C++ Arithmetic Evaluator

Version 0.1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 03/Dec/2023 | 0.1 | Compilation of test results from GitHub CI/CD for v1.0.0 release | Cody Duong |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

1. Purpose 4

2. Test Cases 4

2.1 Manual Test Cases 4

2.1.1 Other Manual Tests 5

2.2 Automatic Test Cases 6

2.2.1 Test\_ASTNode.cpp 6

2.2.2 Test\_ExpressionParser.cpp 8

# Purpose

The Test Case Specification document for the C++ Arithmetic Evaluator defines test cases for items that should be tested. This document is a compilation of the GitHub workflow results from the GitHub repository for the project as well as a collection of some manual tests.

***These test cases were compiled from source code tagged v1.0.0***. (<https://github.com/codyduong/EECS-348-Project/releases/tag/1.0.0>)

# Test Cases

## Manual Test Cases

Manual test cases are run manually. The inputs and results are compiled below. They are run through the command line in `-c` mode.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Test Data** | **Expected** | **Received** | **Pass/Fail** |
| Manual-1 | Expression or number | "1++1" | “Error: Expected an expression or number, received: +” | “Error: Expected an expression or number, received: +” | Pass |
| Manual-2 | Mismatched open parenthesis | "(1 + 1" | “Error: Expected closing parenthesis, received: \\0 (end of expression)” | “Error: Expected closing parenthesis, received: \\0 (end of expression)” | Pass |
| Manual-3 | Mismatched close parenthesis | "1 + 1)" | "Error: Missing opening parenthesis" | "Error: Missing opening parenthesis" | Pass |
| Manual-4 | Complex expression 1 | “3 + 2 - 1 / 2 \* 4 ^ 2 + 1” | “-2” | “-2” | Pass |
| Manual-5 | Complex expression 2 | “3 \*( 1 + 2) - (( 4 + 4) \* 3) / 2” | “-3” | “-3” | Pass |
| Manual-6 | Implicit multiplication | “1(2)” | “Error: Expected an operator, received: (“ | “Error: Expected an operator, received: (“ | Pass |
| Manual-7 | Modulo Remainder + | “-7%-3” | “2” | “2” | Pass |

### Other Manual Tests

These tests contain the whole command line string, with expected input and output string. Separate input strings indicate multiple commands, with each string data corresponding to each expected and received.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Test Data** | **Expected** | **Received** | **Pass/Fail** |
| Cli-1 | Argument expression | “expression\_parser 1+1” | “2” | “2” | Pass |
| Cli-2 | -c Flag | “expression parser -c”,  “1+1”,  “\04” (CTRL+D) | “Enter an algebraic expression (type 'quit', 'exit', or press CTRL+D to exit loop):”,  “2”,  Exits program | “Enter an algebraic expression (type 'quit', 'exit', or press CTRL+D to exit loop):”,  “2”,  Exits program (0) | Pass |
| Cli-3 | --configuration Flag | “expression parser -c”,  “1+1”,  “\04” (CTRL+D) | “Enter an algebraic expression (type 'quit', 'exit', or press CTRL+D to exit loop):”,  “2”,  Exits program (0) | “Enter an algebraic expression (type 'quit', 'exit', or press CTRL+D to exit loop):”,  “2”,  Exits program (0) | Pass |
| Cli-4 | Argument and -c Flag | “expression parser --configuration” | “-c/--configuration flag is mutually exclusive with an arithmetic input. Please specify one or the other.”,  Exits program (1) | “-c/--configuration flag is mutually exclusive with an arithmetic input. Please specify one or the other.”,  Exits program (1) | Pass |
| Cli-5 | -v Flag | “expression parser 1+1 -c -v” | “Version 1.0.0”,  Exits program (0) | “Version 1.0.0”,  Exits program (0) | Pass |

## Automatic Test Cases

These test cases are a compilation of the automatically run tests described in the folder in the source repository [`tests/\*\*`](https://github.com/codyduong/EECS-348-Project/tree/main/tests), and were run using [GoogleTest](https://github.com/google/googletest) in a GitHub Actions pipeline (defined here: <https://github.com/codyduong/EECS-348-Project/blob/main/.github/workflows/test.yaml>) which runs on every pull request or merge to master. For the most recent automatic tests (held for 90 days) view at <https://github.com/codyduong/EECS-348-Project/actions?query=branch%3Amain>.

