Exception handling

You are required to write a Java program that reads an integer from the user and calculates its factorial. Your program should handle possible exceptions gracefully, providing appropriate error messages and ensuring the program does not terminate unexpectedly due to invalid input or other exceptions.

Requirements:

Input Handling:

Prompt the user to enter an integer.

Handle input exceptions such as NumberFormatException if the user enters non-numeric values.

Factorial Calculation:

Calculate the factorial of the entered integer.

Handle exceptions related to factorial calculation, such as ArithmeticException if the number is negative (factorial of negative numbers is undefined).

import java.util.Scanner;

public class FactorialCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter an integer to calculate its factorial: ");

String input = scanner.nextLine();

try {

int number = Integer.parseInt(input);

if (number < 0) {

throw new ArithmeticException("Factorial of negative numbers is undefined.");

}

long factorial = calculateFactorial(number);

System.out.println(number + "! = " + factorial);

} catch (NumberFormatException e) {

System.out.println("Input must be a valid integer.");

} catch (ArithmeticException e) {

System.out.println(e.getMessage());

}

}

private static long calculateFactorial(int n) {

if (n == 0 || n == 1) {

return 1;

}

long factorial = 1;

for (int i = 2; i <= n; i++) {

factorial \*= i;

}

return factorial;

}

}