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.

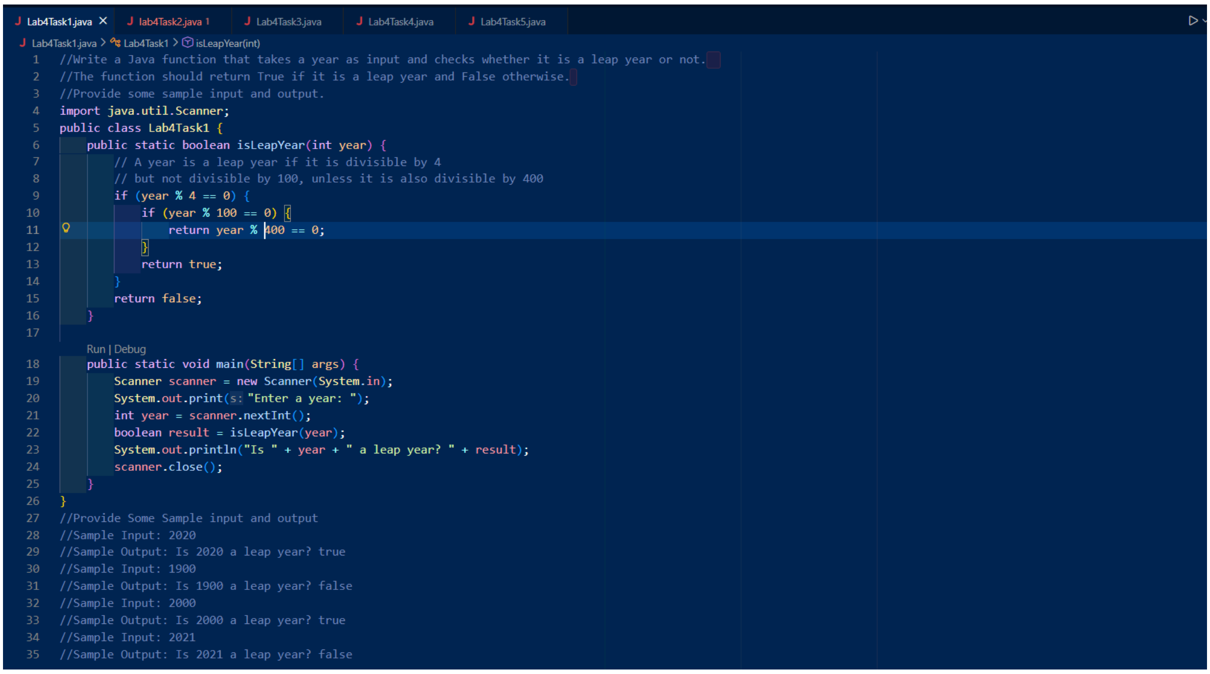
Prompt Used:

