Jayhawks

[1]Arithmetic Expression Evaluator

User's Manual

Version 1.0

[1]

Revision History

Date	Version	Description	Author
29/11/2023	0.1	The initial user manual	Alexandra, Deborah, Riley, Timo, Victor, Ellia
03/12/2023	1.0	The finished user manual	Alexandra, Deborah, Riley, Timo, Victor, Ellia

Table of Contents

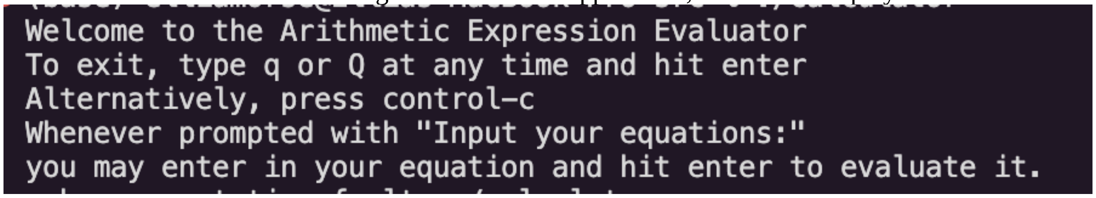
1. Purpose

The intention of this document is to provide an easy-to-understand guide on how to use the Arithmetic Expression Evaluator.

2. Introduction

The Arithmetic Expression Evaluator enables users to calculate expressions involving numerical constants. The software supports the following operations: addition, subtraction, multiplication, division, exponentiation, and modulo.

To install the program, download the necessary files from GitHub. Using your terminal, navigate to the downloaded file, unzip it, go into the unzipped folder, and then the “Src” directory. Then execute the “make” command. Finally, execute “./calculator”. This will begin the calculator application, which will display further instructions.



```
Welcome to the Arithmetic Expression Evaluator
To exit, type q or Q at any time and hit enter
Alternatively, press control-c
Whenever prompted with "Input your equations:"
you may enter in your equation and hit enter to evaluate it.
```

3. Getting started

3.1 Basic Operators

3.1.1 Addition

To perform addition, enter the numbers with a ‘+’ sign between them. For instance, to add 6 and 7, type ‘6+7’ into the terminal and press the Enter key. The result, 13, will be displayed in the terminal. Addition supports both whole numbers and decimal numbers.

3.1.2 Subtraction

To perform subtraction, enter the numbers with a ‘-‘ sign between them. For instance, to subtract 3 from 9, type ‘9-3’ in the terminal and press the Enter key. The result, 6, will be displayed in the terminal. Subtraction supports both whole numbers and decimal numbers.

3.1.3 Multiplication

To perform multiplication, enter the numbers with a ‘*’ sign between them. For instance, to multiply 2 by 6, type ‘2*6’ in the terminal and press the Enter key. The result, 12, will be displayed in the terminal. Multiplication supports both whole numbers and decimal numbers.

3.1.4 Division

To perform division, enter the numbers with a ‘/’ sign between them. For instance, to divide 9 by 3, type ‘9/3’ in the terminal and press the Enter key. The result, 3, will be displayed in the terminal. It’s important to note that division by zero is not allowed. For instance, attempting ‘5/0’ will result in an error being displayed to the terminal. Division supports both whole numbers and decimal numbers.

3.1.5 Exponent

To perform exponentiation, enter the numbers with a ‘^’ sign between them. For instance, to find 2 to the power of 4, type ‘2^4’ in the terminal and press the Enter key. The result, 16, will be displayed in the terminal. Exponents supports both whole numbers and decimal numbers.

3.1.6 Modulo

To find the remainder after division, enter the number with a '%' sign between them. For instance, to find the remainder when 10 is divided by 3, input '10%3' in the terminal and press the Enter key. The result, 1, will be displayed to the terminal. It's important to note that modulo by zero is not allowed. For instance, attempting '5%0' will result in an error being displayed to the terminal. Modulo supports both whole numbers and decimal numbers.

3.2 Handling

3.2.1 Parenthesis Handling

Parenthesis has the highest precedence. Any operation within them will be processed first. It is crucial to ensure that each opening parenthesis is paired with a corresponding closing parenthesis. For instance, in the expression '(4+3)*2', the opening parenthesis is followed by a closing parenthesis. However, in '2*(4+3-1', there is an opening parenthesis without a corresponding closing parenthesis. This will result in an error.

3.2.2 Error Handling

In the event of an error, the Arithmetic Expression Evaluator provides a clear error message to assist users in identifying issues. Common error scenarios include mismatched parentheses, division by zero, or invalid input characters.

