

Java Freshers

Candidate-ID	0012400035
Candidate Name	Raju Regulla
Email	raju.regulla@gmail.com
Mobile	7207084420

Screening Test

Completed on : 29-Nov-2024 02:11 AM

#	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)	No	0
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	Skipped	0

Overall Score : 9/11 (Promoted)

Coding Test

Completed on : 29-Nov-2024 08:14 AM

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");
        }
        return java.util.Arrays.stream(numbers).min().getAsInt();
    }

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

    public static void main(String[] args) {
        // Example test cases
        int[] numbers = {5, 3, 8, 1, 2};

        // Testing findSmallest
        int smallest = findSmallest(numbers);
        System.out.println("Smallest number: " + smallest); // Output: 1

        // Testing findLargest
        int largest = findLargest(numbers);
        System.out.println("Largest number: " + largest); // Output: 8
    }
}
```

2. Question: Unique Words Extractor Write a program to extract all unique words from a given text. A word is defined as a sequence of alphanumeric characters separated by non-word characters (e.g., spaces, punctuation). The program should meet the following criteria: Extract words that are at least 8 characters long. Convert all words to lowercase for uniqueness. Preserve the order of their first appearance in the text. Implement the functionality in the `extractUniqueWords` method. The method should return a `Set` containing the filtered words. Sample Input : "The development of artificial intelligence is advancing rapidly. Innovative technologies are shaping the future of the world." Expected Output : Unique words with at least 8 characters: development artificial intelligence advancing innovative technologies

Marks Scored : 4

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

public class WordProcessor {

    public static Set extractUniqueWords(String content) {
        if (content == null || content.isEmpty()) {
            return Collections.emptySet();
        }

        return Arrays.stream(content.split("\\W+")) // Split by non-word characters
            .filter(word -> word.length() >= 8) // Filter words with at least 8 characters
            .map(String::toLowerCase) // Convert to lowercase for uniqueness
            .collect(Collectors.toCollection(LinkedHashSet::new)); // Collect into a LinkedHashSet to maintain order
    }

    public static void main(String[] args) {
        // Example content for testing
        String content = "The development of artificial intelligence is advancing rapidly. " +
            "Innovative technologies are shaping the future of the world.";

        // Extract unique words with at least 8 characters
        Set uniqueWords = extractUniqueWords(content);

        // Print the unique words
        System.out.println("Unique words with at least 8 characters:");
        uniqueWords.forEach(System.out::println);
    }
}
```

3. Question Write a program to demonstrate most simple usage, application of design pattern prototype. Expected Output: Original: Shape[type=Circle, color=Red] Cloned: Shape[type=Circle, color=Red] After modifying cloned object: Original: Shape[type=Circle, color=Red] Cloned: Shape[type=Circle, color=Blue]

Marks Scored : 6

```
// PrototypeDemo.java
class Shape implements Cloneable {
    private String type;
    private int[] dimensions;

    public Shape(String type, int[] dimensions) {
        this.type = type;
        this.dimensions = dimensions;
    }

    public String getType() {
        return type;
    }
}
```

```

public int[] getDimensions() {
    return dimensions;
}

@Override
protected Shape clone() {
    try {
        Shape clone = (Shape) super.clone();
        clone.dimensions = this.dimensions.clone();
        return clone;
    } catch (CloneNotSupportedException e) {
        return null;
    }
}

public class PrototypeDemo {
    public static void main(String[] args) {
        Shape original = new Shape("Circle", new int[]{7});
        Shape cloned = original.clone();
        cloned.getDimensions()[0] = 14;
        System.out.println("Original Shape: " + original.getType() + " with dimensions " + java.util.Arrays.toString(original.getDimensions()));
        System.out.println("Cloned Shape: " + cloned.getType() + " with dimensions " + java.util.Arrays.toString(cloned.getDimensions()));
    }
}

```

Overall Score : 12/12 (Promoted)

Interview

Interviewed on : 02-Dec-2024 10:30 AM

Interviewd by : Srinivas Gadepalli

Interview Summary

OOPS	1.0
Java Programming	2.5
Java Technology	1.5
Average interview rating	1.67

Question wise rating:

OOPS	
UML	1

Abstraction	1
Inheritance	1
Average rating	1.0

Java Programming	
OOPS Implementation	2
Memory management	3
Average rating	2.5

Java Technology	
JVM	1
JDK/JRE	2
Average rating	1.5

Interviewer Remarks:

No Remarks

Interviewer's Feedback:

Interviewer	Decision	Notes
Srinivas Gadepalli	Not Hire	He didn't do any project, looks like he doesn't have any hands on training ..only book based knowledge