

Java Freshers

Candidate-ID	0012400057
Candidate Name	Minnakurthi Satish
Email	minnakurthisatishk@gmail.com
Mobile	7993520528

Screening Test

Completed on : 30-Nov-2024 03:19 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	No	0
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	Skipped	0

Overall Score : 9/11 (Promoted)

Coding Test

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("amount can not be negative");
        }
        if(annualRateOfInterest < 0) {
            throw new IllegalArgumentException("Interest rate cannot be negative");
        }
        if(loanTenure <= 0){
            throw new ArithmeticException("loanTenure cannot be zero");
        }

        double rateOfInterest = annualRateOfInterest / 12 / 100;

        int numberOfInstallments = loanTenure * 12;

        double emi = (principal * rateOfInterest * Math.pow(1 + rateOfInterest, numberOfInstallments)) / (Math.pow(1 + rateOfInterest, numberOfInstallments));

        return emi;
    }

    // Main method to test the EMI calculation
    public static void main(String[] args) {
        // Create an instance of EMI_Calculator
        EMI_Calculator calculator = new EMI_Calculator();

        // Sample inputs
        double principal = 500000; // Example principal amount (e.g., 5,00,000)
        double annualRateOfInterest = 7.5; // Example annual interest rate (e.g., 7.5%)
        int loanTenure = 20; // Example loan tenure in years (e.g., 20 years)

        // Calculate EMI
        double emi = calculator.calculateEMI(principal, annualRateOfInterest, loanTenure);

        // Output the result
        System.out.println("The calculated EMI is: " + 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) {
        // CODE HERE

        return
    }

    public static void main(String[] args) {
        // Example input string
        String: "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 wordCounts = getWordsWithMinLength(input, minLength);

        // Print the result
        System.out.println("Words with minimum length " + minLength + ":");
        wordCounts.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 : 0

Skipped

Overall Score : 0/12 (Failed)