**EvalEx**

User’s Manual

Version 1.0

Revision History

| **Date** | **Version** | **Description** | **Author** |
| --- | --- | --- | --- |
| 29/11/23 | 1.0 | Added all of the details for the user manual. | Tommy Lam, Will Whitehead, Joshua Lee, Max Biundo, Joe Hotze, Gunther Leuchtefeld |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

**EvalEx User’s Manual**

# Purpose

The purpose of EvalEx is to provide an easy and accurate calculator on the computer that doesn’t require a WiFi connection. EvalEx provides fast and accurate results to any unary bases calculation needs. Our software handles parenthesis and order of operations correctly so you don’t have to. EvalEx makes crunching numbers and handling operations easy.

# Introduction

EvalEx is a basic calculator software, able to take in a wide range of inputs and accurately display outputs. EvalEx boasts many features, including handling of extraneous parentheses, unary operators, arithmetic operations, addition, subtraction, multiplication, division, and modulo. Installation is simple, as the file folder includes the source code as well as an executable.

# Getting started

EvalEx is a simple to use system that can be used on an operating system capable of running C++. In order to use the program it must first be started, at which point the user will be prompted with a small menu.

In order to use the calculator features of the program, simply type in an expression, as you would into a normal calculator. Currently, many operations are supported, those being:  
**+** (Addition) **-** (Subtraction) **\*** (Multiplication)

**/** (Division) **%** (Modulo) **^** (Exponentiation)

**()** (Parenthesis) **+/-** (Positive or Negative)

There are a few caveats to using the calculator correctly, though. Most importantly of all, **Space characters are relevant to the functions of the calculator.** Spaces matter in order to distinguish between attaching a positive or negative operator to an integer, and addition and subtraction. Negative and positive operators should be attached directly to an integer or set of parentheses, while spaces should be placed between expressions. Parentheses should also directly be connected to the integers within them. For example:

8 - (5 / -2) is correct form.

10 + - 2 is incorrect form and will result in an error.

Additionally, parentheses multiplication, such as in the case of 4(5 + 3), is not supported. Please utilize the multiplication operator (\*) for situations such as these. For example: 4 \* (5 + 3) would work and would not raise errors. Further examples of expressions that can be entered into the calculator can be found in the Examples section below.

# Advanced features

EvalEx has been extensively tested to ensure accuracy and precision. Advanced features have been included that push EvalEx to be a better calculation tool than many others on the market. Its state of the art handling of extraneous parentheses as well as its inclusion of the modulo operator make EvalEx unique and bring it leagues ahead of the competition. Handling of unary operators has also been included.

# Troubleshooting

Most problems within EvalEx have very simple solutions. First, check that your input for unmatched parentheses, letters or special characters that EvalEx cannot handle, etc. Consistent errors not regarding syntactical errors should be fixed by deleting the executable and recompiling the source code into a new executable. To do this, the first step is to open the file that EvalEx is in, in a new terminal window. Next, input this command:

$\g++ .\EvalEx.cpp -o .\evalex.exe

This will recompile the code and create a new executable that is free from memory leaks and may work better.

# Examples

Example 1: Simple Operators

46

Example 2: Utilizing Parentheses

120

Example 3: Unary Operators

(-5) + (-4)

-9

Example 4: Modulo

15 % 4

3

\**Modulo is a type of division that only outputs the remainder.*

Example 5: Extraneous Parentheses

(((6 \* (4 + 1)))) + ((5 + 3))

38

# Glossary of terms

# Syntax - a rule that dictates the combinations of symbols and words that are considered to be correctly structured programs or expressions in a programming language.

Compile - the process of translating source code written in a programming language into an executable

Executable - a file that can be activated to run the source code and provide a way for a user to input and receive output

Extraneous - not needed; extra

Unary Operator - work on a single operand

# FAQ

*Q: Can I customize the appearance of the calculator?  
A:* Currently there is no way to customize the appearance of the calculator, but this is a planned future.

*Q: Is there a way to handle errors or invalid input?*

*A:* Please refer to the documentation on error handling or review your syntax.

*Q: “Is there a way to input factorials into the calculator?”*

*A:* Currently there is no way to input the sign for factorials, but EvalEx is built to handle large operations. Writing out a factorial like “5 \* 4 \* 3 \* 2 \* 1”, will still give the same output as inputting “5!”.

*Q: “Can EvalEx do calculus?”*

*A: Not currently. It isn’t built to handle derivatives and integrals as of now, but it may have that feature in the future.*

*Q: “Can you set variables in EvalEx?”*

*A:* At the moment, variables are not something you can do in EvalEx, but any other function that does not have variables will work.