

<b>Assignment Number:</b> 4.3(Present assignment number)/24(Total number of assignments)		
<b>B.NITHIN</b>	<b>2303a51803</b>	<b>B-26</b>
<b>Q.No.</b>	<b>Question</b>	<b>Expected Time to complete</b>

<b>Q.No.</b>	<b>Question</b>	<b>Expected Time to complete</b>
1	<p><b>Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques</b></p> <p><b>Lab Objectives</b></p> <ul style="list-style-type: none"> <li>• To explore and apply different levels of prompt examples in AI-assisted code generation</li> <li>• To understand how zero-shot, one-shot, and few-shot prompting affect AI output quality</li> <li>• To evaluate the impact of context richness and example quantity on AI performance</li> <li>• To build awareness of prompt strategy effectiveness for different problem types</li> </ul> <p><b>Lab Outcomes (LOs)</b></p> <p><b>After completing this lab, students will be able to:</b></p> <ul style="list-style-type: none"> <li>• Use zero-shot prompting to instruct AI with minimal context</li> <li>• Use one-shot prompting with a single example to guide AI code generation</li> <li>• Apply few-shot prompting using multiple examples to improve AI responses</li> <li>• Compare AI outputs across different prompting strategies</li> </ul>	Week2 - Wednesday
	<p><b>Task 1: Zero-Shot Prompting – Leap Year Check</b></p> <p><b>Scenario</b></p> <p>Zero-shot prompting involves giving instructions without providing examples.</p> <p><b>Task Description</b></p> <p>Use zero-shot prompting to instruct an AI tool to generate a Python function that:</p> <ul style="list-style-type: none"> <li>• Accepts a year as input</li> <li>• Checks whether the given year is a leap year</li> <li>• Returns an appropriate result</li> </ul> <p><b>Note:</b> No input-output examples should be provided in the prompt.</p> <p><b>Expected Output</b></p>	

	<pre>J LeapYearChecker.java &gt; Language Support for Java(TM) by Red Hat &gt; ⚙ LeapYearChecker 1  //generate the java code for check the year is leap year or not and take the input from user 2  import java.util.Scanner; 3 4  public class LeapYearChecker { 5      Run main   Debug main   Run   Debug 6      public static void main(String[] args) { 7          Scanner scanner = new Scanner(System.in); 8          System.out.print("Enter a year: "); 9          int year = scanner.nextInt(); 10 11         if (isLeapYear(year)) { 12             System.out.println(year + " is a leap year."); 13         } else { 14             System.out.println(year + " is not a leap year."); 15         } 16 17         scanner.close(); 18     } 19 20     public static boolean isLeapYear(int year) { 21         return (year % 4 == 0 &amp;&amp; year % 100 != 0)    (year % 400 == 0); 22     } </pre> <p>PROBLEMS 13    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS</p> <pre>PS C:\Users\nitin\OneDrive\Desktop\ai&gt; &amp; 'C:\Program Files\Java\jdk-25\bin\java.exe' '--enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\nitin\AppData\Roaming\Code\User\workspaceStorage\d5b583de86868a227dc4815ce34c8c7\redhat.java\jdt_ws\ai_df4ca976\bin' 'LeapYearChecker' Enter a year: 2024 2024 is a leap year. PS C:\Users\nitin\OneDrive\Desktop\ai&gt;</pre>
	<p><b>Task 2: One-Shot Prompting – Centimeters to Inches Conversion</b></p> <p><b>Scenario</b> One-shot prompting guides AI using a single example.</p> <p><b>Task Description</b> Use one-shot prompting by providing one input-output example to generate a Python function that:</p> <ul style="list-style-type: none"> <li>• Converts centimeters to inches</li> <li>• Uses the correct mathematical formula</li> </ul> <p><b>Example provided in prompt:</b> Input: 10 cm → Output: 3.94 inches</p> <p><b>Expected Output</b></p>

```
CentimetersToInches.java // generate python code to convert centimeters to inches
import java.util.Scanner;
public class CentimetersToInches {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter length in centimeters: ");
        double cm = scanner.nextDouble();
        double inches = cm / 2.54;
        System.out.println(cm + " cm = " + inches + " inches");
        scanner.close();
    }
}
```

PROBLEMS 14 OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\nitin\OneDrive\Desktop\ai> cd 'c:\Users\nitin\OneDrive\Desktop\ai' & 'c:\Program Files\Java\jdk-25\bin\java.exe' '-enable-preview' '-XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'G:\Users\nitin\AppData\Roaming\Code\User\workspaceStorage\d50583de86868a227dc4815ce34c8c7\redhat\java\dt_ws\ai_1_dfc4a976\bin' 'CentimetersToInches'
Enter length in centimeters: 10
10.0 cm = 3.937007874015748 inches
PS C:\Users\nitin\OneDrive\Desktop\ai>
```

### 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

