

Arithmetic Expression Evaluator

User's Manual

Version 1.0

\

Arithmetic Expression Evaluator	Version: 1.0
Software Architecture Document	Date: 12/6/2024
Group 4	

Revision History

Date	Version	Description	Author
9/25/2024	0.0	Initial meeting, discussion of meeting times, and assigning roles for the project	Alexander Carrillo
10/16/2024	1.0	Software requirements discussion, reviewing rough code, talking about next steps	Alexander Carrillo
11/4/2024	1.0	Software architecture design discussion and writing of document	Alexander Carrillo
12/2/2024	1.0	Use Case and Implementation Discussion, writing of document 05 use cases	Alexander Carrillo
12/6/2024	1.0	Code and Users Manual discussion	Alexander Carrillo

Arithmetic Expression Evaluator	Version: 1.0
Software Architecture Document	Date: 12/6/2024
Group 4	

Table of Contents

1.	Purpose	4
2.	Introduction	4
3.	Getting started	4
4.	Advanced features	4
5.	Troubleshooting	4
6.	Example of uses	4
7.	Glossary	5
8.	FAQ	5

Arithmetic Expression Evaluator	Version: 1.0
Software Architecture Document	Date: 12/6/2024
Group 4	

Test Case

1. Purpose

This document is meant to provide an explanation on how to use the Arithmetic Expression Evaluator effectively to assist in the evaluation of multiple different kinds of operations

2. Introduction

The Arithmetic Expression Evaluator is meant to assist in the solving and evaluation of Arithmetic problems. It has a selection of operators such as addition (+), subtraction (-), multiplication(*), division (/), modulus (%), and exponents (**). The program uses the order of operations PEMDAS, parentheses, exponents, multiplication/division, and addition/subtraction. Because of this it also is able to give priority to what is in the parentheses first and then expand out from there.

3. Getting started

To use the calculator, input the expression into the top box and then hit the Enter button at the bottom of the program. Then the Results box below the input will return the solved expression. If an error is encountered it will be shown in the Results box ie: "The given equation was invalid, please try again."

4. Advanced features

Our program features an interactive GUI that shows as an external window. The window will display an input and output box, along with an enter button.

5. Troubleshooting

- First check to see if your logic is correct
- Make sure your program isn't dividing or modulating by zero at any point in the expression
- Check to see if you have any unmatched parentheses
- Make sure that the symbols in the program are valid i.e. no '.', '[]', '{ }', ect.

6. Examples

Addition

- input: 6+9
- output: 15

Subtraction

- input: 7-3
- output: 4

multiplication

- input: 4*3
- output: 12

division

- input: 16/8
- output: 2

exponent

Arithmetic Expression Evaluator	Version: 1.0
Software Architecture Document	Date: 12/6/2024
Group 4	

- input: $5*2$
- output: 25

modulus

- input: $9\%4$
- output: 1

PEMDAS

- input: $10/(8-3)$
- output: 2

7. Glossary of terms

PEMDAS: order of operations, parentheses, exponents, multiplication/division, and addition/subtraction

Operators: the symbols that are used to tell the program what operation to do to the current two numbers

Modulus (%): An operator that isn't commonly found on calculators, it returns the remainder of the division, ex. $5\%2=1$

GUI: Graphic user interface, the face of the program, what the user interacts with to use the program

Error Message: A prompt that is given to the user when they input an invalid sequence that results in a string that states: "The given equation was invalid, please try again."

8. FAQ

- Can the system store previously entered expressions and their results?
 - No, the system does not have an internal save feature.
- Can the program handle square roots
 - while there is no operator for square roots you can do an exponent be less than one as a decimal, and you can get the same answers
- What order of operations does the calculator use?
 - The calculator used PEMDAS for its order of operations
- What does the program do if you try to divide by zero?
 - The program will notice if there is a zero to the left of any division operator (/), but the division operator function also checks to see each time it is called if the denominator is a zero or not then will return an error
- What happens if the calculator is given an invalid input?
 - The program will return an error message to the output box, then the user is able to try a different input