_option) {
37+    String result = reverseWordsWithCase(input, case_option);
38+    System.out.println(result);
39+  }

```

```

39
40
41
42
43
44
45
46
47
48
}

```

	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 Seigolonhct Erolagnab	Orpiw Seigolonhct Erolagnab	✓
✓	Wipro Technologies, Bangalore 1	Orpiw ,seigolonhceT Erolagnab	Orpiw ,seigolonhceT Erolagnab	✓

Passed all tests! ✓

## 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 import java.util.Set;
3
4 public class CommonCharSum {
5     public static int calculatesingleDigitSum(char[] input1, char[] input2) {
6         Set<Character> set1 = new HashSet<>();
7         Set<Character> set2 = new HashSet<>();
8         for (char ch : input1) {
9             set1.add(ch);
10        }
11        for (char ch : input2) {
12            set2.add(ch);
13        }
14        set1.retainAll(set2);
15        int sum1 = 0;
16        for (char ch : set1) {
17            sum1 += (int) ch;
18        }
19
20        return getsingleDigitSum(sum1);
21    }
22
23    public static int getsingleDigitSum(int sum) {
24        while (sum >= 10) {
25            int tempSum = 0;
26            while (sum > 0) {
27                tempSum += sum % 10;
28                sum /= 10;
29            }
30            sum = tempSum;
31        }
32        return sum;
33    }
}

```





Finish review

# MINI PROJECT

## ONLINE MOVIE TICKET BOOKING SYSTEM

### CODE :

```
package movieticketbooking;  
  
import javax.swing.*; import  
java.awt.*; import  
java.util.List; import  
java.util.Map;  
  
  
public class MovieTicketBookingSystem { private static  
DatabaseOperation db = new DatabaseOperation(); private static int  
loggedInUserID = -1; // User session management public static void  
main(String[] args) { showMainMenu();  
}  
  
// Main Menu private static void
```

```
showMainMenu() {  
    JFrame mainWindow = new JFrame("Movie Ticket Booking System");  
    mainWindow.setSize(400, 300);  
    mainWindow.setLayout(new GridLayout(3, 1));  
  
    JButton adminButton = new JButton("Admin Login");  
    JButton userButton = new JButton("User Login");  
    JButton exitButton = new JButton("Exit");  
  
    mainWindow.add(adminButton);  
    mainWindow.add(userButton);    mainWindow.add(exitButton);  
  
    adminButton.addActionListener(e -> {  
        mainWindow.dispose();  
        showAdminLogin();  
    });  
    userButton.addActionListener(e -> {      mainWindow.dispose();  
        showUserLogin();  
    });  
    exitButton.addActionListener(e -> System.exit(0));  
    mainWindow.setVisible(true);  
}  
  
// ADMIN FUNCTIONS  private static void showAdminLogin()  
{    JFrame adminLogin = new  
JFrame("Admin Login");    adminLogin.setSize(300,  
200);    adminLogin.setLayout(new GridLayout(3,  
2));  
  
JLabel usernameLabel = new JLabel("Admin Name:");
```

```

JTextField usernameField = new JTextField();
JLabel passwordLabel = new JLabel("Password:");
JPasswordField passwordField = new JPasswordField();

JButton loginButton = new JButton("Login");
JButton backButton = new JButton("Back");

adminLogin.add(usernameLabel);
adminLogin.add(usernameField);
adminLogin.add(passwordLabel);      adminLogin.add(passwordField);
adminLogin.add(loginButton);      adminLogin.add(backButton);

loginButton.addActionListener(e -> {
    String adminName = usernameField.getText();
    String password = new String(passwordField.getPassword());

    String sql = "SELECT Admin_name, Password FROM admin WHERE Admin_name = ?";
    Map<String, Object> admin = db.validatePass(sql, adminName);

    if (admin != null && password.equals(admin.get("Password"))) {
        JOptionPane.showMessageDialog(adminLogin, "Login successful!");
        adminLogin.dispose();
        showAdminDashboard();
    } else {
        JOptionPane.showMessageDialog(adminLogin, "Invalid credentials!");
    }
});

backButton.addActionListener(e -> {      adminLogin.dispose();
showMainMenu();
});

