Summary For Remote Coding Test (Calculator)

*Time taken to complete: 8 hours*

The Calculator solution consists of 3 projects:

Graphical user interface, text, application

Description automatically generated

1. Calculator Console Application (CalculatorConsole)
2. Calculator Class Library (CalculatorService)
3. Calculator Test Project (CalculatorService.Test)

## Calculator Console Application

It acts as a simple user interface to collect user input (mathematical expression) and return the correct result of the mathematical expression. It applies **Calculate method** from **Calculator Class Library** to calculate the mathematical expression.

## Calculator Class Library

It helps to solve the mathematical expression and return the correct result. It’s applying the logic of multiplication and division are granted a higher precedence than addition and subtraction, and the part of the expression within the parenthesis should be evaluated first. It consists of two classes which are **Calculator** and **Operation** class.

Graphical user interface, text

Description automatically generated

**Calculator Class** consists of two functions:

|  |
| --- |
| public static double Calculate(string sum) {   1. Split the mathematical expression using space separator and store it in an array. 2. Loop through the array and insert the numbers and operatorsinto two different stacks. While looping through the array:    1. When there is closed parenthesis (“)”), it will calculate the subtotal within the parenthesis first and insert the subtotal into the numbers stack.    2. When there are times (“\*”) or divide (“/”), it will perform the multiplication and division first before inserting into the numbers stack. 3. In the end, it will call Subtotal function to calculate the total based on the numbers stack and operators stacks.   } |

*Subtotal function*is created to calculate the subtotal inside the parenthesis and the total of the remainders in the operators stack and numbers stack.

|  |
| --- |
| private static string Subtotal(Stack<string> operators, Stack<String> numbers) {   1. Reverse the sequence of elements in the operators and numbers stacks. 2. Calculate the total based on the elements in the operators and numbers stacks.   } |

**Operation Class** provides functionalities of mathematical operators such as Add, Minus, Times, and Divides. It provides the basic operation of a mathematical expression ([Left number] [Operator] [Right number]). E.g. (1 + 1).

## Calculator Test Project

This test project is created to prove that **Calculate** method manages to solve the provided mathematical expressions.