The ID is a simplified test\_suite\_name concatenated with iterated integer. The test case description is simply the test\_name.

For std::runtime\_error results, the test implementation can be checked in the tests source folder to ensure that the correct error string was thrown.

### Test\_ASTNode.cpp

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Description** | **Left** | **OP** | **Right** | **Expected** | **Received** | **Pass/Fail** |
| NumberNodeEvaluate-1 | NumberNode Evaluate | 42.0 |  |  | 42.0 | 42.0 | Pass |
| NumberNodeToString-1 | NumberNode ToString | 42.0 |  |  | "42.000000" | "42.000000" | Pass |
| Addition-1 | Addition Evaluate | 20.0 | + | 22.0 | 42.0 | 42.0 | Pass |
| Addition-2 | Addition Evaluate Decimal | 5.5 | + | 7.33 | 12.83 | 12.83 | Pass |
| Addition-3 | Addition Negative 1 | 20.0 | + | -22.0 | -2.0 | -2.0 | Pass |
| Addition-4 | Addition Negative 2 | -20.0 | + | 22.0 | 2.0 | 2.0 | Pass |
| Addition-5 | Addition Negative 3 | -20.0 | + | -22.0 | -42.0 | -42.0 | Pass |
| Addition-6 | Addition ToString | 20.0 | + | 22.0 | "(20.000000 + 22.000000)" | "(20.000000 + 22.000000)" | Pass |
| Subtraction-1 | Subtraction Evaluate | 20.0 | - | 22.0 | -2.0 | -2.0 | Pass |
| Subtraction-2 | Subtraction Evaluate Decimal | 5.5 | - | 7.33 | -1.83 | -1.83 | Pass |
| Subtraction-3 | Subtraction Negative 1 | 20.0 | - | -22.0 | 42.0 | 42.0 | Pass |
| Subtraction-4 | Subtraction Negative 2 | -20.0 | - | 22.0 | -42.0 | -42.0 | Pass |
| Subtraction-5 | Subtraction Negative 3 | -20.0 | - | -22.0 | 2.0 | 2.0 | Pass |
| Subtraction-6 | Subtraction ToString | 20.0 | - | 22.0 | "(20.000000 - 22.000000)" | "(20.000000 - 22.000000)" | Pass |
| Multiplication-1 | Multiplication Evaluate | 6.0 | \* | 7.0 | 42.0 | 42.0 | Pass |
| Multiplication-2 | Multiplication Evaluate Decimal | 0.1 | \* | 0.2 | 0.02 | 0.02 | Pass |
| Multiplication-3 | Multiplication Negative 1 | 6.0 | \* | -7.0 | -42.0 | -42.0 | Pass |
