
<Team 19>

<EECS 348 Term Project User Manual>

User's Manual

Version <1.0>

[Note: The following template is provided for use with the Unified Process for EDUcation. Text enclosed in square brackets and displayed in blue italics (style=InfoBlue) is included to provide guidance to the author and should be deleted before publishing the document. A paragraph entered following this style will automatically be set to normal (style=Body Text).]

[To customize automatic fields (which display a gray background when selected), select File>Properties and replace the Title, Subject and Company fields with the appropriate information for this document. After closing the dialog, automatic fields may be updated throughout the document by selecting Edit>Select All (or Ctrl-A) and pressing F9, or simply click on the field and press F9. This must be done separately for Headers and Footers. Alt-F9 will toggle between displaying the field names and the field contents. See Word help for more information on working with fields.]

<Arithmetic Expression Evaluator>	Version: <1.0>
User's Manual	Date: <05/12/24>
<document identifier>	

Revision History

Date	Version	Description	Author
<05/12/24>	<1.0>	<Created user manual>	<Kyle Fleming>

<Arithmetic Expression Evaluator>	Version: <1.0>
User's Manual	Date: <05/12/24>
<document identifier>	

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>
User's Manual	Date: <05/12/24>
<document identifier>	

Test Case

1. Purpose

This manual will provide a simple guide on using the Arithmetic Expression Evaluator ; a C++ program made to parse/ evaluate arithmetic expressions. It will cover the program's features, usage, and troubleshooting to ensure a smooth and straightforward user experience.

2. Introduction

The Arithmetic Expression Evaluator is a cli tool built to evaluate mathematical expressions that use operators such as: +, -, *, /, %, **. It follows operator precedence (PEMDAS), including support for parentheses.

Features:

- support for basic mathematical operators
- Handling of operator precedence and parentheses
- Clear error messages for invalid input
- Straightforward command-line interface

3. Getting started

1. Installing and running the program (all commands entered in terminal)
 - a. Install the main.cpp file
 - b. Compile with
 - i. `g++ main.cpp -o Evaluator`
 - c. Run the compiled executable
 - i. `./Evaluator`
2. Using the program
 - a. Enter a valid arithmetic expression
 - i. `1 + 2 * (3 / 4)`
 - b. Press "Enter" to evaluate the expression
 - c. Repeat a and b until you want to quit, in which type the following to end program
 - i. `quit`
 - ii. `Quit`
 - d. Supported operators
 - i. `+` (addition)
 - ii. `-` (subtraction)
 - iii. `*` (multiplication)
 - iv. `/` (division)
 - v. `%` (Modulus)
 - vi. `**` (Exponentiation)

4. Advanced features

1. Decimal support
 - a. This program supports evaluating floating point numbers, so input isn't constrained.

5. Troubleshooting

1. Invalid input:
 - a. Make sure that everything is a valid number or operator
2. Unmatched Parentheses:
 - a. Make sure all parentheses are correctly paired
3. Division by Zero:

<Arithmetic Expression Evaluator>	Version: <1.0>
User's Manual	Date: <05/12/24>
<document identifier>	

- a. Make sure nothing is being divided by zero
4. Unsupported Operators:
 - a. Use only operators listed in 2.d

6. Examples

1. Valid Expressions
 - a. $1 + 2$
 - i. Evaluates to 3
 - b. $1 + 2 * (4 / 2)$
 - i. Evaluates to 5
 - c. $2 ** 4$
 - i. Evaluates to 16
2. Invalid Expressions
 - a. $1 + (2$
 - i. Unmatched parentheses
 - b. $1 / 0$
 - i. Division by zero
 - c. $1 \& 2$
 - i. Invalid operator

7. Glossary of terms

- CLI
 - Command-line-interface: the interface in which one interacts with the program
- PEMDAS
 - The order of precedence: parentheses, exponents, multiplication, division, addition, subtraction.

8. FAQ

- What happens if I enter something invalid?
 - The program will display an error message that describes the issue.
- Can I evaluate decimals?
 - The program currently only supports integers.
- How do I end the program?
 - Enter quit or Quit.
- Is there a graphical version?
 - No, there is only a command-line tool at the moment.