Assignment Number: **4.3**(Present assignment number)/**24**(Total number of assignments)		
B.NITHIN	**2303a51803**	**B-26**

Q.No.	Question	*Expected Time to complete*
1	**Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques** **Lab Objectives** • To explore and apply different levels of prompt examples in AI-assisted code generation • To understand how zero-shot, one-shot, and few-shot prompting affect AI output quality • To evaluate the impact of context richness and example quantity on AI performance • To build awareness of prompt strategy effectiveness for different problem types **Lab Outcomes (LOs)** **After completing this lab, students will be able to:** • Use zero-shot prompting to instruct AI with minimal context • Use one-shot prompting with a single example to guide AI code generation • Apply few-shot prompting using multiple examples to improve AI responses • Compare AI outputs across different prompting strategies	Week2 - Wednesday
	Task 1: Zero-Shot Prompting – Leap Year Check **Scenario** Zero-shot prompting involves giving instructions without providing examples. **Task Description** Use zero-shot prompting to instruct an AI tool to generate a Python function that: • Accepts a year as input • Checks whether the given year is a leap year • Returns an appropriate result **Note:** No input-output examples should be provided in the prompt. **Expected Output**	

LeapYearChecker.java > Language Support for Java(TM) by Red Hat > LeapYearChecker

```java
//generate the java code for check the year is leap year or not and take the input from user
import java.util.Scanner;

public class LeapYearChecker {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s: "Enter a year: ");
        int year = scanner.nextInt();

        if (isLeapYear(year)) {
            System.out.println(year + " is a leap year.");
        } else {
            System.out.println(year + " is not a leap year.");
        }

        scanner.close();
    }

    public static boolean isLeapYear(int year) {
        return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
    }
}
```

PROBLEMS 13 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\nitin\OneDrive\Desktop\ai>  & 'C:\Program Files\Java\jdk-25\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\nitin\AppData\Roaming\Code\User\workspaceStorage\d5b583de86868a227dcb4815ce34c8c7\redhat.java\jdt_ws\ai_df4ca976\bin' 'LeapYearChecker'
Enter a year: 2024
2024 is a leap year.
PS C:\Users\nitin\OneDrive\Desktop\ai>
```

Task 2: One-Shot Prompting – Centimeters to Inches Conversion

Scenario

One-shot prompting guides AI using a single example.

Task Description

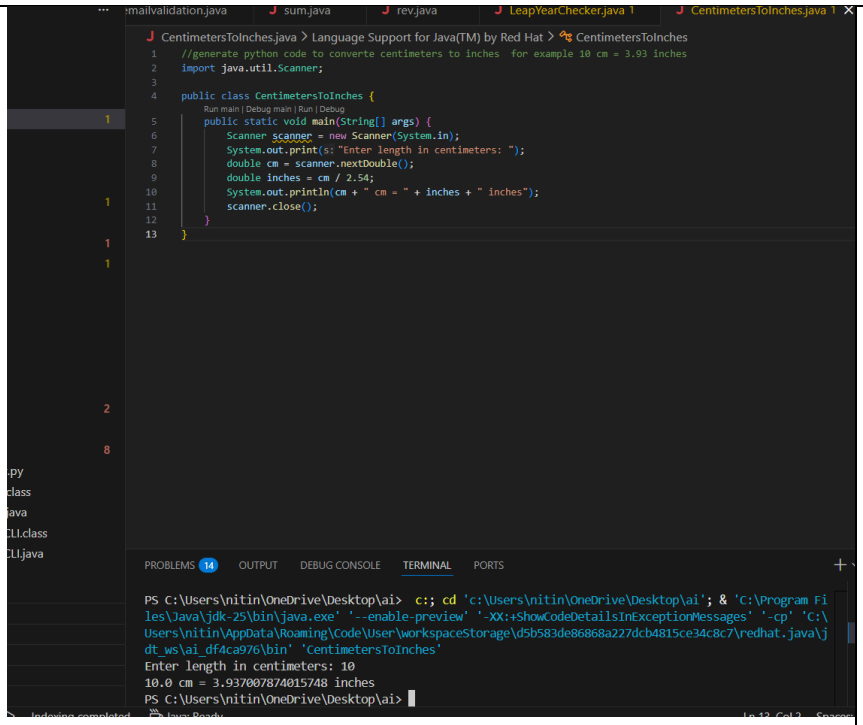
Use one-shot prompting by providing one input-output example to generate a Python function that:

- Converts centimeters to inches
- Uses the correct mathematical formula

Example provided in prompt:

Input: 10 cm → Output: 3.94 inches

Expected Output



Task 3: Few-Shot Prompting – Name Formatting

Scenario

Few-shot prompting improves accuracy by providing multiple examples.

Task Description

Use few-shot prompting with 2–3 examples to generate a Python function that:

- Accepts a full name as input
- Formats it as “Last, First”

Example formats:

- "John Smith" → "Smith, John"
- "Anita Rao" → "Rao, Anita"

Expected Output

NameFormatter.java > Language Support for Java(TM) by Red Hat > NameFormatter