```

```
    adminLogin.setVisible(true);

}

private static void showAdminDashboard() {
    JFrame adminDashboard = new JFrame("Admin Dashboard");
    adminDashboard.setSize(400, 400);      adminDashboard.setLayout(new
    GridLayout(5, 1));

    JButton addMovieButton = new JButton("Add Movies");
    JButton addTheaterButton = new JButton("Add Theaters");
    JButton addShowtimeButton = new JButton("Add Showtimes");
    JButton viewMoviesButton = new JButton("View Movies and Showtimes");      JButton
    logoutButton = new JButton("Logout");      adminDashboard.add(addMovieButton);
    adminDashboard.add(addTheaterButton);      adminDashboard.add(addShowtimeButton);
    adminDashboard.add(viewMoviesButton);      adminDashboard.add(logoutButton);

    addMovieButton.addActionListener(e -> {
        adminDashboard.dispose();
        addMovie();
    });

    addTheaterButton.addActionListener(e -> {
        adminDashboard.dispose();
        addTheater();
    });

    addShowtimeButton.addActionListener(e -> {      adminDashboard.dispose();
        addShowtime();
    });
}
```

```
});

viewMoviesButton.addActionListener(e -> {
    adminDashboard.dispose();
    viewMoviesAndShowtimes();
});

logoutButton.addActionListener(e -> {
    adminDashboard.dispose();
    showMainMenu();
});

adminDashboard.setVisible(true);
}

private static void addMovie() {
    JFrame addMovieFrame = new JFrame("Add Movie");
    addMovieFrame.setSize(300, 400);
    addMovieFrame.setLayout(new GridLayout(6, 2));
    JLabel movieidLabel = new JLabel("Movie ID:");
    JTextField movieidField = new JTextField();
    JLabel titleLabel = new JLabel("Movie Title:");
    JTextField titleField = new JTextField();
    JLabel genreLabel = new JLabel("Genre:");
    JTextField genreField = new JTextField();
    JLabel durationLabel = new JLabel("Duration (mins):");
    JTextField durationField = new JTextField();
    JLabel synopsisLabel = new JLabel("Synopsis:");
    JTextField synopsisField = new JTextField();
    JLabel ratingLabel = new JLabel("Rating (0-10):");
    JTextField ratingField = new JTextField();
```

```
 JButton submitButton = new JButton("Submit");      JButton
cancelButton = new JButton("Cancel");
addMovieFrame.add(movieidLabel);
addMovieFrame.add(movieidField);
addMovieFrame.add(titleLabel);
addMovieFrame.add(titleField);
addMovieFrame.add(genreLabel);
addMovieFrame.add(genreField);
addMovieFrame.add(durationLabel);
addMovieFrame.add(durationField);
addMovieFrame.add(synopsisLabel);
addMovieFrame.add(synopsisField);
addMovieFrame.add(ratingLabel);
addMovieFrame.add(ratingField);
addMovieFrame.add(submitButton);
addMovieFrame.add(cancelButton);
```

```
submitButton.addActionListener(e -> {
int movie_id;
String title = titleField.getText();      String
genre = genreField.getText();
int duration;
double rating;
try {
movie_id = Integer.parseInt(movieidField.getText());      duration =
Integer.parseInt(durationField.getText());      rating =
Double.parseDouble(ratingField.getText());
} catch (NumberFormatException ex) {
JOptionPane.showMessageDialog(addMovieFrame, "Invalid duration or rating.
Please enter numbers.");
}
```

```

return;
}

String synopsis = synopsisField.getText();

String sql = "INSERT INTO movies (movie_ID, title, genre, Duration, Synopsis, rating)
VALUES (?,?,?,?,?,?)";

Object[] values = {movie_id,title,genre,duration,synopsis,rating};

int rowsAffected = db.executeUpdate(sql, values);

if (rowsAffected > 0) {
    JOptionPane.showMessageDialog(addMovieFrame, "Movie added successfully!");
} else {
    JOptionPane.showMessageDialog(addMovieFrame, "Failed to add the movie.");
}

addMovieFrame.dispose();
showAdminDashboard();
});

cancelButton.addActionListener(e -> {
addMovieFrame.dispose();
showAdminDashboard();
});

addMovieFrame.setVisible(true);
}

private static void addTheater() {
JFrame addTheaterFrame = new JFrame("Add Theater");
addTheaterFrame.setSize(300, 400);      addTheaterFrame.setLayout(new
GridLayout(5, 2));
}

