ASSIGNMENT-4.3

2303a51595
B-10

TASK-1

Prompt :

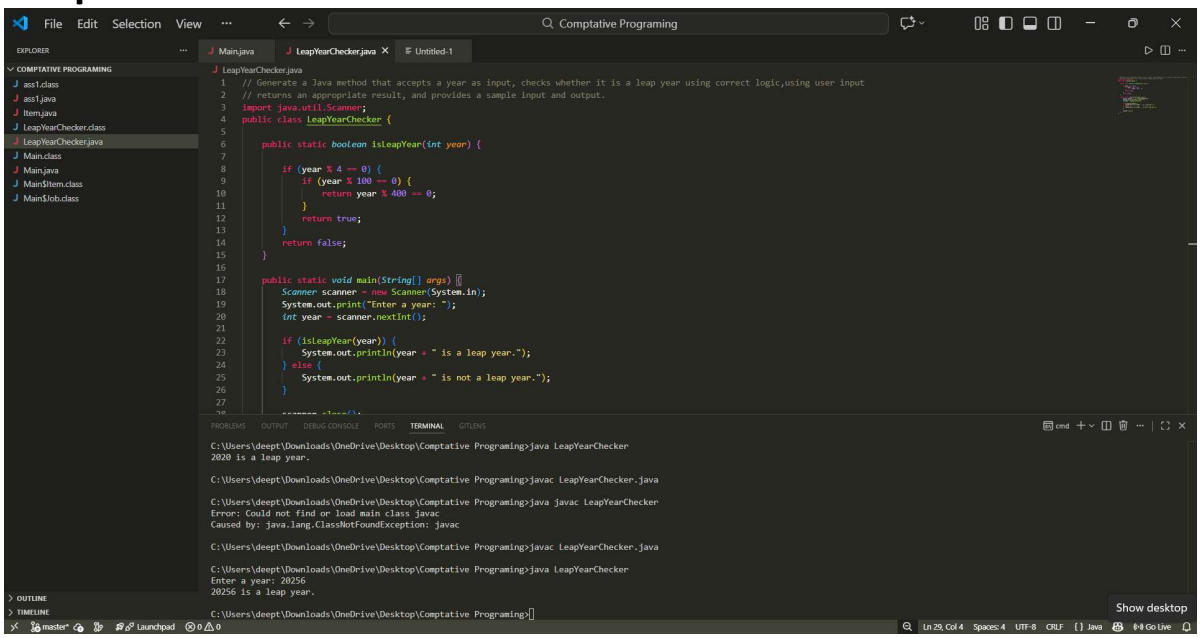
Write a Java program that takes an integer from the user and calculates its factorial using a loop. The program should display the final factorial value clearly.

