

# LAB ASSIGNMENT-4.2

**Hall Ticket no : 2303A51430**

**Name : MADASI SATHWIK**

## **Task 1: Zero-Shot Prompting – Leap Year Check.**

### **Scenario:**

Zero-shot prompting involves giving instructions without providing examples.

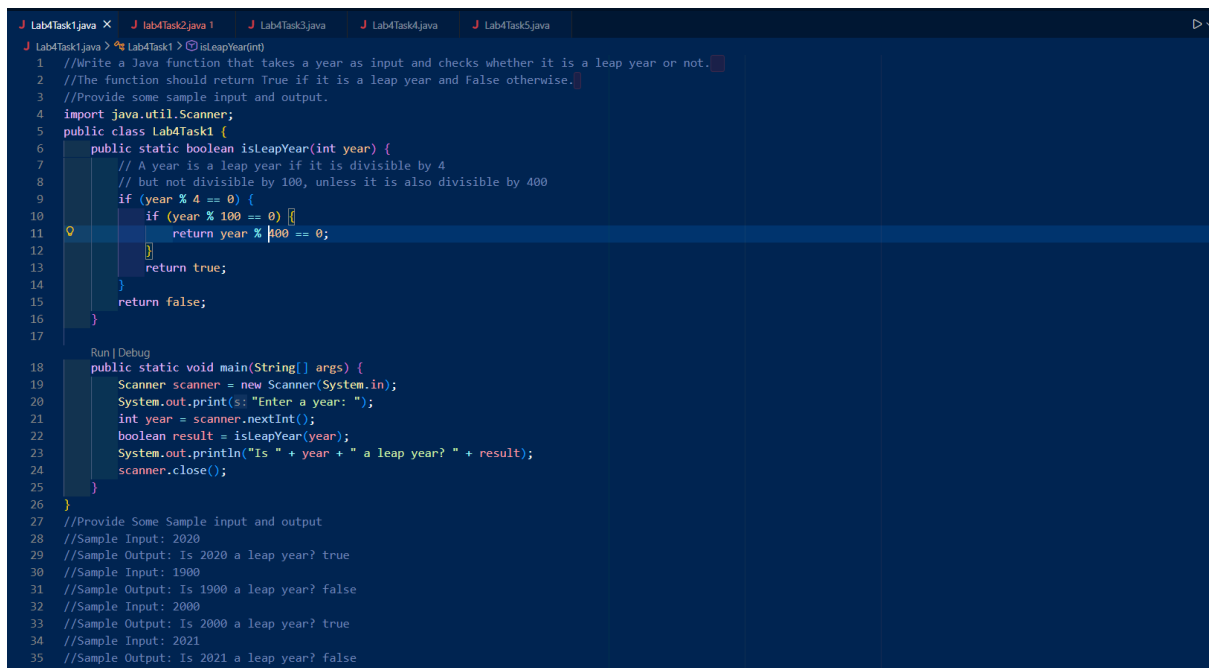
### **Prompt Used:**

**//Write a Java function that takes a year as input and checks whether it is a leap year or not.**

**//The function should return True if it is a leap year and False otherwise.**

**//Provide some sample input and output.**

### **CODE:**



```
1 //Write a Java function that takes a year as input and checks whether it is a leap year or not.
2 //The function should return True if it is a leap year and False otherwise.
3 //Provide some sample input and output.
4 import java.util.Scanner;
5 public class Lab4Task1 {
6     public static boolean isLeapYear(int year) {
7         // A year is a leap year if it is divisible by 4
8         // but not divisible by 100, unless it is also divisible by 400
9         if (year % 4 == 0) {
10             if (year % 100 == 0) {
11                 return year % 400 == 0;
12             }
13             return true;
14         }
15         return false;
16     }
17
18     public static void main(String[] args) {
19         Scanner scanner = new Scanner(System.in);
20         System.out.print("Enter a year: ");
21         int year = scanner.nextInt();
22         boolean result = isLeapYear(year);
23         System.out.println("Is " + year + " a leap year? " + result);
24         scanner.close();
25     }
26 }
27 //Provide Some Sample input and output
28 //Sample Input: 2020
29 //Sample Output: Is 2020 a leap year? true
30 //Sample Input: 1900
31 //Sample Output: Is 1900 a leap year? false
32 //Sample Input: 2000
33 //Sample Output: Is 2000 a leap year? true
34 //Sample Input: 2021
35 //Sample Output: Is 2021 a leap year? false
```