```java
//generate a java code that accept the full name as input as a single string and then format it to "Last Name, First N
import java.util.Scanner;

public class NameFormatter {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s: "Enter your full name: ");
        String fullName = scanner.nextLine();

        String formattedName = formatName(fullName);
        System.out.println("Formatted name: " + formattedName);

        scanner.close();
    }

    public static String formatName(String fullName) {
        String[] parts = fullName.trim().split(regex: "\\s+");
        if (parts.length < 2) {
            return fullName;
        }

        String lastName = parts[parts.length - 1];
        StringBuilder firstNameAndMiddleNames = new StringBuilder();

        for (int i = 0; i < parts.length - 1; i++) {
            firstNameAndMiddleNames.append(parts[i]).append(str: " ");
        }

        return lastName + ", " + firstNameAndMiddleNames.toString().trim();
    }
}
```

PROBLEMS 15 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\nitin\OneDrive\Desktop\ai>  c:; cd 'c:\Users\nitin\OneDrive\Desktop\ai'; & 'C:\Program Fi
les\Java\jdk-25\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\
Users\nitin\AppData\Roaming\Code\User\workspaceStorage\d5b583de86868a227dcb4815ce34c8c7\redhat.java\j
dt_ws\ai_df4ca976\bin' 'NameFormatter'
Enter your full name: bodasu nithin
Formatted name: nithin, bodasu
PS C:\Users\nitin\OneDrive\Desktop\ai>
```

Task 4: Comparative Analysis – Zero-Shot vs Few-Shot

Scenario

Different prompt strategies may produce different code quality.

Task Description

- Use zero-shot prompting to generate a function that counts vowels in a string
- Use few-shot prompting for the same problem
- Compare both outputs based on:
    - Accuracy
    - Readability
    - Logical clarity

