

Java Full Stack Developer

Candidate-ID	0012500072
Candidate Name	Sathwik Reguri
Email	reguri.sathwik@gmail.com
Mobile	9381470106
Source	Careers

Screening Test

Completed on : 11-Nov-2025 05:36 PM

#	Question	Response	Marks
1	Do you have professional experience between 1 to 5 years?	Yes	1
2	Would you be open to relocating to Hyderabad?	Yes	1
3	Are you familiar with Agile methodologies, including Scrum or Kanban?	Yes	1
4	Are you open to traveling for work when necessary, either domestically or internationally?	Yes	1
5	Will you be flexible any role in Software Engineering	Yes	1
6	Explain about yourself why you are the right candidate for this Job	video response	0
7	Have you recently completed your education and training, and there no is major Gap (more then one year after the education)	Yes	1
8	Are you, willing to work on-premises (there is no work from home)	Yes	1
9	Are you, available immediately to join us, if we hire you	Yes	1
10	if we hire you, will you be prepared to stay and work with us at least for two years	Yes	1

Overall Score : 9/10 (Promoted)

Coding Test

Completed on : 11-Nov-2025 05:54 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

```
public class NumberUtils {

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

        int smallest = numbers[0];
        for (int num : numbers) {
            if (num < smallest) {
                smallest = num;
            }
        }
        return smallest;
    }

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

        int largest = numbers[0];
        for (int num : numbers) {
            if (num > largest) {
                largest = num;
            }
        }
        return largest;
    }
}
```

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.trim().isEmpty()) {
            return Collections.emptyMap();
        }

        // Convert to lowercase to ensure case-insensitive counting
        String[] words = input.toLowerCase().split("\\W+");
        Map wordCount = new HashMap<>();

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

        // Sort by frequency (desc), then alphabetically (asc)
        return wordCount.entrySet()
            .stream()
            .sorted((a, b) -> {
                int freqCompare = b.getValue().compareTo(a.getValue());
                if (freqCompare == 0) {
                    return a.getKey().compareTo(b.getKey());
                }
                return freqCompare;
            })
            .collect(Collectors.toMap(
                Map.Entry::getKey,
                Map.Entry::getValue,
                (e1, e2) -> e1,
                LinkedHashMap::new
            ));
    }

    // Optional main method for local testing
    public static void main(String[] args) {
        String input = "This is a test. This test is simple and effective.";
        int minLength = 3;

        Map result = getWordsWithMinLength(input, minLength);
        System.out.println(result);
    }
}

```

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() {  
        if (instance == null) {  
            instance = new Singleton(); // Lazy initialization  
        }  
        return instance;  
    }  
  
    // Example method to demonstrate functionality  
    public String getMessage() {  
        return "Hello from Singleton!";  
    }  
  
    // Main method to test the Singleton functionality  
    public static void main(String[] args) {  
        Singleton obj1 = Singleton.getInstance();  
        Singleton obj2 = Singleton.getInstance();  
  
        System.out.println("Instance 1: " + obj1);  
        System.out.println("Instance 2: " + obj2);  
  
        if (obj1 == obj2) {  
            System.out.println("Both instances are the same (Singleton verified).");  
        } else {  
            System.out.println("Instances are different (Singleton failed).");  
        }  
  
        System.out.println("Message from Singleton: " + obj1.getMessage());  
    }  
}
```

Overall Score : 12/12 (Promoted)

Interview

Interviewed on : 11-Nov-2025 09:00 PM

Interviewd by : Srinivas Gadepalli

Interview Summary

Abstraction	2.0
Average interview rating	2.0

Question wise rating:

Abstraction	
Explain me with example usecase of Abstraction in Java ?	2
Average rating	2.0

Interviewer Remarks:

No Remarks

Interviewer's Feedback:

Interviewer	Decision	Notes
Srinivas Gadepalli	Not Hire	sound issue other wise everything fine