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

### **Scenario:**

**One-shot prompting guides AI using a single example.**

### **Prompt Used:**

**//Write a Java function to convert centimeters to inches.**

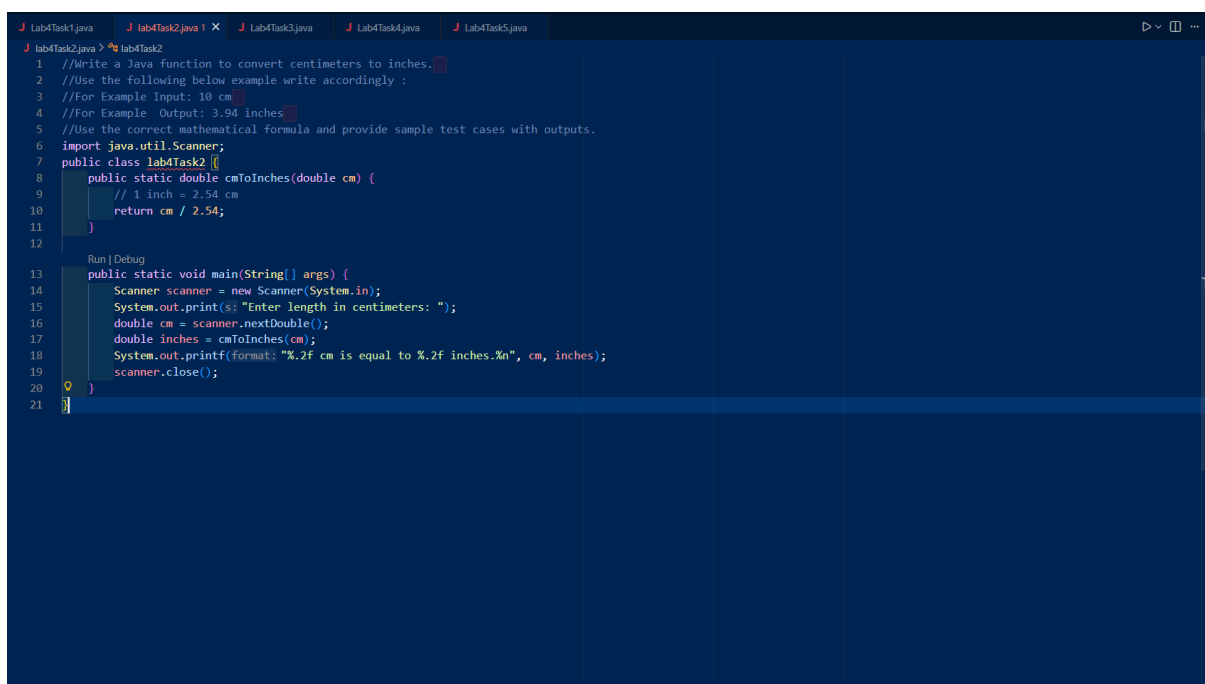
**//Use the following below example write accordingly :**

**//For Example Input: 10 cm**

**//For Example Output: 3.94 inches**

**//Use the correct mathematical formula and provide sample test cases with outputs.**

### **CODE:**

A screenshot of an IDE window showing a Java file named 'lab4Task2.java'. The code implements a static method 'cmToInches' that takes a double 'cm' and returns its value in inches using the formula 'cm / 2.54'. It also includes a 'main' method that uses a 'Scanner' to take user input, call the conversion method, and print the result with two decimal places. The code is as follows:

```
1 //Write a Java function to convert centimeters to inches.
2 //Use the following below example write accordingly :
3 //For Example Input: 10 cm
4 //For Example Output: 3.94 inches
5 //Use the correct mathematical formula and provide sample test cases with outputs.
6 import java.util.Scanner;
7 public class lab4Task2 {
8     public static double cmToInches(double cm) {
9         // 1 inch = 2.54 cm
10        return cm / 2.54;
11    }
12
13    public static void main(String[] args) {
14        Scanner scanner = new Scanner(System.in);
15        System.out.print("Enter length in centimeters: ");
16        double cm = scanner.nextDouble();
17        double inches = cmToInches(cm);
18        System.out.printf(format: "%.2f cm is equal to %.2f inches.\n", cm, inches);
19        scanner.close();
20    }
21 }
```