//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:



```java
//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.
import java.util.Scanner;
public class Lab4Task1 {
    public static boolean isLeapYear(int year) {
        // A year is a leap year if it is divisible by 4
        // but not divisible by 100, unless it is also divisible by 400
        if (year % 4 == 0) {
            if (year % 100 == 0) {
                return year % 400 == 0;
            }
            return true;
        }
        return false;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s: "Enter a year: ");
        int year = scanner.nextInt();
        boolean result = isLeapYear(year);
        System.out.println("Is " + year + " a leap year? " + result);
        scanner.close();
    }
}
//Provide Some Sample input and output
//Sample Input: 2020
//Sample Output: Is 2020 a leap year? true
//Sample Input: 1900
//Sample Output: Is 1900 a leap year? false
//Sample Input: 2000
//Sample Output: Is 2000 a leap year? true
//Sample Input: 2021
//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.

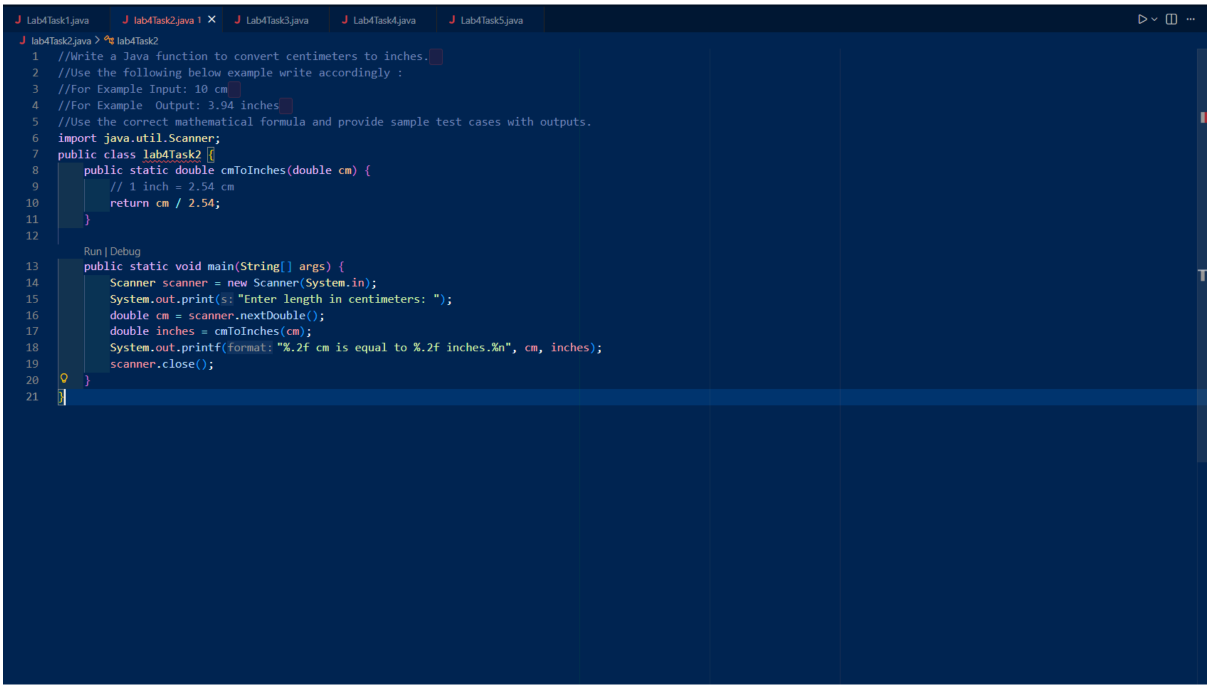
//Use the following below example write accordingly :

//For Example Input: 10 cm

//For Example Output: 3.94 inches

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

CODE:



```java
//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.
import java.util.Scanner;
public class lab4Task2 {
    public static double cmToInches(double cm) {
        // 1 inch = 2.54 cm
        return cm / 2.54;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s: "Enter length in centimeters: ");
        double cm = scanner.nextDouble();
        double inches = cmToInches(cm);
        System.out.printf(format: "%.2f cm is equal to %.2f inches.%n", cm, inches);
        scanner.close();
    }
}
```

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"

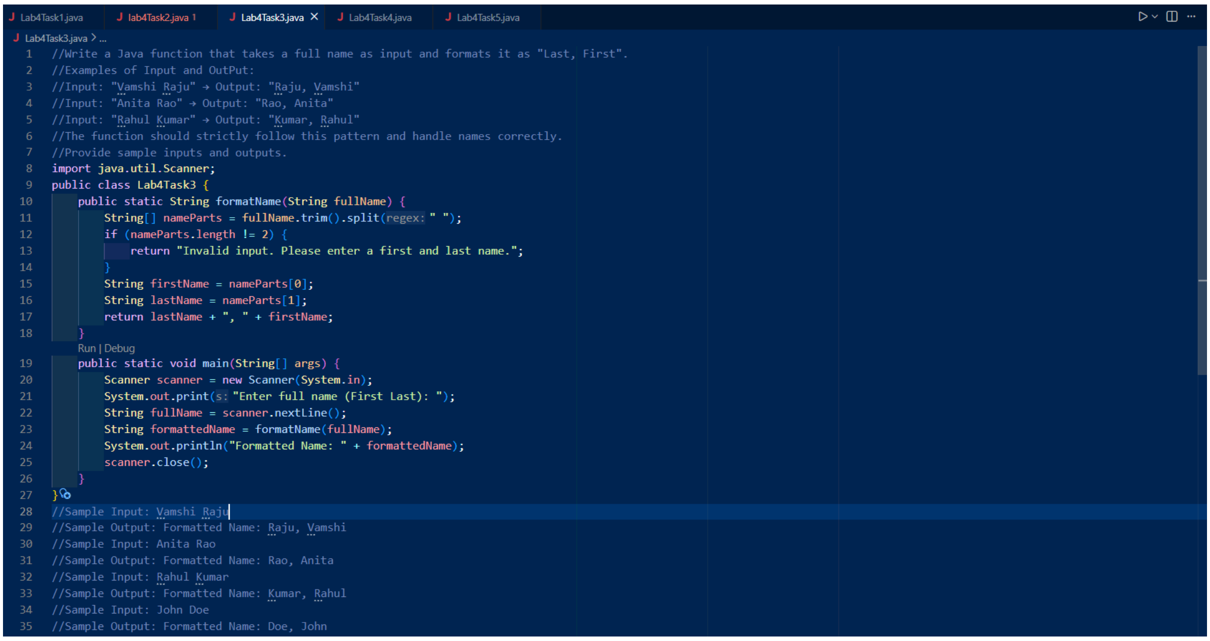
//Input: "Anita Rao" → Output: "Rao, Anita"

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

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

//Provide sample inputs and outputs.

CODE:



```java
//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.
import java.util.Scanner;
public class Lab4Task3 {
    public static String formatName(String fullName) {
        String[] nameParts = fullName.trim().split(regex: " ");
        if (nameParts.length != 2) {
            return "Invalid input. Please enter a first and last name.";
        }
        String firstName = nameParts[0];
        String lastName = nameParts[1];
        return lastName + ", " + firstName;
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s: "Enter full name (First Last): ");
        String fullName = scanner.nextLine();
        String formattedName = formatName(fullName);
        System.out.println("Formatted Name: " + formattedName);
        scanner.close();
    }
}
//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
```

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.

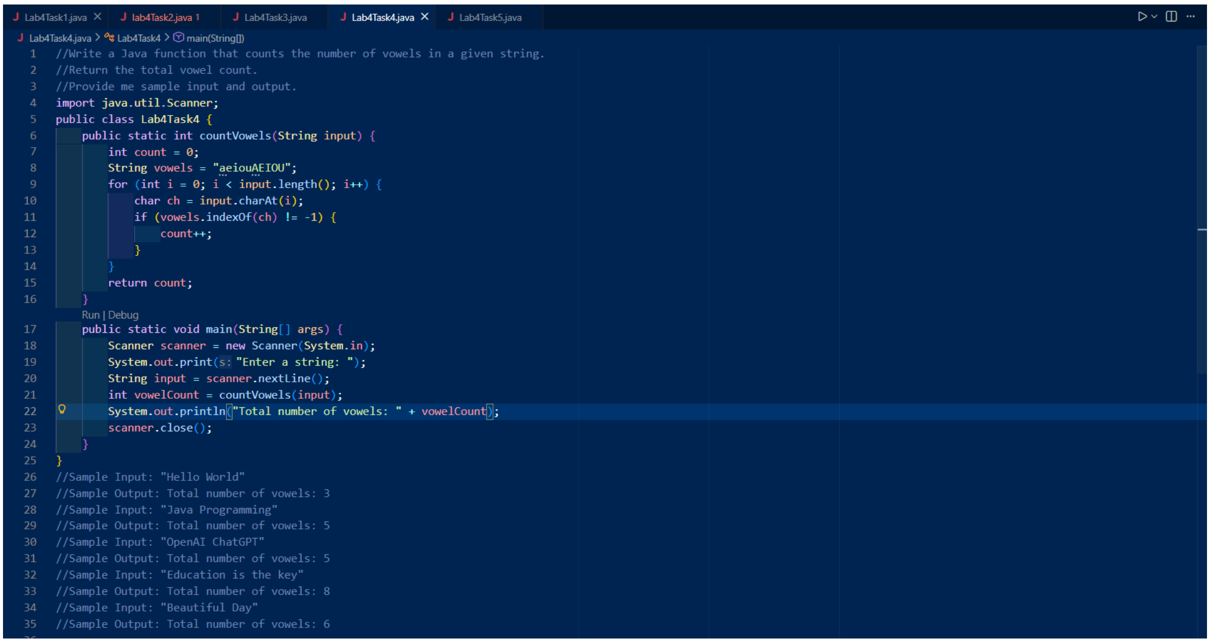
Prompt Used:

//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:



```java
//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.
import java.util.Scanner;
public class Lab4Task4 {
    public static int countVowels(String input) {
        int count = 0;
        String vowels = "aeiouAEIOU";
        for (int i = 0; i < input.length(); i++) {
            char ch = input.charAt(i);
            if (vowels.indexOf(ch) != -1) {
                count++;
            }
        }
        return count;
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print(s: "Enter a string: ");
        String input = scanner.nextLine();
        int vowelCount = countVowels(input);
        System.out.println("Total number of vowels: " + vowelCount);
        scanner.close();
    }
}
//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
```

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:

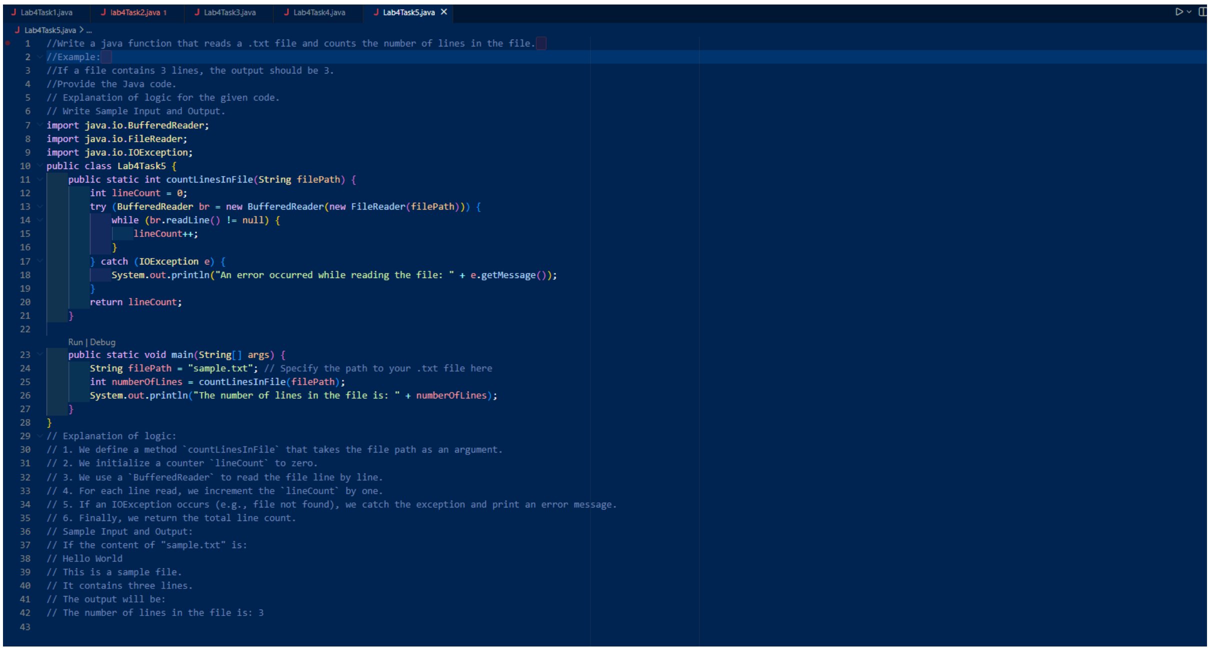
//If a file contains 3 lines, the output should be 3.

//Provide the Java code.

// Explanation of logic for the given code.

// Write Sample Input and Output.

CODE:



```java
//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.
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
public class Lab4Task5 {
    public static int countLinesInFile(String filePath) {
        int lineCount = 0;
        try (BufferedReader br = new BufferedReader(new FileReader(filePath))) {
            while (br.readLine() != null) {
                lineCount++;
            }
        } catch (IOException e) {
            System.out.println("An error occurred while reading the file: " + e.getMessage());
        }
        return lineCount;
    }

    public static void main(String[] args) {
        String filePath = "sample.txt"; // Specify the path to your .txt file here
        int numberOfLines = countLinesInFile(filePath);
        System.out.println("The number of lines in the file is: " + numberOfLines);
    }
}
// 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

```

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