

Java Freshers

Candidate-ID	0012400075
Candidate Name	Rajesh kumar
Email	bodduna.rajeshkumar@gmail.com
Mobile	9182045463
Source	Naukari

Screening Test

Completed on : 13-Dec-2024 06:19 PM

#	Question	Response	Marks
1	Have you fully completed your education and prepared to work	Yes	1
2	Have you taken any long term training course on Java Technologies	Yes	1
3	Have you recently completed your education and training, and there is no major Gap (more than one year after the education)	Yes	1
4	Are you, willing to work on-premises (there is no work from home)	Yes	1
5	Can you write simple Java Programs on your own	Yes	1
6	Are you, available immediately to join us, if we hire you	Yes	1
7	Did you complete any kind of Developer Certification in Java	Yes	1
8	Do you know how to write unit testcases using Junit	Yes	1
9	if we hire you, will you be prepared to stay and work with us at least for two years	Yes	1
10	Will you be flexible any role in Software Engineering	Yes	1
11	Explain about yourself why you are the right candidate for this Job	video response	1

Overall Score : 11/11 (Promoted)

Coding Test

Completed on : 17-Dec-2024 06:37 PM

1. Write a program or method to find the smallest and largest numbers in an array of integers. Implement two methods: `findSmallest(int[] numbers)`: Returns the smallest number in the array. `findLargest(int[] numbers)`: Returns the largest number in the array. Sample Input : If the input array is null or empty, the method should throw an `IllegalArgumentException` with the message "Array cannot be null or empty". Use Java's Stream API to solve the problem. Expected Output : Smallest number: 1 Largest number: 20

Marks Scored : 2

```
import java.util.Arrays;
public class NumberUtils {
    public static void main(String[] args)
    {
        int a[]={4,8,90,23,12,3};
        System.out.println("Smallest number: "+findSmallest(a));
        System.out.println("Largest number: "+findLargest(a));
    }

    public static int findSmallest(int[] numbers) {
        if (numbers == null || numbers.length == 0) {
            throw new IllegalArgumentException("Array cannot be null or empty");
        }
        // Implement the logic here

        return Arrays.stream(numbers).min().getAsInt();

        // Placeholder return value
    }

    public static int findLargest(int[] numbers) {
        if (numbers == null || numbers.length == 0) {
            throw new IllegalArgumentException("Array cannot be null or empty");
        }
        // Implement the logic here

        return Arrays.stream(numbers).max().getAsInt();
        // Placeholder return value
    }
}
```

2. Write a Java program that counts the occurrences of words in a given input string that are of a specified minimum length or longer. The program should return a map of words sorted in descending order of their frequency. If two words have the same frequency, they should be sorted alphabetically in ascending order. Sample Input : Ignore case sensitivity (e.g., "Test" and "test" should be treated as the same word). Words are sequences of alphanumeric characters. Punctuation and other symbols should not be considered part of words. Words shorter than the specified minimum length should be excluded from the count. Expected Output : Words with minimum length 8: exceptional: 2 wonderful: 1

Marks Scored : 4

```
import java.util.*;
import java.util.stream.Collectors;

public class WordCounter {

    public static Map getWordsWithMinLength(String input, int minLength) {
        if (input == null || input.isEmpty()) {
            return Collections.emptyMap();
        }

        String[] words = input.split("\\W+");
        Map wordCount = new HashMap<>();

        for (String word : words) {
            if (word.length() >= minLength) {
                word=word.toLowerCase();
                wordCount.put(word, wordCount.getOrDefault(word, 0) + 1);
            }
        }

        return wordCount.entrySet()
            .stream()
            .sorted((e1, e2)->{
                int freqCompare=e2.getValue().compareTo(e1.getValue());
                return (freqCompare !=0) ? freqCompare : e1.getKey().compareTo(e2.getKey());
            })
            .collect(Collectors.toMap(
                Map.Entry::getKey,
                Map.Entry::getValue,
                (e1,e2)->e1,
                LinkedHashMap::new
            ));
    }

    //implement your code here
}

public static void main(String[] args)
{
    String input="This is an Exceptional day! yes, truly an exceptional and wonderful day";
    int minLength=8;

    Map result = getWordsWithMinLength(input,minLength);
    System.out.println("Words with minimum length "+minLength+":");
    result.forEach((word,count)->System.out.println(word+": "+count));
}
}
```

3. Question: Write Java Program to use and apply the concept of Design Pattern Singleton with simplest program. Expected Output: Instance 1: Singleton@15db9742 Instance 2: Singleton@15db9742 Both instances are the same (Singleton verified). Message from Singleton: Hello from Singleton!

Marks Scored : 6

```
public class Singleton {

    // Private static instance of the class
    private static Singleton instance;

    // Private constructor to prevent instantiation
    private Singleton() {}

    // Public method to provide access to the instance
    public static Singleton getInstance() {
        // Implement your code here
        if(instance == null){
            instance = new Singleton();
        }
        return instance;
    }

    // Example method to demonstrate functionality
    public String getMessage() {
        // Implement your code here
        return "Hello from Singleton!";
    }

    // Main method to test the Singleton functionality
    public static void main(String[] args) {
        // Implement your code here
        Singleton instance1 = Singleton.getInstance();
        System.out.println("Instance 1: "+instance1);

        Singleton instance2 = Singleton.getInstance();
        System.out.println("instance 2: "+instance2);

        if(instance1 == instance2){
            System.out.println("Are both instances the same (Singleton verified).");
        }else{
            System.out.println("Instance are different (Singleton failed)");
        }

        System.out.println("Message from Singleton: "+instance1.getMessage());
    }
}
```

Overall Score : 12/12 (Promoted)

Interview

Interviewed on : 21-Dec-2024 12:30 PM

Interviewd by : Srinivas Gadepalli

Interview Summary

Java Technology	1.0
Personal	1.0
Average interview rating	1.0

Question wise rating:

Java Technology	
JVM	1
JDK/JRE	1
Average rating	1.0

Personal	
Career Plan	1
Average rating	1.0

Interviewer Remarks:

No Remarks

Interviewer's Feedback:

Interviewer	Decision	Notes
Srinivas Gadepalli	Not Hire	little bit communication issues, okay kind of person