## **Task 3: Few-Shot Prompting – Name Formatting**

### **Scenario:**

**Few-shot prompting improves accuracy by providing multiple examples.**

### **Prompt Used:**

**//Write a Java function that takes a full name as input and formats it as "Last, First".**

**//Examples of Input and Output:**

**//Input: "Vamshi Raju" → Output: "Raju, Vamshi"**

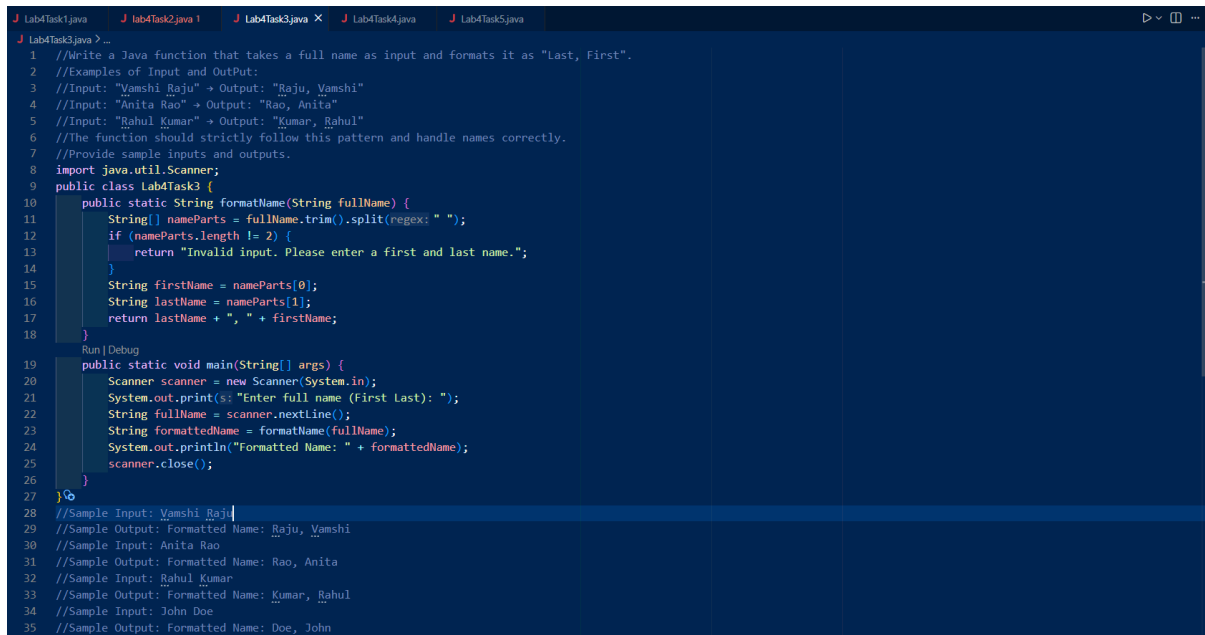
**//Input: "Anita Rao" → Output: "Rao, Anita"**