```

```
JLabel idLabel = new JLabel("Theater ID:");
JTextField idField = new JTextField();
JLabel nameLabel = new JLabel("Theater Name:");
JTextField nameField = new JTextField();
JLabel capacityLabel = new JLabel("Seating Capacity:");
JTextField capacityField = new JTextField();
JLabel locationLabel = new JLabel("Location:");
JTextField locationField = new JTextField();
JLabel screenLabel = new JLabel("Screen ID:");
JTextField screenField = new JTextField();
JLabel foodLabel = new JLabel("Food Add On:");
JTextField foodField = new JTextField();

JButton submitButton = new JButton("Submit");
JButton cancelButton = new JButton("Cancel");

addTheaterFrame.add(idLabel);
addTheaterFrame.add(idField);      addTheaterFrame.add(nameLabel);
addTheaterFrame.add(nameField);
addTheaterFrame.add(capacityLabel);
addTheaterFrame.add(capacityField);
addTheaterFrame.add(locationLabel);
addTheaterFrame.add(locationField);
addTheaterFrame.add(screenLabel);
addTheaterFrame.add(screenField);
addTheaterFrame.add(foodLabel);
addTheaterFrame.add(foodField);
addTheaterFrame.add(submitButton);
addTheaterFrame.add(cancelButton);
```

```

submitButton.addActionListener(e -> {
    String name = nameField.getText();
    int capacity;
    try {
        capacity = Integer.parseInt(capacityField.getText());
    } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(addTheaterFrame, "Invalid capacity. Please enter a number.");
    }
    String location = locationField.getText();
    String foodAddon=foodField.getText();      int theatreID,screenID;
    try {
        theatreID = Integer.parseInt(screenField.getText());
        screenID = Integer.parseInt(screenField.getText());
    } catch (NumberFormatException ex) {
        JOptionPane.showMessageDialog(addTheaterFrame, "Invalid screen ID. Please enter a number.");
    }
}

String sql = "INSERT INTO theatre (theatre_id,theatre_name,seating_capacity,location,screen_id,food_add_on) VALUES (?,?,?,?,?,?)";
Object[] values = {theatreID, name, capacity, location, screenID,foodAddon};
int rowsAffected = db.executeUpdate(sql, values);

if (rowsAffected > 0) {
    JOptionPane.showMessageDialog(addTheaterFrame, "Theater added successfully!");
} else {
    JOptionPane.showMessageDialog(addTheaterFrame, "Failed to add the theater.");
}

```

```
        }

        addTheaterFrame.dispose();

showAdminDashboard();

});

cancelButton.addActionListener(e -> {

addTheaterFrame.dispose();

showAdminDashboard();

});

addTheaterFrame.setVisible(true);

}

private static void addShowtime() {

JFrame addShowtimeFrame = new JFrame("Add Showtime");

addShowtimeFrame.setSize(300, 400);      addShowtimeFrame.setLayout(new

GridLayout(6, 2));



JLabel showtimeIDLabel = new JLabel("Showtime ID:");

JTextField showtimeIDField = new JTextField();

JLabel movieIDLabel = new JLabel("Movie ID:");

JTextField movieIDField = new JTextField();

JLabel theaterIDLabel = new JLabel("Theater ID:");

JTextField theaterIDField = new JTextField();

JLabel screenIDLabel = new JLabel("Screen ID:");

JTextField screenIDField = new JTextField();

JLabel dayLabel = new JLabel("Day:");

JTextField dayField = new JTextField();

JLabel dateLabel = new JLabel("Date (YYYY-MM-DD):");

JTextField dateField = new JTextField();
```

```

 JButton submitButton = new JButton("Submit");
 JButton cancelButton = new JButton("Cancel");

 addShowtimeFrame.add(showtimeIDLabel);
 addShowtimeFrame.add(showtimeIDField);
 addShowtimeFrame.add(movieIDLabel);
 addShowtimeFrame.add(movieIDField);
 addShowtimeFrame.add(theaterIDLabel);
 addShowtimeFrame.add(theaterIDField);
 addShowtimeFrame.add(screenIDLabel);
 addShowtimeFrame.add(screenIDField);
 addShowtimeFrame.add(dayLabel);      addShowtimeFrame.add(dayField);
 addShowtimeFrame.add(dateLabel);
 addShowtimeFrame.add(dateField);
 addShowtimeFrame.add(submitButton);
 addShowtimeFrame.add(cancelButton);
 submitButton.addActionListener(e -> {
     int showtimeID, movieID,
     theaterID, screenID;
     String day = dayField.getText();
     String date = dateField.getText();
     try {
         showtimeID = Integer.parseInt(showtimeIDField.getText());
         movieID = Integer.parseInt(movieIDField.getText());          theaterID
         = Integer.parseInt(theaterIDField.getText());           screenID =
         Integer.parseInt(screenIDField.getText());
     } catch (NumberFormatException ex) {
         JOptionPane.showMessageDialog(addShowtimeFrame, "Invalid Movie ID, Theater
         ID, or Screen ID.");
     }
     return;
}