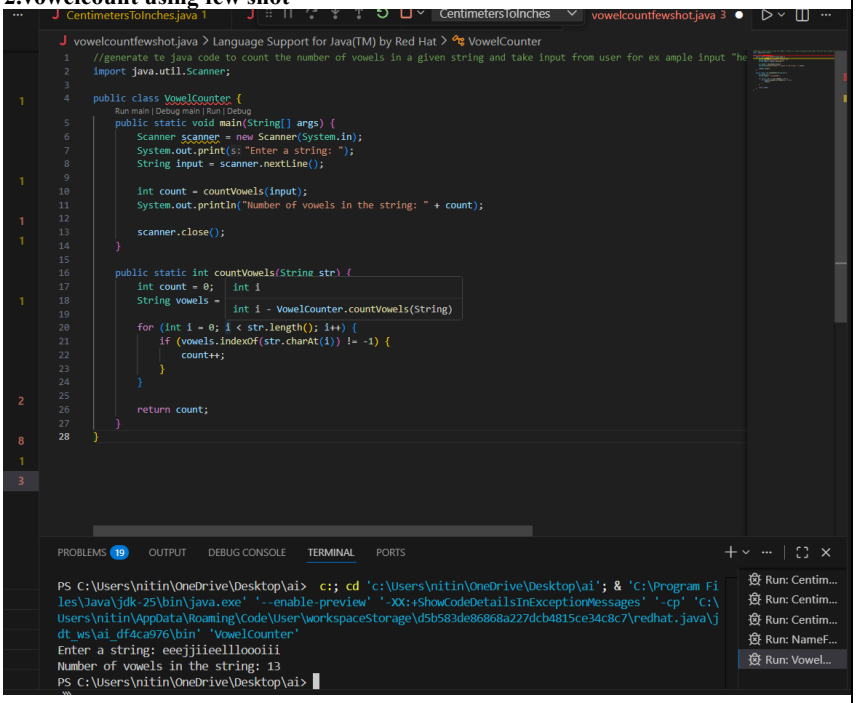
Expected Output

1.vowel count using zero shot

```java
public class VowelCounter {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s: "Enter a string: ");
        String input = scanner.nextLine();

        int count = countVowels(input);
        System.out.println("Number of vowels in the string: " + count);

        scanner.close();
    }

    public static int countVowels(String str) {
        int count = 0;
        String vowels = "aeiouAEIOU";

        for (int i = 0; i < str.length(); i++) {
            if (vowels.indexOf(str.charAt(i)) != -1) {
                count++;
            }
        }

        return count;
    }
}
```

```
LEMS 16  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

:\Users\nitin\OneDrive\Desktop\ai>  c:; cd 'c:\Users\nitin\OneDrive\Desktop\ai'; & 'C:\Program F
\Java\jdk-25\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:
s\nitin\AppData\Roaming\Code\User\workspaceStorage\d5b583de86868a227dcb4815ce34c8c7\redhat.java\
ws\ai_df4ca976\bin' 'VowelCounter'
er a string: abhiiramesh
ber of vowels in the string: 5
:\Users\nitin\OneDrive\Desktop\ai>
```

2.vowelcount using few shot



Task 5: Few-Shot Prompting – File Handling

Scenario

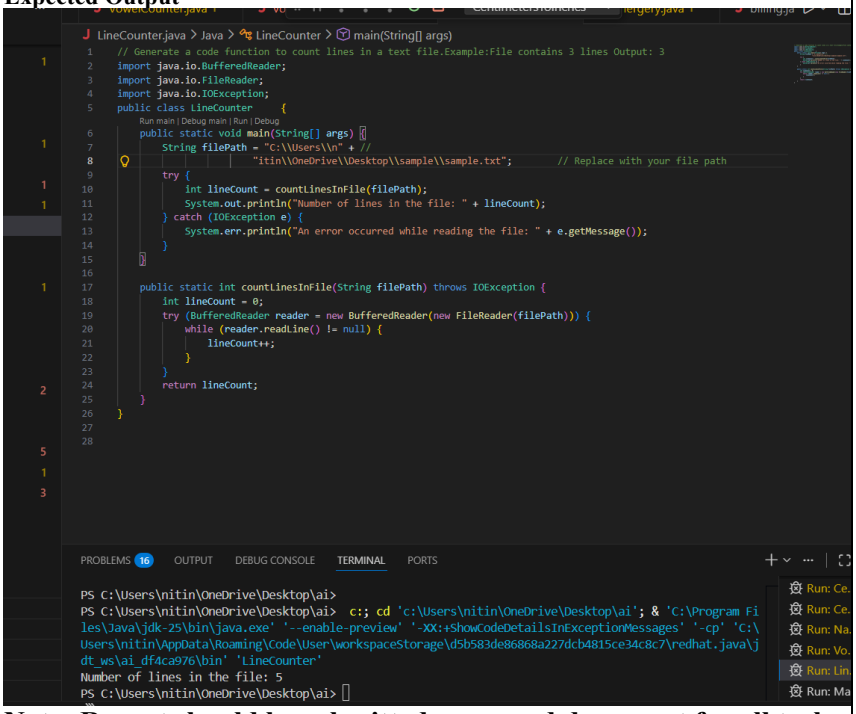
File processing requires clear logical understanding.

Task Description

Use few-shot prompting to generate a Python function that:

- Reads a .txt file
- Counts the number of lines in the file
- Returns the line count

Expected Output



Note: Report should be submitted as a word document for all tasks in a single document with prompts, comments & code explanation, and output and if required, screenshots.