**//Input: "Rahul Kumar" → Output: "Kumar, Rahul"**

**//The function should strictly follow this pattern and handle names correctly.**

**//Provide sample inputs and outputs.**

## **CODE:**



```
1 //Write a Java function that takes a full name as input and formats it as "Last, First".
2 //Examples of Input and OutPut:
3 //Input: "Vamshi Raju" → Output: "Raju, Vamshi"
4 //Input: "Anita Rao" → Output: "Rao, Anita"
5 //Input: "Rahul Kumar" → Output: "Kumar, Rahul"
6 //The function should strictly follow this pattern and handle names correctly.
7 //Provide sample inputs and outputs.
8 import java.util.Scanner;
9 public class Lab4Task3 {
10     public static String formatName(String fullName) {
11         String[] nameParts = fullName.trim().split(regex: " ");
12         if (nameParts.length != 2) {
13             return "Invalid Input. Please enter a first and last name.";
14         }
15         String firstName = nameParts[0];
16         String lastName = nameParts[1];
17         return lastName + ", " + firstName;
18     }
19     public static void main(String[] args) {
20         Scanner scanner = new Scanner(System.in);
21         System.out.print(s: "Enter full name (First Last): ");
22         String fullName = scanner.nextLine();
23         String formattedName = formatName(fullName);
24         System.out.println("Formatted Name: " + formattedName);
25         scanner.close();
26     }
27 }
28 //Sample Input: Vamshi Raju
29 //Sample Output: Formatted Name: Raju, Vamshi
30 //Sample Input: Anita Rao
31 //Sample Output: Formatted Name: Rao, Anita
32 //Sample Input: Rahul Kumar
33 //Sample Output: Formatted Name: Kumar, Rahul
34 //Sample Input: John Doe
35 //Sample Output: Formatted Name: Doe, John
```

## **Sample Input and Output:**

**//Sample Input: Vamshi Raju**

**//Sample Output: Formatted Name: Raju, Vamshi**

**//Sample Input: Anita Rao**

**//Sample Output: Formatted Name: Rao, Anita**

**//Sample Input: Rahul Kumar**

**//Sample Output: Formatted Name: Kumar, Rahul**

**//Sample Input: John Doe**

**//Sample Output: Formatted Name: Doe, John**

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

### **Scenario:**

Different prompt strategies may produce different code quality.