```

```

    }

    String sql = "INSERT INTO showtimes (showtime_id, movie_ID, theatre_id, screen_id,
day, date) VALUES (?,?,?,?,?,?)";

    Object[] values = {showtimeID,movieID,theaterID,screenID,day,date};      int
rowsAffected = db.executeUpdate(sql, values);

    if (rowsAffected > 0) {
        JOptionPane.showMessageDialog(addShowtimeFrame, "Showtime added
successfully!");
    } else {
        JOptionPane.showMessageDialog(addShowtimeFrame, "Failed to add the
showtime.");
    }
    addShowtimeFrame.dispose();
    showAdminDashboard();
}

cancelButton.addActionListener(e -> {
    addShowtimeFrame.dispose();
    showAdminDashboard();
});

addShowtimeFrame.setVisible(true);
}

private static void viewMoviesAndShowtimes() {
    JFrame viewMoviesFrame = new JFrame("Movies and Showtimes");
    viewMoviesFrame.setSize(600,           600);
    viewMoviesFrame.setLayout(new GridLayout(0, 1));
}

```

```

List<Map<String, Object>> movies = db.getRecords("SELECT * FROM movies");      for

(Map<String, Object> movie : movies) {

    String movieDetails = "Movie ID: " + movie.get("movie_ID") +
        ", Title: " + movie.get("title") +
        ", Genre: " + movie.get("genre") +
        ", Duration: " + movie.get("Duration") +
        ", Rating: " + movie.get("rating");      JLabel

    movieLabel = new JLabel(movieDetails);
    viewMoviesFrame.add(movieLabel);

    int movieID = (int) movie.get("movie_ID");

    List<Map<String, Object>> showtimes = db.getRecords("SELECT * FROM showtimes
WHERE movie_ID = " + movieID);

    for (Map<String, Object> showtime : showtimes) {

        String showtimeDetails = " Showtime ID: " + showtime.get("showtime_id") +
            ", Theater ID: " + showtime.get("theatre_id") +
            ", Screen ID: " + showtime.get("screen_id") +
            ", Day: " + showtime.get("day") +
            ", Date: " + showtime.get("date");

        JLabel showtimeLabel = new JLabel(showtimeDetails);
        viewMoviesFrame.add(showtimeLabel);
    }
}

JButton backButton = new JButton("Back");
backButton.addActionListener(e      ->      {
    viewMoviesFrame.dispose();
    showAdminDashboard();
});

viewMoviesFrame.add(backButton);      viewMoviesFrame.setVisible(true);