Code :

```
import java.util.Scanner;
public class LeapYearChecker {

    public static boolean isLeapYear(int year) {

        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("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();

Output :



Analysis :

In this task, leap year conditions are checked using standard rules. The program correctly identifies a leap year based on divisibility by 4, 100, and 400.

TASK-2

Prompt :

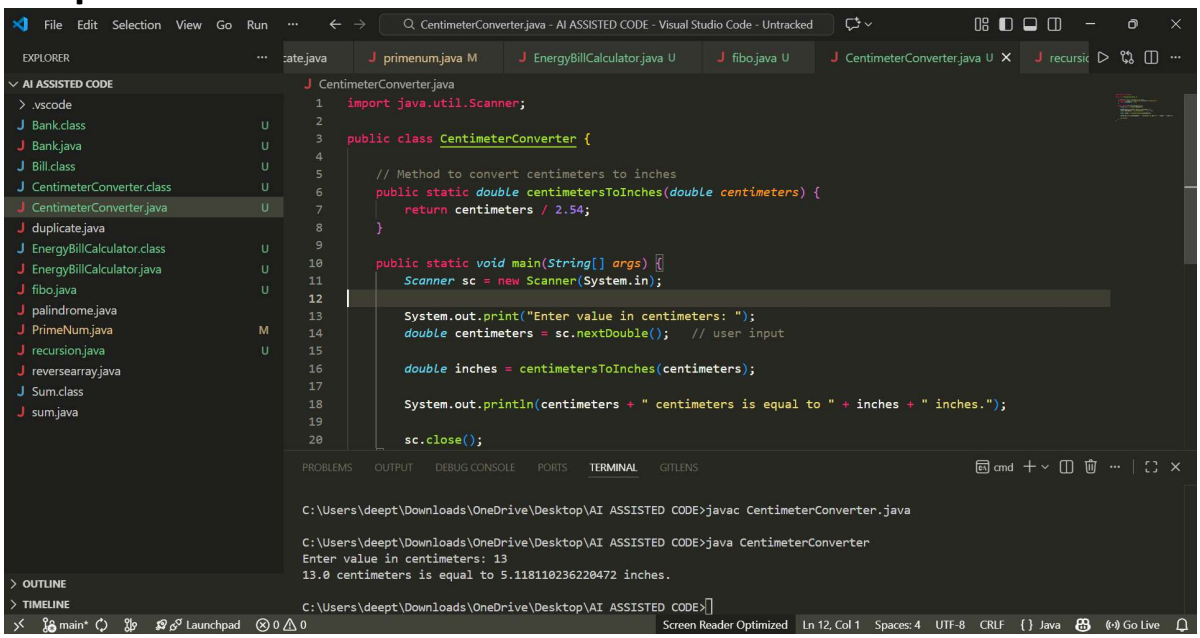
Write a Java program to convert centimeters to inches using user input.
Example: If the input is 10 cm, the output should be 3.94 inches.
Use the correct conversion formula and display the result properly.

Code :

```
import java.util.Scanner;
```

```
public class CentimeterConverter {

   // Method to convert centimeters to inches
   public static double centimetersToInches(double centimeters) {
      return centimeters / 2.54;
   }
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);

      System.out.print("Enter value in centimeters: ");
      double centimeters = sc.nextDouble();   // user input

      double inches = centimetersToInches(centimeters);

      System.out.println(centimeters + " centimeters is equal to " + inches + "
inches.");

      sc.close();
   }
}
```

Output :



Analysis :

The program converts centimeters to inches using the correct mathematical formula. The output values are accurate for different inputs.

TASK-3

Prompt :

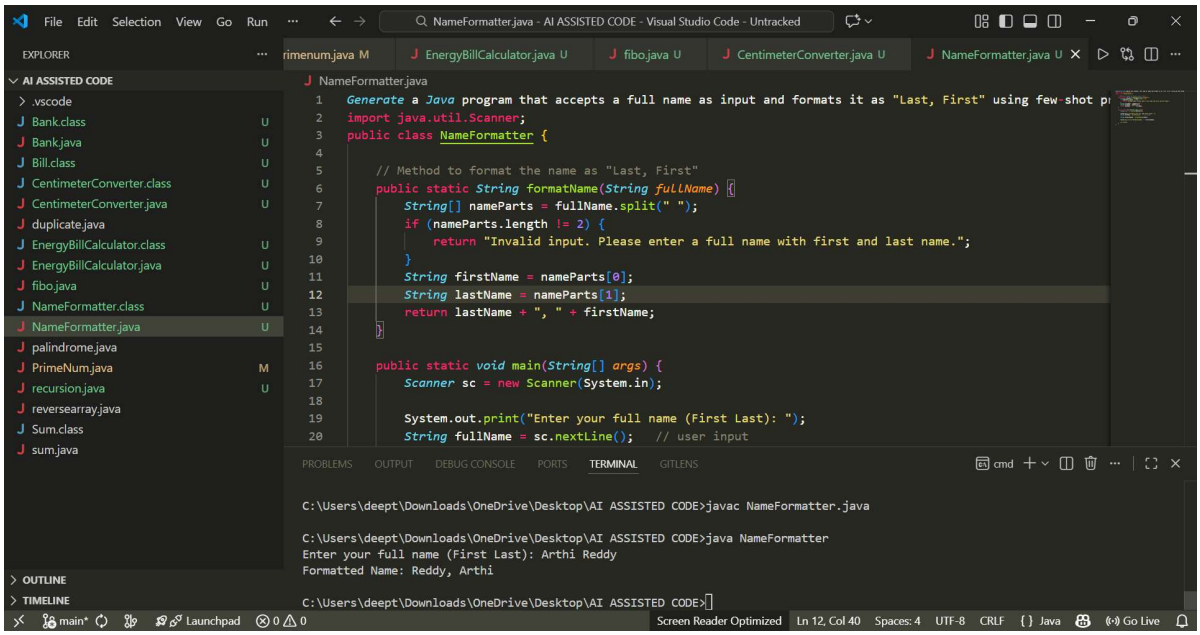
Write a Java program that accepts a full name from the user and formats it as “Last, First”.

Code :

