
EECS 348 Group 14

Arithmetic Expression Evaluator in C++

User's Manual

Version <1.0>

EECS 348 Group 14	Version: <1.0>
User's Manual	Date: 11/30/23
<document identifier>	

Revision History

Date	Version	Description	Author
11/30/23	1.0	First draft of user manual	Vincent Dick

EECS 348 Group 14	Version: <1.0>
User's Manual	Date: 11/30/23
<document identifier>	

Table of Contents

1. Purpose	4
2. Introduction	4
3. Getting started	4
4. Advanced features	5
5. Troubleshooting	5
6. Example of uses	5
7. Glossary	7
8. FAQ	7

EECS 348 Group 14	Version: <1.0>
User's Manual	Date: 11/30/23
<document identifier>	

Test Case

1. Purpose

This document will provide an overview of the usage of the AEEC. The proper use of the AEEC will be outlined, including how to use the AEEC, features of the AEEC, possible troubleshooting that might be needed, and examples of usage. A glossary for terms used and an FAQ is included for additional information.

2. Introduction

The purpose of the AEEC is to calculate expressions given to it by the user using the PEMDAS order of operations. This means the AEEC has the ability to add, subtract, multiply, divide, use an exponent, use a modulus, and group expressions in parentheses. The AEEC also has the ability to perform these on either an integer or decimal number. To install and run it, you must download the source files from GitHub, compile the code in the terminal of the local machine, and run the executable.

3. Getting started

To run the program, the user must compile the code locally on their machine. To do this, the local machine should have a C++ compiler installed like g++. MacOS and UNIX systems should natively support C++ compiling. Otherwise, to check if your machine has the g++ compiler, open a command prompt and run this command:

```
g++ --version
```

If no version of g++ is found on the local machine, for Windows users, you will need to download MSYS2, which can be found at the following link: <https://www.msys2.org/>. We recommend following their “Getting Started” guide for installing MSYS2, as it will also lead you through installing g++. After installing g++ you will need to navigate to the folder where the AEEC files have been downloaded to your local machine. If you clone the files from GitHub, they will be in a folder named EECS348_Project. Navigate into the EECS348_Project folder.

```
cd [download_directory]/EECS348_Project/
```

Once you’ve navigated to the project folder in the terminal, run the Makefile to compile the code.

```
make
```

Finally, use the following command to run the executable from the terminal.

```
./build/aeec
```

Now, the user should have the AEEC running in their terminal, and the user will be continually prompted to input an expression. Valid operators for use in the AEEC include:

- + (addition OR positive unary operator)
- - (subtraction OR negative unary operator)
- * (multiplication)
- / (division)
- ^ (exponent)
- % (modulo)

EECS 348 Group 14	Version: <1.0>
User's Manual	Date: 11/30/23
<document identifier>	

Parentheses can be used if a user wants certain operations to take precedence over other operations in the expression. The user may press ENTER once they are finished typing an expression to submit the expression to the AEEC for evaluation. After an expression is evaluated, the result will be printed to the terminal, and the user will be prompted for another expression. To exit the program, the user can type “quit” into an empty prompt instead of an equation and press ENTER, or may also close out of the terminal entirely. For issues encountered, refer to the troubleshooting section.

4. Advanced features

The AEEC can take decimal numbers up to 15 decimal places as operands. This also means that the answer resulting from the expression entered can have up to 15 decimal places of accuracy.

5. Troubleshooting

Compilation:

If you don't have a Windows machine and your machine doesn't have a compiler for C and C++, then please use your internet browser to find a suitable compiler for your operating system. If the provided commands don't seem to work on your machine, please look up the proper syntax for navigating and using your machine's terminal.

Calculator Usage:

If the AEEC is given an expression it cannot evaluate, it will print an appropriate error message instead of attempting to evaluate the expression. If you're still receiving what seems to be an incorrect result, double-check the input to make sure your expression is what you intended to have evaluated. The user must refrain from using numbers with decimals larger than 15 decimal places, as the AEEC doesn't support decimal accuracy beyond 15 decimal places, and the result of an expression in which a number contains more than 15 decimals may be incorrect.

