Git Goblins

Calc-U-Later
User's Manual

Version <1.0>

Revision History

Date	Version	Description	Author
11/22/2023	1.0	Created a User Manual for Calc-U-Later	Sabeen, James, Brett, Daniel, Kaden, Anna

Table of Contents

User Manual

1. Purpose

This document is to help users get familiar and, if needed, troubleshoot with Calc-U-Later. It provides information regarding running the program, features, troubleshooting, examples, glossary of terms, and FAQ.

2. Introduction

Calc-U-Later is a program designed to perform arithmetic calculations given by the user. This software allows you to evaluate expressions using various operators and functions, such as addition, subtraction, multiplication, division, exponentiation, modulus, and mixed operators. To run the code, choose one of the following sections based on your operating system (O.S).

2.1 Mac

Open command prompt. Find the folder that holds the program and enter that directory. Type the following prompt:

./calculator

2.2 Linux

Open terminal. Find the folder that holds the program and enter that directory. Type the following prompt:

./calculator

2.2 Windows

Open terminal. Find the folder that holds the program and enter that directory. Type the following prompt:

./windows_calculator.exe

Your program should now be running and ready for you to input expressions.

3. Getting started

Step 1: Open the software by running the compiled version, or, if a given version doesn't work for your operating system, compile it using g++ version 13

- Open your preferred IDE or terminal
- Navigate to the directory with the software.
- Run the program using the executable that matches your O.S.
    - If no executable is found for your O.S. or the given executable isn't working on your specific O.S., look at 5.1

Step 2: Input an Expression

- Look for an input area or command line where you can enter your arithmetic expression.
- Type in the expression you want to evaluate. For example, you can enter something like '2+3*(5-1)'.
- NOTE: You can't use spaces since the terminal will parse your input with spaces as multiple arguments if you use spaces. Therefore, adhere closely to the example provided just above where there are no spaces and the string is input as one coherent collection.

Step 3: Understand Operator Precedence

- Be aware of the order of operations (operator precedence). Our software follows the rule ordering: parenthesis as highest priority, division, multiplication, and modulus (from left to right) as second priority, and addition and subtraction (from left to right) as last priority.
- In the example '2+3*(5-1)', the expression inside the brackets is evaluated first, then the multiplication, and finally, the addition.

Step 4: Use Parentheses for Clarity

- If your expression involves complex operations, use parentheses to explicitly define the order of evaluation. For example, '(2+3)*(5-1)' ensures that the addition inside the first set of parentheses is done before the multiplication.

Step 5: Utilize Arithmetic Operators

- Use standard arithmetic operators for basic operations:
  - '+' for addition.
  - '-' for subtraction.
  - '*' for multiplication.
  - '/' for division.
  - '%' for modulus.
  - '(' and ')' for order of operations.

Step 6: Review and Execute

- Double-check your expression for correctness.
- Execute the calculation, by hitting "enter".

Step 7: Interpret the Results

- After entering the expression, the software should display the result of the calculation.

Step 8: Handle Errors

- If there are errors in your expression, the software may provide error messages. Carefully review these messages to identify and correct any mistakes.

Step 9: Exiting the Program

- When you would like to exit the program, enter "stop"

4. Advanced features

The current version of the software primarily focuses on parsing and evaluating arithmetic expressions with standard operators (+, -, *, /, %, and ^) and supports the use of parentheses to define precedence. At this stage, the program does not include advanced features such as saving and loading expressions or defining custom variables and functions. The goal is to provide a lightweight and efficient tool for basic arithmetic calculations.

5. Troubleshooting

5.1 Compiling The Program

If you are having trouble compiling the program, ensure the following:

- Your O.S. has a C++ compiler installed, preferably gcc-13 for maximum compatibility.
- You selected the corrected O.S. in the Purpose section.
- Verify that your compiler is properly configured and accessible from the command line or integrated development environment IDE.
- If no executable is found in the folder for your O.S. or the provided executable for whatever reason isn't working, compile your own executable and run that in terminal.

5.2 Incorrect Output

If the program displays the incorrect evaluation, ensure that you computed the expression correctly.

5.3 Incorrect Input

If the program cannot evaluate the given expression, ensure you correctly formatted and inputted the expression. NOTE: you can not enter spaces.

6. Examples

The following are just some examples of expressions you can input into Calc-U-Later:

- (2+4)-(3+9)-4%2
- ((22))+(14)/22
- (2)^(15%6)
- (22+14/24*(2*8))^(4/3)

7. Glossary of terms

Terminal: A terminal serves as a text-based gateway to the computer, allowing users to input commands, handle files, run programs, and access documents through textual interactions.

Compiler: A compiler is a specialized software tool designed to convert the source code of a programming language into machine code, bytecode, or another programming language. The source code is commonly composed in a high-level, human-readable language like Java or C++.

IDE: An integrated development environment (IDE) is a software application designed to enhance the efficiency of software code development for programmers. By consolidating features like software editing, building, testing, and packaging into a user-friendly application, an IDE boosts developer productivity.

Executable: An executable file, often denoted by the extension ".exe", is a computer file housing a coded set of instructions that the system can directly execute upon the user's click on the file icon. While the majority of executable files bear the common EXE extension, there exist numerous other formats for executable files.

Operating System (O.S.): An operating system is a software component that serves as an intermediary between computer hardware and the computer user. It provides a set of essential services and functions that allow other software applications to run on a computer or device.

8. FAQ

Q1: What does the software do?

A: The software is designed to parse and evaluate arithmetic expressions. It can handle basic operations such as addition, subtraction, multiplication, division, modulus, and exponentiation, while also supporting the use of parentheses for precedence and grouping.

Q2: How do I enter an arithmetic expression?

A: You can enter arithmetic expressions directly into the program's user interface or input prompt. Simply type the expression using numeric constants and supported operators.

Q3: What operators are supported?

A: The software supports the following operators:

+ (addition)

- (subtraction)

* (multiplication)

/ (division)

% (modulus)

^ (exponentiation)

Q4: Can I use parentheses in my expressions?

A: Yes, parentheses can be used to define precedence and group operations. For example, '(2+3)*4' will prioritize the addition before multiplication.

Q5: Is there a limit to the length or complexity of expressions?

A: The software is designed to handle a wide range of expressions, but extremely long or complex expressions might encounter limitations. In practical scenarios, it should handle most everyday arithmetic calculations.

Q6: How does the software handle errors in expressions?

A: The program includes error handling mechanisms. If an expression contains invalid syntax, division/modulus by zero, or mismatched parentheses, the software will provide a clear error message to guide the user.

Q7: Can I save or load expressions for later use?

A: The current version of the software does not support saving or loading expressions. It is intended for real-time evaluation of arithmetic expressions.

Q8: Are custom variables or functions supported?

A: No, the software does not currently support custom variables or functions. It focuses on basic arithmetic calculations without introducing variables or user-defined functions.

Q9: Is there a graphical user interface (GUI)?

A: No, there is not a graphical user interface. The entirety of the user interface exists inside of the terminal on the command line.

Q10: How can I exit the program?

A: You can exit the program anytime by typing "stop".