| Multiplication-4 | Multiplication Negative 2 | -6.0 | \* | 7.0 | -42.0 | -42.0 | Pass |
| Multiplication-5 | Multiplication Negative 3 | -6.0 | \* | -7.0 | 42.0 | 42.0 | Pass |
| Multiplication-6 | Multiplication ToString | 6.0 | \* | 7.0 | "(6.000000 \* 7.000000)" | "(6.000000 \* 7.000000)" | Pass |
| Exponentiation-1 | Exponentiation Evaluate | 2.0 | ^ | 3.0 | 8.0 | 8.0 | Pass |
| Exponentiation-2 | Exponentiation Evaluate Decimal | 2.5 | ^ | 2.0 | 6.25 | 6.25 | Pass |
| Exponentiation-3 | Exponentiation Zero Exponent | 5.0 | ^ | 0.0 | 1.0 | 1.0 | Pass |
| Exponentiation-4 | Exponentiation Negative Base | -2.0 | ^ | 2.0 | 4.0 | 4.0 | Pass |
| Exponentiation-5 | Exponentiation Negative Exponent | 3.0 | ^ | -2.0 | 1.0 / 9.0 | 1.0 / 9.0 | Pass |
| Exponentiation-6 | Exponentiation Decimal Base | 1.5 | ^ | 2.0 | 2.25 | 2.25 | Pass |
| Exponentiation-7 | Exponentiation Decimal Exponent | 2.0 | ^ | 0.5 | sqrt(2.0) | sqrt(2.0) | Pass |
| Exponentiation-8 | Exponentiation Zero Base | 0.0 | ^ | 3.0 | 0.0 | 0.0 | Pass |
| Exponentiation-9 | Exponentiation Zero Base Zero Exponent | 0.0 | ^ | 0.0 | 1.0 | 1.0 | Pass |
| Exponentiation-10 | Exponentiation ToString | 2.0 | ^ | 3.0 | "(2.000000 ^ 3.000000)" | "(2.000000 ^ 3.000000)" | Pass |
| Modulo-1 | Modulo Evaluate | 10.0 | % | 3.0 | 1.0 | 1.0 | Pass |
| Modulo-2 | Modulo Evaluate Decimal | 5.5 | % | 2.0 | 1.5 | 1.5 | Pass |
| Modulo-3 | Modulo Negative 1 | 10.0 | % | -3.0 | 1.0 | 1.0 | Pass |
| Modulo-4 | Modulo Negative 2 | -7.0 | % | 3.0 | 2.0 | 2.0 | Pass |
| Modulo-5 | Modulo Negative 3 | -10.0 | % | -3.0 | 2.0 | 2.0 | Pass |
| Modulo-6 | Modulo ToString | 10.0 | % | 3.0 | "(10.000000 % 3.000000)" | "(10.000000 % 3.000000)" | Pass |
| Modulo-7 | Modulo Division By Zero | 10.0 | % | 0.0 | - | - | Pass |
| Modulo-2 | Modulo Evaluate Decimal | 5.5 | % | 0 | std::runtime\_error | std::runtime\_error | Pass |

