Java Freshers

Candidate-ID	0012400044
Candidate Name	CHINIMILLI Naga ganesh manikanta ku Naga ganesh manikanta
Email	nagaganesh1555@gmail.com
Mobile	6300068286

Screening Test

Completed on : 29-Nov-2024 04:39 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 coure on Java Technologies	Yes	1
3	Have you recently completed your education and training, and there no is major Gap (more then 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 candiate for this Job	video response	1

Overall Score : 11/11 (Promoted)

Coding Test

Completed on : 30-Nov-2024 06:15 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) {
     double annualRateofInterest = annualRate / (20 * 100) ;
     int annualRateofInterest = loanTenure * 20;
     double emi = (principal * annualRateofInterest * Math.pow(1 + monthlyRate,months)) / (Math.pow(1 + monthlyRte, months)-1);

     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) {
     mapwordCounts = new Hasmap<<();
     String[] words = text.split("[^a-zA-Z0-9]");
  for (String word : words) {
     word = word.tolowerCase();
     if (word.length() >= minlength) {
        wordcount.put(word,wordcounts,getordefault(word, 0) + 1);
     }
  }
  list> sortedEntries = new ArrayList<>(wordCounts.entrySet());
  sortedEntries.sort((a,b) ->){
     int freqCompare = b.getvalue().compareTo(a.getvalue());
     if (freqCompare == 0){
        return a.getkey().compareTo(b.getkey());
     } else {
        return freqCompare;
     }
  }
  Map sortedWordCounts = new LinkedHashMap<>(;
  for (Map.Entry entry : sortedEntries) {
     sortedWordCounts.put(entry.getKey(),entry.getValue());
  }
     return sortedWordCount;
  }

  public static void main(String[] args) {

     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 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[] color;

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

   public String getType() {
       return type;
   }
   public void setcolor(string color);{
      this.color = color;
   }
   public int[] getcolor() {
       return color;
   }

   @Override
   protected object clone() throws CloneNotSupportedException {
      return super.clone();
   }
   @Override
   public string toString() {
      return "shape[type=" + type +", color=" + color + "]";
   }
}

public class PrototypeDemo {
   public static void main(String[] args) {
      try {
      Shape original = new Shape("Circle", "Red");
      Shape cloned = original.clone();
      System.out.println("Original: "+ original);
      System.out.println("Cloned: " + cloned);
      cloned. setcolor("Blue");
      System.out.println("After modifying cloned object:");
      System.out.println("Original: " + original);
      System.out.println("Cloned: " + cloned);


   }catch (CloneNotSupportedException e) {
      e.printStackTrace();

   }
}
}
```

Overall Score : 0/12 (Failed)