```
import java.util.Scanner;
public class NameFormatter {

   // Method to format the name as "Last, First"
   public static String formatName(String fullName) {
      String[] nameParts = fullName.split(" ");
      if (nameParts.length != 2) {
         return "Invalid input. Please enter a full name with first and last
name.";
      }
      String firstName = nameParts[0];
      String lastName = nameParts[1];
      return lastName + ", " + firstName;
   }

   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);

      System.out.print("Enter your full name (First Last): ");
      String fullName = sc.nextLine();   // user input

      String formattedName = formatName(fullName);
```

```
        System.out.println("Formatted Name: " + formattedName);

        sc.close();
    }
}
```

Output :



Analysis :

In this task, multiple examples are used to understand the required output format. The program takes a full name as input, separates the first name and last name, and rearranges them into “Last, First” format. By following the given examples, the output is consistent and correctly formatted for all valid inputs.

TASK-4

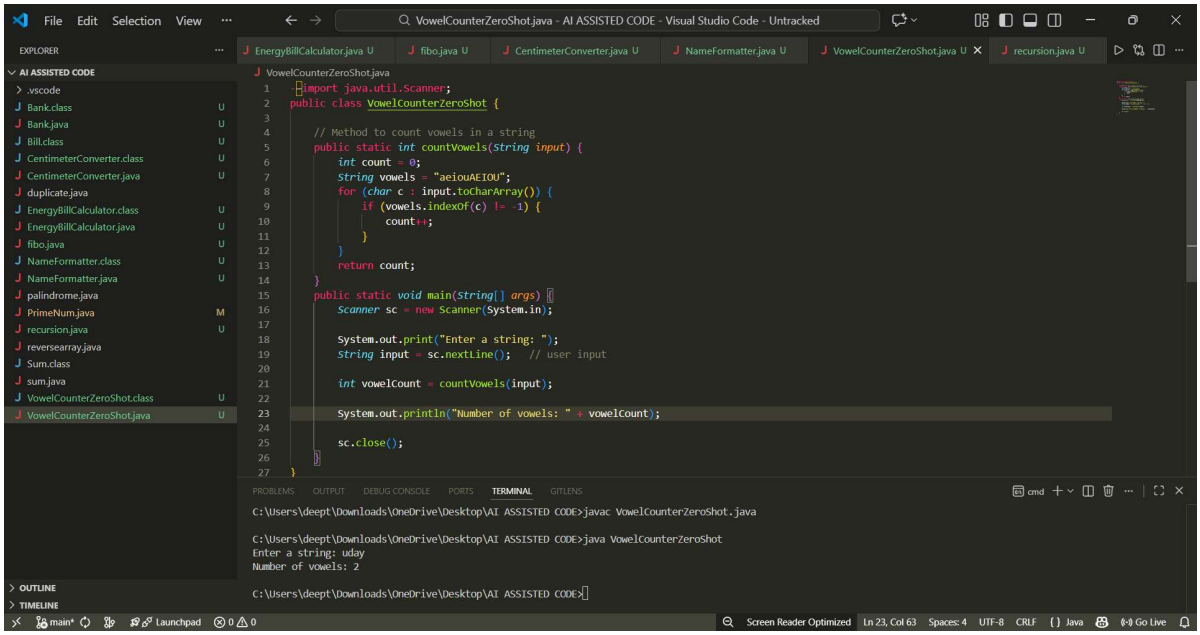
Prompt :

Write a Java program that takes a string as input and counts the number of vowels present in it.

CODE :