```

J NameFormatter.java > Language Support for Java(TM) by Red Hat >  NameFormatter
1 //generate a java code that accept the full name as input as a single string and then format it to "Last Name, First N
2 import java.util.Scanner;
3
4 public class NameFormatter {
5     Run main | Debug main | Run | Debug
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8         System.out.print("Enter your full name: ");
9         String fullName = scanner.nextLine();
10
11         String formattedName = formatName(fullName);
12         System.out.println("Formatted name: " + formattedName);
13
14         scanner.close();
15     }
16
17     public static String formatName(String fullName) {
18         String[] parts = fullName.trim().split(regex: "\\s+");
19         if (parts.length < 2) {
20             return fullName;
21         }
22
23         String lastName = parts[parts.length - 1];
24         StringBuilder firstNameAndMiddleNames = new StringBuilder();
25
26         for (int i = 0; i < parts.length - 1; i++) {
27             firstNameAndMiddleNames.append(parts[i]).append(str: " ");
28         }
29
30         return lastName + ", " + firstNameAndMiddleNames.toString().trim();
31     }

```

PROBLEMS 15    OUTPUT    DEBUG CONSOLE    **TERMINAL**    PORTS

```

PS C:\Users\nitin\OneDrive\Desktop\ai> c:; cd '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\d5b583de86868a227dcba815ce34c8c7\redhat.java\jdt_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**

```
public class VowelCounter {
    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("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;
    }
}

ELEMES 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS

C:\Users\nitin\OneDrive\Desktop\ai> c:; cd '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\d5b583de86868a227dcba4815ce34c8c7\redhat.java\dt_ws\ai_df4ca976\bin' 'VowelCounter'
Enter a string: abhiramsh
Number of vowels in the string: 5
C:\Users\nitin\OneDrive\Desktop\ai> 2.vowelcount using few shot
```

CentimetersToInches.java 1 J vowelcountfewshot.java 3 ● ▶ □ ...

```
J vowelcountfewshotjava > Language Support for Java(TM) by Red Hat > VowelCounter
1 // generate te Java code to count the number of vowels in a given string and take Input from user for ex ample input "he
2 import java.util.Scanner;
3
4 public class VowelCounter {
5     Run main | Debug main | Run | Debug
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8         System.out.print("Enter a string: ");
9         String input = scanner.nextLine();
10
11         int count = countVowels(input);
12         System.out.println("Number of vowels in the string: " + count);
13
14     }
15
16     public static int countVowels(String str) {
17         int count = 0; int i
18         String vowels = "aeiouAEIOU";
19         for (int i = 0; i < str.length(); i++) {
20             if (vowels.indexOf(str.charAt(i)) != -1) {
21                 count++;
22             }
23         }
24
25         return count;
26     }
27 }
28 }
```

PROBLEMS 19 OUTPUT DEBUG CONSOLE TERMINAL PORTS + × | ×

```
C:\Users\nitin\OneDrive\Desktop\ai> c:; cd '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\d5b583de86868a227dcba4815ce34c8c7\redhat.java\dt_ws\ai_df4ca976\bin' 'VowelCounter'
Enter a string: eeejjiieellloooiiii
Number of vowels in the string: 13
PS C:\Users\nitin\OneDrive\Desktop\ai>
```

## Task 5: Few-Shot Prompting – File Handling

### **Task 5. 1 Scenario**

**Scenario**  
File processing requires clear logical understanding.

### **Task Description**

**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

The screenshot shows a Java code editor with the file `LineCounter.java` open. The code implements a class `LineCounter` that reads a file and counts the number of lines. The terminal below shows the execution of the code and its output.

```

J LineCounter.java > Java > LineCounter > main(String[] args)
1 // Generate a code function to count lines in a text file.Example:File contains 3 lines Output: 3
2 import java.io.BufferedReader;
3 import java.io.FileReader;
4 import java.io.IOException;
5 public class LineCounter {
6     Run main | Debug main | Run | Debug
7     public static void main(String[] args) {
8         String filePath = "C:\\Users\\n\\Desktop\\sample\\sample.txt"; // Replace with your file path
9         try {
10             int lineCount = countLinesInFile(filePath);
11             System.out.println("Number of lines in the file: " + lineCount);
12         } catch (IOException e) {
13             System.err.println("An error occurred while reading the file: " + e.getMessage());
14         }
15     }
16
17     public static int countLinesInFile(String filePath) throws IOException {
18         int lineCount = 0;
19         try (BufferedReader reader = new BufferedReader(new FileReader(filePath))) {
20             while (reader.readLine() != null) {
21                 lineCount++;
22             }
23         }
24         return lineCount;
25     }
26 }
27
28
5
1
3

```

TERMINAL

```

PROBLEMS 16 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\n\OneDrive\Desktop\ai>
PS C:\Users\n\OneDrive\Desktop\ai> c:; cd 'c:\Users\n\OneDrive\Desktop\ai'; & 'C:\Program Files\Java\jdk-25\bin\java.exe' '--enable-preview' '--XX:+ShowCodeDetailsInExceptionMessages' '-cp' 'C:\Users\n\AppData\Roaming\Code\User\workspaceStorage\d5b583de86868a227dcba4815ce34c8c7\redhat\java\dt_wsl\ai_dfacac976\bin' 'linecounter'
Number of lines in the file: 5
PS C:\Users\n\OneDrive\Desktop\ai>

```

+ · | ○

Run: Ce.  
Run: Ce.  
Run: Na.  
Run: Vo.  
Run: Lin.  
Run: Ma.

**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.