```

```
}

// USER FUNCTIONS  private static void showUserLogin()

{    JFrame userLogin = new
JFrame("User Login");    userLogin.setSize(300,
200);    userLogin.setLayout(new GridLayout(3,
2));

JLabel usernameLabel = new JLabel("Username:");
JTextField usernameField = new JTextField();
JLabel passwordLabel = new JLabel("Password:");
JPasswordField passwordField = new JPasswordField();

JButton loginButton = new JButton("Login");
JButton backButton = new JButton("Back");

userLogin.add(usernameLabel);
userLogin.add(usernameField);
userLogin.add(passwordLabel);    userLogin.add(passwordField);
userLogin.add(loginButton);    userLogin.add(backButton);

loginButton.addActionListener(e -> {
    String username = usernameField.getText();
    String password = new String(passwordField.getPassword());

    String sql = "SELECT UserID, Password FROM users WHERE Username = ?";
    Map<String, Object> user = db.validatePass(sql, username);

    if (user != null && password.equals(user.get("Password"))) {
loggedInUserID = (int) user.get("UserID");
 JOptionPane.showMessageDialog(userLogin, "Login successful!");
    }
})
```

```
        userLogin.dispose();

    showUserDashboard();

} else {

    JOptionPane.showMessageDialog(userLogin, "Invalid credentials!");

}

});

backButton.addActionListener(e -> {

    userLogin.dispose();

showMainMenu();

});

userLogin.setVisible(true);

}

private static void showUserDashboard() {

JFrame userDashboard = new JFrame("User Dashboard");

userDashboard.setSize(400, 400);      userDashboard.setLayout(new

GridLayout(4, 1));



JButton viewShowtimesButton = new JButton("View Showtimes");

JButton bookTicketButton = new JButton("Book Ticket");

JButton viewBookingsButton = new JButton("View Bookings");

JButton logoutButton = new JButton("Logout");



userDashboard.add(viewShowtimesButton);

userDashboard.add(bookTicketButton);

userDashboard.add(viewBookingsButton);      userDashboard.add(logoutButton);




viewShowtimesButton.addActionListener(e -> {      userDashboard.dispose();

viewUserShowtimes();
```

```

});
```

```

bookTicketButton.addActionListener(e -> {
    userDashboard.dispose();
    bookTicket();
});
```

```

viewBookingsButton.addActionListener(e -> {           userDashboard.dispose();
    viewUserBookings();
});
```

```

logoutButton.addActionListener(e -> {
    loggedInUserID = -1;           userDashboard.dispose();
    showMainMenu();
});
```

```

    userDashboard.setVisible(true);
}
```

```

private static void viewUserShowtimes() {
    JFrame viewShowtimesFrame = new JFrame("Available Showtimes");
    viewShowtimesFrame.setSize(600,                               600);
    viewShowtimesFrame.setLayout(new GridLayout(0, 1));

    List<Map<String, Object>> showtimes = db.getRecords("SELECT * FROM showtimes");
    for (Map<String, Object> showtime : showtimes) {
        String showtimeDetails = "Showtime ID: " + showtime.get("showtime_id") +
            ", Movie ID: " + showtime.get("movie_ID") +
            ", Theater ID: " + showtime.get("theatre_id") +
            ", Screen ID: " + showtime.get("screen_id") +
            ", Day: " + showtime.get("day") +
    }
}
```

```

    ", Date: " + showtime.get("date");

JLabel showtimeLabel = new JLabel(showtimeDetails);

viewShowtimesFrame.add(showtimeLabel);

}

JButton backButton = new JButton("Back");

backButton.addActionListener(e -> {
    viewShowtimesFrame.dispose();
    showUserDashboard();
});

viewShowtimesFrame.add(backButton); viewShowtimesFrame.setVisible(true);

}

private static void bookTicket() {

    JFrame bookTicketFrame = new JFrame("Book Ticket");

bookTicketFrame.setSize(300, 300); bookTicketFrame.setLayout(new
GridLayout(4, 2));

JLabel showtimeIDLabel = new JLabel("Showtime ID:");

JTextField showtimeIDField = new JTextField();

JLabel seatsLabel = new JLabel("Seats (e.g., A1,A2):");

JTextField seatsField = new JTextField();

JButton bookButton = new JButton("Book");

JButton cancelButton = new JButton("Cancel");



bookTicketFrame.add(showtimeIDLabel);

bookTicketFrame.add(showtimeIDField);

bookTicketFrame.add(seatsLabel); bookTicketFrame.add(seatsField);

bookTicketFrame.add(bookButton);

bookTicketFrame.add(cancelButton);

bookButton.addActionListener(e -> {

    int showtimeID;

    String selectedSeats = seatsField.getText();
}