```
-–import java.util.Scanner;
public class VowelCounterZeroShot {

   // Method to count vowels in a string
   public static int countVowels(String input) {
      int count = 0;
      String vowels = "aeiouAEIOU";
      for (char c : input.toCharArray()) {
         if (vowels.indexOf(c) != -1) {
            count++;
         }
      }
      return count;
   }
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);

      System.out.print("Enter a string: ");
      String input = sc.nextLine();   // user input

      int vowelCount = countVowels(input);

      System.out.println("Number of vowels: " + vowelCount);

      sc.close();
   }
}
```

Output :



Analysis :

The vowel-counting program written using examples shows clearer logic and is easier to read. Providing examples helps in understanding the expected behavior and improves the overall quality of the code.

TASK-5

Prompt :

Write a Java program that reads a .txt file and counts the total number of lines in it.

Example:

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

The program should take the file path as input and display the line count.

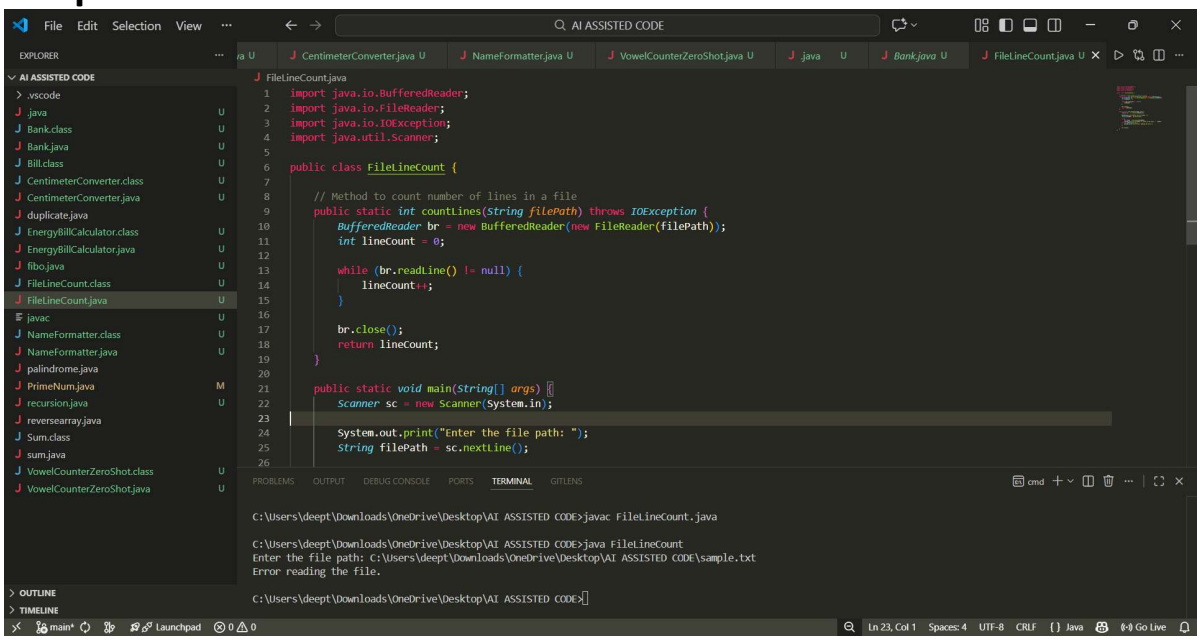
Code :

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.io.IOException;
import java.util.Scanner;
```

```
public class FileLineCount {

   // Method to count number of lines in a file
   public static int countLines(String filePath) throws IOException
{
      BufferedReader br = new BufferedReader(new
FileReader(filePath));
      int lineCount = 0;

      while (br.readLine() != null) {
         lineCount++;
      }

      br.close();
      return lineCount;
   }

   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);

      System.out.print("Enter the file path: ");
      String filePath = sc.nextLine();

      try {
         int lines = countLines(filePath);
         System.out.println("Number of lines in the file: " + lines);
      } catch (IOException e) {
         System.out.println("Error reading the file.");
      }
```

```
        sc.close();
    }
}
```

Output :



Analysis :

This program works by taking the file path from the user, reading the file line by line, and increasing a counter for each line, which finally gives the total number of lines in the file while handling errors safely.