3.3 Quitting the program

3.3.1 Quit with a Capital Q

To exit the program, type a capitalized 'Q' into the terminal and press the Enter key.

3.3.2 Quit with a Lowercase q

For an alternative exit method, type a lowercase 'q' into the terminal and press the Enter key.

3.3.3 Quit with Ctrl-C

Use the keyboard shortcut Ctr-C to promptly exit the program at any time.

3.4 Interpreting the results

3.4.1 Results without errors

Upon successful execution of an expression, the program will display the result as a numerical value in the terminal.

3.4.2 Results with errors

In the event of an error during calculation, the program will provide an error message indicating the nature of the issue. Users should review the error message for details and adjust input accordingly.

4. Advanced features

This program does not have advanced features.

5. Troubleshooting

5.1 Mathematical Errors

5.1.1 Dividing by Zero

Attempting to divide a number by zero will lead to an error. For example, '5/0' is invalid. To correct this, ensure that the divisor is a non-zero number, like '5/2'.

5.1.2 Modulo by Zero

Attempting to perform modulo zero will lead to an error. For example, '5%0' is invalid. To correct, ensure that the divisor in the modulo operation is a non-zero number, like '5%2'.

5.2 Invalid Expressions

5.2.1 Unmatched Parentheses

An unmatched parentheses error occurs when there is an uneven number of opening and closing parentheses. For example, '28(4+3-1' is invalid. To correct, add a closing parenthesis, like '2*(4+3-1)'.

5.2.2 Operators with Operands

Ensure that each operator has the correct number of operands. For example, '3+*2' is invalid. To correct, provide the missing operand, like '3+5*2' or remove the extra operator, like '3+2'.

5.2.3 Missing Operators

Ensure that each operand has the correct number of operators. For example, '3(4+2)' is invalid. To correct, add the missing operator, like '3*(4+2)' or remove the extra operand, like '(4+2)'.

5.2.4 Invalid Characters

Ensure that only valid characters are used. Any character that is not a digit, operator, or parenthesis is considered invalid. For example, '((7*3)@2)' is considered invalid because of the @ character.

6. Examples

Here are some examples of expressions with various operators that this calculator will evaluate.

Addition: 6+7, 5+0, 2.5+2.5, (2+(-3))
Subtraction: 3-2, 2-0, (2-(-3)), (-2)-(-3)
Multiplication: 2*1, 2*(-3), 2*2.5
Division: 1/3, (9/(-3)), 9/3
Exponentiation: 5^2, 2^.5, -2^-2
Modulo: 5%2.3, 5%-2
Combinations: 2^(4+3*1)

7. Glossary of terms

Operators – are symbols or keywords that represent specific operations to be performed on one or more operands. In the context of a calculator or programming language, operators define actions such as addition (+), subtraction (-), multiplication (*), division (/). They dictate how operands should be manipulated to produce a desired result.

Operands – Operands are the values upon which operators act. They are the inputs that undergo the specified operation. In a mathematical expression like '2+3', the numbers '2' and '3' are the operands, and the '+' is the operator. In this calculator, operands are static constants.

Modulo – is represented by the symbol '%', is a mathematical operation that returns the remainder as a result of division.

Terminal – In the context of computing, a terminal refers to a text-based interface that allows users to interact with a computer through a command line. It is a command-line interface (CLI) where users can enter text commands to perform various tasks. The terminal provides a way to navigate the file system, execute programs, and access system functions. It is an essential tool for developers, systems administrators, and users who prefer a text-based approach to interacting with their computers. In this context, the user is expected to be using a terminal within a Unix environment.

8. FAQ

Question: Who was this calculator developed by?

Answer: This app was developed by Alexandra Stratton, Riley Sirimongkhon-Dyck, Timo Aranjo, Victor Maduka, Ellia Morse, Deboarh Onuosa.

Question: What language was this application built in?

Answer: The calculator app was built in C++

Question: Why was this calculator made?

Answer: The calculator was built as the final project for EECS 348 Software Engineering at the University of Kansas

Question: Is there shortcut to retype the text of my last entry?

Answer: The calculator currently does this feature. You will manually have to retype and edit your expression.

Question: How do I exit the program?

Answer: You can exit the program by typing a capitalized ‘Q’, a lowercase ‘q’, or using the keyboard shortcut “Ctrl-C”

Question: Can I use variables in my calculations?

Answer: The calculator currently does not support variables. You can only perform calculations with numerical values.