

DAY 25:

ASSIGNMENT 1:

Task 1: Java IO Basics

Write a program that reads a text file and counts the frequency of each word using `FileReader` and `FileWriter`.

ANSWER:

```
import java.io.*;
import java.util.*;

public class WordFrequencyCounter {

    public static void main(String[] args) {
        // Check if the input file is provided
        if (args.length < 2) {
            System.out.println("Usage: java WordFrequencyCounter <inputFile> <outputFile>");
            return;
        }

        String inputFile = args[0];
        String outputFile = args[1];

        // Use a TreeMap to store word frequencies (sorted by word)
        Map<String, Integer> wordCounts = new TreeMap<>();

        try (FileReader fileReader = new FileReader(inputFile);
            BufferedReader bufferedReader = new BufferedReader(fileReader)) {
```

```

String line;

while ((line = bufferedReader.readLine()) != null) {

    // Split the line into words using space and punctuation as delimiters

    String[] words = line.split("\\W+");


    for (String word : words) {

        if (!word.isEmpty()) {

            word = word.toLowerCase(); // Convert word to lowercase

            wordCounts.put(word, wordCounts.getOrDefault(word, 0) + 1);

        }

    }

}

} catch (IOException e) {

    System.out.println("Error reading the file: " + e.getMessage());

}

try (FileWriter fileWriter = new FileWriter(outputFile);

    BufferedWriter bufferedWriter = new BufferedWriter(fileWriter)) {

    for (Map.Entry<String, Integer> entry : wordCounts.entrySet()) {

        bufferedWriter.write(entry.getKey() + ": " + entry.getValue());

        bufferedWriter.newLine();

    }

} catch (IOException e) {

    System.out.println("Error writing the file: " + e.getMessage());

}

}

}

```

Explanation:

1. Reading the File:

- The program uses `FileReader` and `BufferedReader` to read the input file line by line.
- Each line is split into words using a regular expression (`\\W+`), which matches any non-word character (i.e., anything that's not a letter, digit, or underscore).

2. Counting Word Frequencies:

- A `TreeMap` is used to store word frequencies, ensuring the words are stored in alphabetical order.
- For each word, the program converts it to lowercase (to ensure case-insensitivity) and updates the word count.

3. Writing the Output:

- The program writes the word frequencies to the output file using `FileWriter` and `BufferedWriter`.
- Each word and its frequency are written on a new line in the format "word: frequency".