```

```

try {
    showtimeID = Integer.parseInt(showtimeIDField.getText());
} catch (NumberFormatException ex) {
    JOptionPane.showMessageDialog(bookTicketFrame, "Invalid Showtime ID.");
    return;
}

String sql = "INSERT INTO tickets (UserID, showtime_id,
selected_seats, payment_status, availability_status) VALUES (?, ?, ?, 'Paid',
'Confirmed')";
Object[] values = {loggedInUserID, showtimeID,
selectedSeats};
int rowsAffected = db.executeUpdate(sql, values);

if (rowsAffected > 0) {
    JOptionPane.showMessageDialog(bookTicketFrame, "Ticket booked successfully!");
} else {
    JOptionPane.showMessageDialog(bookTicketFrame, "Failed to book ticket.");
}
bookTicketFrame.dispose();

showUserDashboard();
});

cancelButton.addActionListener(e -> {
bookTicketFrame.dispose();
showUserDashboard();
});
bookTicketFrame.setVisible(true);
}

private static void viewUserBookings() {
JFrame viewBookingsFrame = new JFrame("My Bookings");
viewBookingsFrame.setSize(600, 600);    viewBookingsFrame.setLayout(new
GridLayout(0, 1));
List<Map<String,
Object>> bookings = db.getRecords("SELECT * FROM tickets WHERE UserID = "

```

```

+ loggedInUserID);      for (Map<String, Object> booking : bookings) {

    String bookingDetails = " Showtime ID: " + booking.get("showtime_id") +
        ", Seats: " + booking.get("selected_seats") +
        ", Status: " + booking.get("availability_status");      JLabel
    bookingLabel = new JLabel(bookingDetails);
    viewBookingsFrame.add(bookingLabel);

}

JButton backButton = new JButton("Back");
backButton.addActionListener(e      ->      {
    viewBookingsFrame.dispose();
    showUserDashboard();
});

viewBookingsFrame.add(backButton);      viewBookingsFrame.setVisible(true);
}

}

package movieticketbooking;
import java.sql.*; import
java.util.*; import javax.swing.*;

public class DatabaseOperation {  static final String DB_URL =
"jdbc:mysql://localhost/moviedb";  static final String USER =
"root";  static final String PASS = "GsJm$2408";

public Connection connectToDatabase() {      Connection
conn = null;

try {
    conn = DriverManager.getConnection(DB_URL, USER, PASS);
} catch (SQLException e) {

```

```

        JOptionPane.showMessageDialog(null, "Database connection failed: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);

    }

    return conn;
}

public int executeUpdate(String sql, Object[] values)
{
    int rowsAffected = 0;      try (Connection conn =
connectToDatabase();

    PreparedStatement ps = conn.prepareStatement(sql)) {
        for (int i = 0; i < values.length; i++) {
            ps.setObject(i + 1, values[i]);
        }
        rowsAffected = ps.executeUpdate();
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, "SQL Update Failed: " + e.getMessage(),
"Error", JOptionPane.ERROR_MESSAGE);
    }
    return rowsAffected;
}

public List<Map<String, Object>> getRecords(String sql) {
    List<Map<String, Object>> records = new ArrayList<>();      try
( Connection conn = connectToDatabase();
PreparedStatement pstmt = conn.prepareStatement(sql);

    ResultSet rs = pstmt.executeQuery()) {
        ResultSetMetaData rsmd = rs.getMetaData();      int
        columnCount = rsmd.getColumnCount();
        while (rs.next()) {
            Map<String, Object> row = new HashMap<>();
            for (int i = 1; i <= columnCount; i++) {

```

```

        row.put(rsmd.getColumnName(i), rs.getObject(i));

    }

    records.add(row);

}

} catch (SQLException e) {

    JOptionPane.showMessageDialog(null, "SQL Query Failed: " + e.getMessage(), "Error",
JOptionPane.ERROR_MESSAGE);

}

return records;

}

```

```

public int getSeatingCapacity(String sql, int parameter)

{   int seatingCapacity = 0;      try (Connection conn =
connectToDatabase();

PreparedStatement ps = conn.prepareStatement(sql)) {

    ps.setInt(1, parameter);

    ResultSet rs = ps.executeQuery();

    if (rs.next()) {

        seatingCapacity = rs.getInt("SeatingCapacity");

    }

} catch (SQLException e) {

    JOptionPane.showMessageDialog(null, "Error fetching seating capacity: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);

}

return seatingCapacity;

}