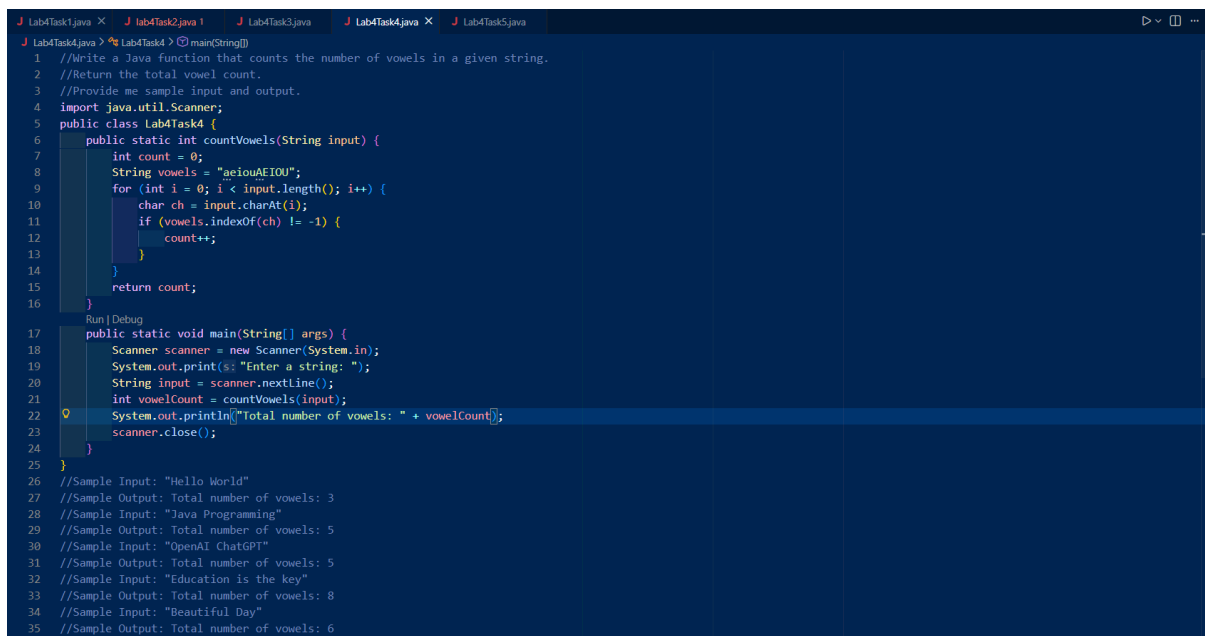
### **Prompt Used:**

**//Write a Java function that counts the number of vowels in a given string.**

**//Return the total vowel count.**

**//Provide me sample input and output.**

### **CODE:**



```
1 //Write a Java function that counts the number of vowels in a given string.
2 //Return the total vowel count.
3 //Provide me sample input and output.
4 import java.util.Scanner;
5 public class Lab4Task4 {
6     public static int countVowels(String input) {
7         int count = 0;
8         String vowels = "aeiouAEIOU";
9         for (int i = 0; i < input.length(); i++) {
10             char ch = input.charAt(i);
11             if (vowels.indexOf(ch) != -1) {
12                 count++;
13             }
14         }
15         return count;
16     }
17     public static void main(String[] args) {
18         Scanner scanner = new Scanner(System.in);
19         System.out.print("Enter a string: ");
20         String input = scanner.nextLine();
21         int vowelCount = countVowels(input);
22         System.out.println("Total number of vowels: " + vowelCount);
23         scanner.close();
24     }
25 }
26 //Sample Input: "Hello World"
27 //Sample Output: Total number of vowels: 3
28 //Sample Input: "Java Programming"
29 //Sample Output: Total number of vowels: 5
30 //Sample Input: "OpenAI ChatGPT"
31 //Sample Output: Total number of vowels: 5
32 //Sample Input: "Education is the key"
33 //Sample Output: Total number of vowels: 8
34 //Sample Input: "Beautiful Day"
35 //Sample Output: Total number of vowels: 6
```

### **Sample Input and Output:**

**//Sample Input: "Hello World"**

**//Sample Output: Total number of vowels: 3**

**//Sample Input: "Java Programming"**

**//Sample Output: Total number of vowels: 5**

**//Sample Input: "OpenAI ChatGPT"**

**//Sample Output: Total number of vowels: 5**

**//Sample Input: "Education is the key"**

**//Sample Output: Total number of vowels: 8**

**//Sample Input: "Beautiful Day"**

**//Sample Output: Total number of vowels: 6**

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

### **Scenario:**

**File processing requires clear logical understanding.**

### **Prompt Used:**

**//Write a java function that reads a .txt file and counts the number of lines in the file.**

**//Example:**

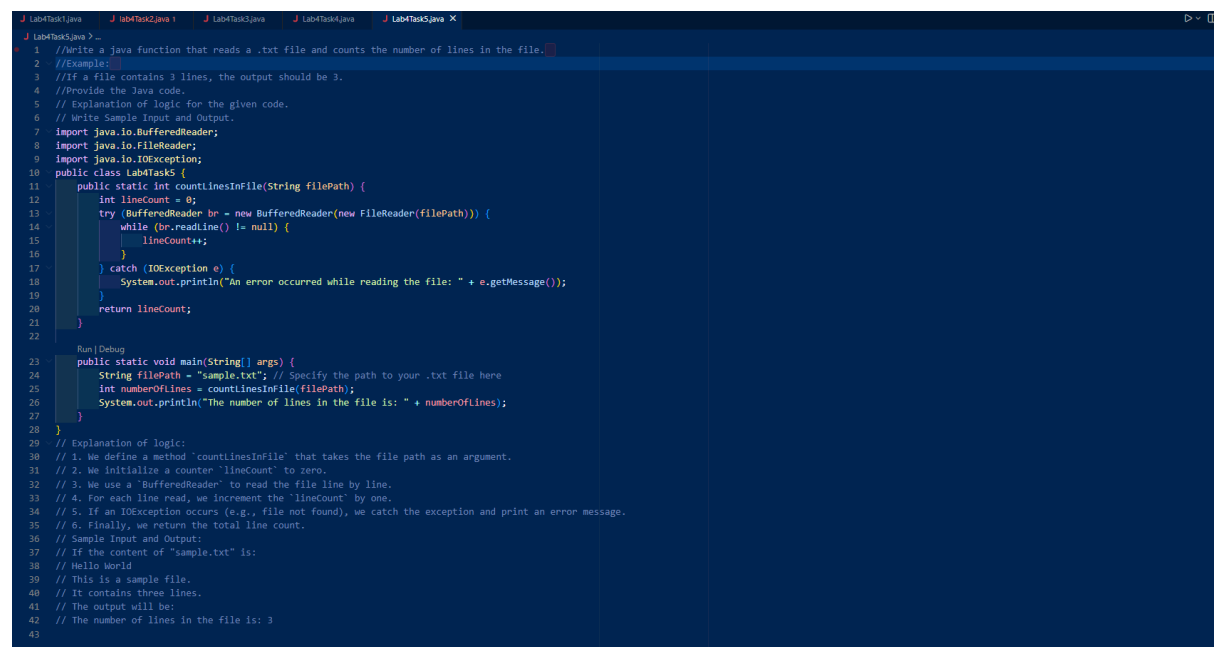
**//If a file contains 3 lines, the output should be 3.**