6. Examples

Once the AEEC is running, it will prompt the user with “Enter an Expression”. For example, if the user wants to add three and four together, then the user will enter “3 + 4” into the terminal. Then the AEEC will print out “<result>”, where <result> is the result of the expression, if valid. So, for this example, it would print out “7”. The result of this example is provided below:

```
Enter an expression:
3 + 4
7
```

The AEEC will prompt you again for an expression again, and will continually do so until the user enters “quit”. As another example, the subtraction of three numbers (8, 5, and 2) would look like the following:

```
Enter an expression:
8 - 5 - 2
1
```

The AEEC also solves expressions that contain multiplication and division. An example in which both of these operators are used is provided below:

```
Enter an expression:
10 * 2 / 5
4
```

EECS 348 Group 14	Version: <1.0>
User's Manual	Date: 11/30/23
<document identifier>	

The AEEC can also process exponents. Below is an example in which the exponential operator is used:

Enter an expression:

```
2 ^ 3
8
```

The AEEC can also process the modulo operation, where the result is the remainder of a number divided by another number. An example of the usage of this operator is below:

Enter an expression:

```
5 % 2
1
```

The AEEC uses parentheses to group expressions together that you want to be solved first. Parentheses may be used in the AEEC to group expressions together or give precedence to certain operations within the expression. For example, if the user wants to evaluate five multiplied by the addition of three and two, you can group the addition of three and two using parentheses so that that part of the expression is evaluated first. The result of this example is provided below:

Enter an expression:

```
5 * (3 + 2)
25
```

The AEEC can also evaluate expressions in which there are parentheses that are unnecessary. In the following example, the multiplication, division, and modulo operations have a higher priority (according to PEMDAS), so those operations are evaluated before the subtraction and addition anyway:

Enter an expression:

```
(5 * 2) - (3 / 1) + (4 % 3)
8
```

This also applies to the following example, where there are multiple parentheses wrapped around a single operation:

Enter an expression:

```
((5 * 2) - ((3 / 1) + ((4 % 3))))
6
```

The AEEC also supports unary operators. An example of the usage of these operations is below:

Enter an Expression:

```
+(-2) * (-3) - ((-4) / (+5))
6.8
```

While the AEEC can handle a variety of operations and expressions, there are cases where the user input is unable to be evaluated and will result in an error with a message explaining what the error is. For example, if there are unmatched parentheses in the equation you put in, the AEEC results in an error. The following is an example of this:

Enter an expression:

```
2 * (4 + 3 - 1
Unbalanced Parentheses
```

The AEEC will also result in an error if there are too many operators in an expression:

EECS 348 Group 14	Version: <1.0>
User's Manual	Date: 11/30/23
<document identifier>	

```
Enter an expression:
* 5 + 2
Too Many Operators
```

There are many characters that the AEEC does not process and, therefore, will not calculate. Below is an example of this error:

```
Enter an expression:
((7*3) @ 2)
Invalid Character '@'
```

Dividing by 0 is not possible, so the AEEC will result in an error if there is a division by 0 at any point in the expression. An example of that happening is shown below:

```
Enter an expression:
4 / 0
Error: Division by Zero
```

There must be a binary operator between the exterior of two parentheses or between the exterior of a parenthesis and an operand; otherwise, there will be an error. An example of this error occurring is below:

```
Enter an expression:
5 (2 + 3)
Improper Parenthesis Usage
```

When a user is ready to quit, they may enter “quit” as their input:

```
Enter an expression:
quit
Exiting...
```

7. Glossary of terms

- AEEC (Arithmetic Expression Evaluator in C++)
- PEMDAS (Parentheses, exponents, multiplication, division, addition, subtraction: method of evaluating mathematical expressions)
- Unary operator (An operator with only one operand.)

8. FAQ

Why do I have to compile the AEEC on my local machine?

The files on GitHub are the source files, so they need to be compiled into an executable. This also ensures that as long as the machine being used has a C++ compiler, the AEEC can be compiled and used.

How do I exit the AEEC when I'm finished using it?

When prompted for an expression, instead, enter “quit” and press ENTER. This will exit the AEEC. The AEEC may also be exited by closing the terminal in which it is running.

Why is the result of my expression wrong?

EECS 348 Group 14	Version: <1.0>
User's Manual	Date: 11/30/23
<document identifier>	

The AEEC will correctly evaluate any expression as long as it's a valid expression, so any unexpected result will come from user input error, such as using the wrong operator or having parentheses around the wrong expression.

Why can't I use variables?

The AEEC is designed to only evaluate expressions involving integers and decimal numbers. The usage of variables was not included in the scope of this project.