<Company Name>

Calculator Project

User's Manual

Version <1.0>

Revision History

Date	Version	Description	Author
12/03/2023	1.0	Specify User's Manual for Calculator Project	Sean Brady, Chris Harvey, Kristin Boeckmann, Naran Bat, Peter Walsdorf, William Morris

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

User's Manual

1. Purpose

The purpose of this program is to provide a fully functional calculator that has functions of addition, subtraction, multiplication, division, modulo, and exponentiation, as well as handling parentheses and using order of operations.

2. Introduction

This software is designed to work as a calculator with all PEMDAS functions and order of operations as well as the modulo feature. You can also assign variables to values or expressions so you can save and use those variables at another time. This program can be installed by downloading the software off GitHub and then running it in the terminal.

Instructions for other platforms (Mac OS Intel/AMD, Linux dists)

1. Run "make clean" target on source code folder
2. Run "make" to compile and generate a binary for your specific device
3. After compilation, ensure executable has permissions to run on your system by running the following command:
   chmod +x ./calculator
4. run application
   ./calculator
   and follow prompts

3. Getting started

After you have the program running, you type an expression into the input line using numbers and functions to get an output. Use the following operators and symbols:

- Addition: +
- Subtraction: -
- Multiplication: *
- Division: /
- Modulo: %
- Exponentiation: ^
- Parentheses: ()
- Denote decimals with a period (e.g., 1.5)
- Denote a negative number with a minus sign (e.g., -2)
- Denote a positive number with a plus sign (e.g., +3)

Invalid characters will result in an error. Spaces between symbols, numbers and operands will NOT result in an error and can be used for better visualization. Ensure proper grouping of parentheses and operands. Results will be displayed under the input line as an output or decimal number.

For help, enter 'help'

4. Advanced features

An advanced feature is being able to save expressions and values to variables. For example:
Input: A = 5
would save A as 5 to be used later. Another example is:
Input: A = 5*3*7
would save A as 105 to be used late. This works for any valid expression, and is useful for saving complicated expressions that are used multiple times, so they don't have to be entered over and over.
There is a table which outputs the current value of all variables each time a new variable is assigned or an existing variable is reassigned. This can be used to keep track of all currently declared variables.

5. Troubleshooting

One common problem would be having open parentheses, as some calculators allow this to occur, but users need to make sure that there is a right parentheses for every left one.
If a segmentation fault occurs, restart the program with "./calculator"

Some users have encountered problems running the provided binaries on their systems, because of the diverse nature of the hardware landscape. This issue can be entirely circumvented by compiling your own binary executables! Since our source code is openly available, you can simply clone it, utilize our provided makefile to clean and recompile using commands below in a unix or bash terminal, and run our application via your own binary.

```
make clean
make
```

6. Examples

input: '2+2' result: '4'
input: '6-3' result: '3'
input: 'A+4' where A=(9-2)*4 result: '32'
input: '7*(8/2)' result: '28'
input: 'A=2' result: 'Current variable values: A=2'
input '2%2' result: '0'

7. Glossary of terms

Modulo - this function returns the remainder of a division operation.
GitHub - is a software platform where developers can store and manage software. It is common for code to be posted to be publicly used here.
Segmentation fault - a segmentation fault, or 'seg fault', is an error caused by accessing memory that does not belong to you, and stops you from corrupting memory.

8. FAQ

Q: How do you access the help menu?
A: You can type in the command "help" to the user interface to retrieve the user help menu.

Q: How do you assign variables?
A: You can assign variables using the name of the variable you want followed by an equal sign, "=" and then the value or the expression you want that variable assigned to. Some examples: "a=2" or "b=a+2/(s%2)".

Q: How do I access the code for this project?
A: You can access the source code for this project on our github.

Q: How do I quit this program?
A: You can quit the program by entering "exit" into the user interface.

Q: How do I run this program?
A: You can run the program by using the make command to compile the binary then enter "./calculator" into the bash or unix terminal of your choice.