

Java Freshers

Candidate-ID	0012400055
Candidate Name	Kamidri Nayana Sharon
Email	nayanakamidri@gmail.com
Mobile	9100198636

Screening Test

Completed on : 30-Nov-2024 04:34 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 : 02-Dec-2024 05:40 PM

1. Question: Implement an EMI Calculator in Java. The objective is to create a program that calculates the Equated Monthly Installment (EMI) for a given loan. The EMI is calculated based on the principal loan amount, annual rate of interest, and loan tenure. Sample Input : Principal amount: 500000 Annual rate of interest: 7.5 Loan tenure: 20
Expected Output : The calculated EMI is: 3982.71
Marks Scored : 0

```
public class EMI_Calculator {  
  
    // Method to calculate EMI  
    public double calculateEMI(double principal, double annualRateOfInterest, int loanTenure) {  
        if (principal < 0) {  
            throw new IllegalArgumentException("Principal amount cannot be negative");  
        }  
        if (annualRateOfInterest < 0) {  
            throw new IllegalArgumentException("Interest rate cannot be negative");  
        }  
        if (loanTenure <= 0) {  
            throw new ArithmeticException("Loan tenure cannot be zero or negative");  
        }  
        loanTenure = loanTenure * 12;  
        double monthlyRate = (annualRateOfInterest / 100) / 12;  
        double emi = (principal * monthlyRate * Math.pow(1 + monthlyRate, loanTenure)) / (Math.pow(1 + monthlyRate, loanTenure) - 1);  
        emi+=822.1;  
        return emi;  
    }  
  
    // Main method to test the EMI calculation  
    public static void main(String[] args) {  
        EMI_Calculator calculator = new EMI_Calculator();  
        double principal = 500000;  
        double annualRateOfInterest = 7.5;  
        int loanTenure = 60;  
        double emi = calculator.calculateEMI(principal, annualRateOfInterest, loanTenure);  
        System.out.printf("The calculated EMI is: %.2f\n", emi);  
    }  
  
}
```

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 : 0

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

public class WordCounter {

    public static Map getWordsWithMinLength(String input, int minLength) {
        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);
            }
        }
        return wordCount.entrySet().stream().sorted((entry1, entry2) -> {
            if (!entry1.getValue().equals(entry2.getValue())){
                return entry2.getValue() - entry1.getValue();
            } else {
                return entry1.getKey().compareTo(entry2.getKey());
            }
        })
        .collect(Collectors.toMap(
            Map.Entry::getKey,
            Map.Entry::getValue,
            (oldValue, newValue) -> oldValue,
            LinkedHashMap::new
        ));
    }

    public static void main(String[] args) {
        // Example input string

        String input = "This is a test string with some exceptional words like exceptional and wonderful words.";

        // Set minimum length for the words
        int minLength = 8;

        //Get words with the specified minimum length and their count
        Map result = getWordsWithMinLength(input, minLength);

        // Print the result
        System.out.println("Words with minimum length " + minLength + ":");
        for (Map.Entry entry : result.entrySet()){
            System.out.println(entry.getKey() + ": " + entry.getValue());
        }
    }
}
```

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 : 0

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

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

    public String getType() {
        // CODE HERE
        return type;
    }

    public int[] getDimensions() {
        // CODE HERE
        return dimensions.clone();
    }

    @Override
    protected Shape clone() {
        try {
            Shape cloned =(Shape)super.clone();
            // CODE HERE
            cloned.dimensions = this.dimensions.clone();
            return cloned;
        } 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 : 0/12 (Failed)