**//Provide the Java code.**

**// Explanation of logic for the given code.**

**// Write Sample Input and Output.**

### **CODE:**

A screenshot of an IDE window titled 'Lab4Task5.java'. The code is a Java program that defines a static method 'countLinesInFile' to count the number of lines in a text file. The code includes imports for 'BufferedReader', 'FileReader', and 'IOException'. The 'countLinesInFile' method uses a 'BufferedReader' to read the file line by line, incrementing a 'lineCount' variable for each line. It includes a try-catch block to handle 'IOException'. The 'main' method calls 'countLinesInFile' with the path 'sample.txt' and prints the result. Comments explain the logic and provide sample input/output.

```
1 //Write a java function that reads a .txt file and counts the number of lines in the file.
2 //Example:
3 //If a file contains 3 lines, the output should be 3.
4 //Provide the Java code.
5 // Explanation of logic for the given code.
6 // Write Sample Input and Output.
7 import java.io.BufferedReader;
8 import java.io.FileReader;
9 import java.io.IOException;
10 public class Lab4Task5 {
11     public static int countLinesInFile(String filePath) {
12         int lineCount = 0;
13         try (BufferedReader br = new BufferedReader(new FileReader(filePath))) {
14             while (br.readLine() != null) {
15                 lineCount++;
16             }
17         } catch (IOException e) {
18             System.out.println("An error occurred while reading the file: " + e.getMessage());
19         }
20         return lineCount;
21     }
22 }
23
24 Run | Debug
25 public static void main(String[] args) {
26     String filePath = "sample.txt"; // Specify the path to your .txt file here
27     int numberOfLines = countLinesInFile(filePath);
28     System.out.println("The number of lines in the file is: " + numberOfLines);
29 }
30
31 // Explanation of logic:
32 // 1. We define a method 'countLinesInFile' that takes the file path as an argument.
33 // 2. We initialize a counter 'lineCount' to zero.
34 // 3. We use a 'BufferedReader' to read the file line by line.
35 // 4. For each line read, we increment the 'lineCount' by one.
36 // 5. If an IOException occurs (e.g., file not found), we catch the exception and print an error message.
37 // 6. Finally, we return the total line count.
38 // Sample Input and Output:
39 // If the content of "sample.txt" is:
40 // Hello World
41 // This is a sample file.
42 // It contains three lines.
43 // The output will be:
44 // The number of lines in the file is: 3
```

### **Explanation of logic:**

**// 1. We define a method `countLinesInFile` that takes the file path as an argument.**

```
// 2. We initialize a counter `lineCount` to zero.  
// 3. We use a `BufferedReader` to read the file line by line.  
// 4. For each line read, we increment the `lineCount` by one.  
// 5. If an IOException occurs (e.g., file not found), we catch the exception and print an  
error message.  
// 6. Finally, we return the total line count.
```

### **Sample Input and Output:**

```
// If the content of "sample.txt" is:  
  
// Hello World  
  
// This is a sample file.  
  
// It contains three lines.  
  
// The output will be:  
  
// The number of lines in the file is: 3
```