```

```

public ArrayList<Integer> getBookedSeats(int showtimeID) {

    String sql = "SELECT SelectedSeats FROM bookings WHERE ShowtimeID = ?";

ArrayList<Integer> bookedSeats = new ArrayList<>();      try (Connection conn =
connectToDatabase());

```

```

PreparedStatement ps = conn.prepareStatement(sql))

{      ps.setInt(1, showtimeID);      ResultSet rs =
ps.executeQuery();

while (rs.next()) {
    bookedSeats.add(rs.getInt("SelectedSeats"));

}

} catch (SQLException e) {

    JOptionPane.showMessageDialog(null, "Error fetching booked seats: " +
e.getMessage(), "Error", JOptionPane.ERROR_MESSAGE);

}

return bookedSeats;
}

public int removeBooking(int bookingID) {

String sql = "DELETE FROM bookings WHERE BookingID = ?";

int rowsAffected = 0;      try (Connection
conn = connectToDatabase();

PreparedStatement ps = conn.prepareStatement(sql)) {

    ps.setInt(1, bookingID);

rowsAffected = ps.executeUpdate();

} catch (SQLException e) {

    JOptionPane.showMessageDialog(null, "Error removing booking: " + e.getMessage(),
"Error", JOptionPane.ERROR_MESSAGE);

}

return rowsAffected;
}

public Map<String, Object> validatePass(String sql, Object... params)

{      Map<String, Object> result = new HashMap<>();      try
(Connection conn = connectToDatabase();

PreparedStatement ps = conn.prepareStatement(sql)) {

```

```

        for (int i = 0; i < params.length; i++) {
            ps.setObject(i + 1, params[i]);
        }

        ResultSet rs = ps.executeQuery();
        if (rs.next()) {
            ResultSetMetaData metaData = rs.getMetaData();
            int columnCount = metaData.getColumnCount();
            for (int i = 1; i <= columnCount; i++) {
                result.put(metaData.getColumnName(i), rs.getObject(i));
            }
        }
    } catch (SQLException e) {
        e.printStackTrace();
    }
    return result.isEmpty() ? null : result; // Return null if no record found
}

public int fetchUserID(String sql, String username) {
    int userID = 0;
    try (Connection conn = connectToDatabase()) {
        PreparedStatement ps = conn.prepareStatement(sql)) {
            ps.setString(1, username);
            ResultSet rs = ps.executeQuery();
            if (rs.next()) {
                userID = rs.getInt("UserID");
            }
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, "Error fetching UserID: " + e.getMessage(),
                    "Error", JOptionPane.ERROR_MESSAGE);
        }
    }
    return userID;
}

```

}

## **SNAPSHOTS:**



Movie Ticket Booking System



**Admin Login**

**User Login**

**Exit**



Admin Login

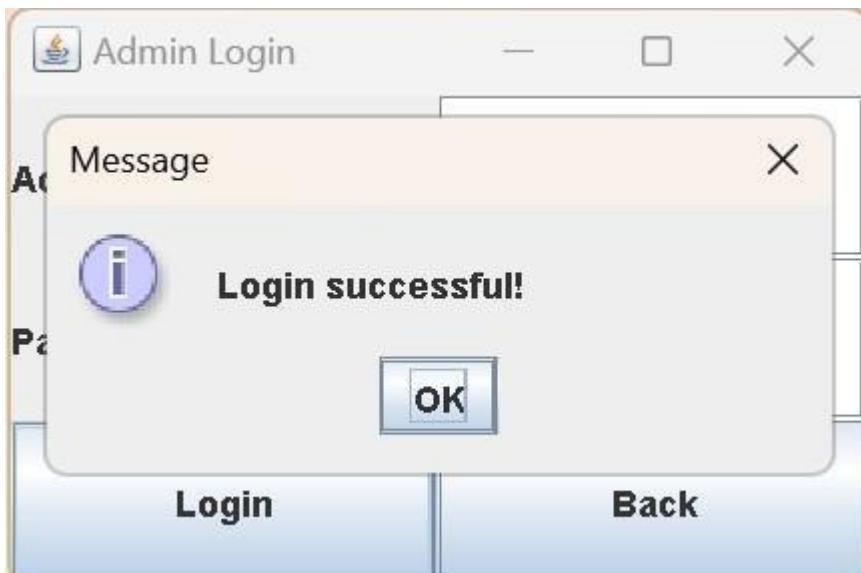


**Admin Name:**

**Password:**

**Login**

**Back**



A screenshot of the "Admin Dashboard" application window. The title bar shows the window title. The main content area contains five buttons arranged vertically:

- Add Movies**
- Add Theaters**
- Add Showtimes**
- View Movies and Showtimes**
- Logout**

Add Movie

Movie ID:	<input type="text"/>	Movie Title:
	<input type="text"/>	
Genre:	<input type="text"/>	
Duration (mins):	<input type="text"/>	Synopsis:
	<input type="text"/>	
Rating (0-10):		<input type="text"/>
Submit	<input type="button" value="Cancel"/>	

Add Showtime

Showtime ID:	<input type="text"/>	Movie ID:
	<input type="text"/>	
Theater ID:	<input type="text"/>	
Screen ID:	<input type="text"/>	Day:
	<input type="text"/>	
Date (YYYY-MM-DD):		<input type="text"/>
Submit	<input type="button" value="Cancel"/>	

Add Theater

Theater ID:	<input type="text"/>	Theater Name:
	<input type="text"/>	
	<input type="text"/>	Seating Capac...
Location:	<input type="text"/>	Screen ID:
	<input type="text"/>	
	<input type="text"/>	Food Add On:
Submit	<input type="button" value="Cancel"/>	



Movies and Showtimes



**Movie ID: 1, Title: Inception, Genre: Science Fiction, Duration: 2.5, Rating: 8.8**

**Showtime ID: 1, Theater ID: 1, Screen ID: 1, Day: Friday, Date: 2024-11-15T09:00**

**Movie ID: 2, Title: The Shawshank Redemption, Genre: Drama, Duration: 2.4, Rating: 9.3**

**Showtime ID: 2, Theater ID: 2, Screen ID: 2, Day: Saturday, Date: 2024-11-16T09:00**

**Movie ID: 3, Title: The Dark Knight, Genre: Action, Duration: 2.5, Rating: 9.0**

**Showtime ID: 3, Theater ID: 3, Screen ID: 3, Day: Sunday, Date: 2024-11-17T09:00**

**Showtime ID: 6, Theater ID: 2, Screen ID: 2, Day: Friday, Date: 2024-12-04T00:00**

**Movie ID: 4, Title: The Godfather, Genre: Crime, Duration: 2.9, Rating: 9.2**

**Showtime ID: 4, Theater ID: 4, Screen ID: 4, Day: Monday, Date: 2024-11-18T09:00**

**Movie ID: 6, Title: amaran, Genre: patriotic, Duration: 90.0, Rating: 8.7**

**Back**



User Login



**Username:**

**Password:**

**Login**

**Back**



**View Showtimes**

**Book Ticket**

**View Bookings**

**Logout**



Available Showtimes



**Showtime ID: 1, Movie ID: 1, Theater ID: 1, Screen ID: 1, Day: Friday, Date: 2024-11-15T09:00**

**Showtime ID: 2, Movie ID: 2, Theater ID: 2, Screen ID: 2, Day: Saturday, Date: 2024-11-16T09:00**

**Showtime ID: 3, Movie ID: 3, Theater ID: 3, Screen ID: 3, Day: Sunday, Date: 2024-11-17T09:00**

**Showtime ID: 4, Movie ID: 4, Theater ID: 4, Screen ID: 4, Day: Monday, Date: 2024-11-18T09:00**

**Showtime ID: 6, Movie ID: 3, Theater ID: 2, Screen ID: 2, Day: friday, Date: 2024-12-04T00:00**

**Back**

 Book Ticket

Showtime ID:

Seats (e.g., A1,A2):

**Book**      **Cancel**

 Book Ticket

Showtime ID: 3

Message X

 **Ticket booked successfully!**

**OK**



**Showtime ID: 1, Seats: A1, Status: Confirmed**

**Showtime ID: 4, Seats: C1, Status: Confirmed**

**Showtime ID: 8, Seats: A4, Status: Confirmed**

**Showtime ID: 3, Seats: D1,D2,D3, Status: Confirmed**

**Showtime ID: 3, Seats: C1,C2, Status: Confirmed**

---

**Back**