### Test\_ExpressionParser.cpp

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ID** | **Description** | **Test Data** | **Expected** | **Received** | **Pass/Fail** |
| Special-1 | UnaryNegative | "-1" | -1 | -1 | Pass |
| Addition-1 | NoWhitespace | "1+1" | 2 | 2 | Pass |
| Addition-2 | OperatorLeadingWhitespace | "1 +1" | 2 | 2 | Pass |
| Addition-3 | OperatorTrailingWhitespace | "1+ 1" | 2 | 2 | Pass |
| Addition-4 | OperatorLeadingAndTrailingWhitespace | "1 + 1" | 2 | 2 | Pass |
| Addition-5 | LeadingWhitespace | " 1+1" | 2 | 2 | Pass |
| Addition-6 | TrailingWhitespace | "1+1 " | 2 | 2 | Pass |
| Addition-7 | OperatorLeadingAndTrailingWhitespaceAndExpressionLeadingAndTrailingWhitespace | " 1 + 1 " | 2 | 2 | Pass |
| Addition-8 | NegativeAddNegative | "-1+-1" | -2 | -2 | Pass |
| Addition-9 | NegativeAddPositive | "-1+5" | 4 | 4 | Pass |
| Addition-10 | PositiveAddNegative | "1+-4" | -3 | -3 | Pass |
| Additon-11 | DuplicateAddOperator | "1++1" | std::runtime\_error | std::runtime\_error | Pass |
| Subtraction-1 | NoWhitespace | "1-1" | 0 | 0 | Pass |
| Subtraction-2 | OperatorLeadingWhitespace | "1 -1" | 0 | 0 | Pass |
| Subtraction-3 | OperatorTrailingWhitespace | "1- 1" | 0 | 0 | Pass |
| Subtraction-4 | OperatorLeadingAndTrailingWhitespace | "1 - 1" | 0 | 0 | Pass |
| Subtraction-5 | LeadingWhitespace | " 1-1" | 0 | 0 | Pass |
| Subtraction-6 | TrailingWhitespace | "1-1 " | 0 | 0 | Pass |
| Subtraction-7 | OperatorLeadingAndTrailingWhitespaceAndExpressionLeadingAndTrailingWhitespace | " 1 - 1 " | 0 | 0 | Pass |
| Subtraction-8 | NegativeSubtractNegative | "-1--1" | 0 | 0 | Pass |
| Subtraction-9 | NegativeSubtractPositive | "-1-5" | -6 | -6 | Pass |
| Subtraction-10 | PositiveSubtractNegative | "1--4" | 5 | 5 | Pass |
| Multiplication-1 | NoWhitespace | "2\*2" | 4 | 4 | Pass |
| Multiplication-2 | OperatorLeadingWhitespace | "2 \*2" | 4 | 4 | Pass |
| Multiplication-3 | OperatorTrailingWhitespace | "2\* 2" | 4 | 4 | Pass |
| Multiplication-4 | OperatorLeadingAndTrailingWhitespace | "2 \* 2" | 4 | 4 | Pass |
| Multiplication-5 | LeadingWhitespace | " 2\*2" | 4 | 4 | Pass |
| Multiplication-6 | TrailingWhitespace | "2\*2 " | 4 | 4 | Pass |
| Multiplication-7 | OperatorLeadingAndTrailingWhitespaceAndExpressionLeadingAndTrailingWhitespace | " 2 \* 2 " | 4 | 4 | Pass |
| Multiplication-8 | NegativeMultiplyNegative | "-2\*-2" | 4 | 4 | Pass |
| Multiplication-9 | NegativeMultiplyPositive | "-2\*4" | -8 | -8 | Pass |
| Multiplication-10 | PositiveMultiplyNegative | "3\*-2" | -6 | -6 | Pass |
| Multiplication-11 | DuplicateMultiplyOperator | "2\*\*2" | std::runtime\_error | std::runtime\_error | Pass |
| Exponentiation-1 | PositiveExponentPositive | "2^3" | 8 | 8 | Pass |
| Exponentiation-2 | NegativeBasePositiveExponent | "-2^2" | 4 | 4 | Pass |
| Exponentiation-3 | PositiveBaseNegativeExponent | "3^(-2)" | 1/9 | 1/9 | Pass |
| Exponentiation-4 | BaseWithWhitespaceExponent | "2 ^ 4" | 16 | 16 | Pass |
| Exponentiation-5 | ExponentiationWithMultipleOperators | "2 + 3^2" | 11 | 11 | Pass |
| Exponentiation-6 | ExponentiationWithParentheses | "(2 + 3)^2" | 25 | 25 | Pass |
| Exponentiation-7 | ExponentiationWithNegativeBaseAndExponent | "(-2)^(-3)" | -0.125 | -0.125 | Pass |
| Exponentiation-8 | ExponentiationWithNegativeBase | "(-2)^3" | -8 | -8 | Pass |
| Exponentiation-9 | ExponentiationWithZeroBase | "0^2" | 0 | 0 | Pass |
| Exponentiation-10 | ExponentiationWithZeroExponent | "3^0" | 1 | 1 | Pass |
| Exponentiation-11 | DuplicateExponentOperator | "2^^2" | std::runtime\_error | std::runtime\_error | Pass |
| Modulus-1 | NoWhitespace | "7%3" | 1 | 1 | Pass |
| Modulus-2 | OperatorLeadingWhitespace | "7 %3" | 1 | 1 | Pass |
| Modulus-3 | OperatorTrailingWhitespace | "7% 3" | 1 | 1 | Pass |
| Modulus-4 | OperatorLeadingAndTrailingWhitespace | "7 % 3" | 1 | 1 | Pass |
| Modulus-5 | LeadingWhitespace | " 7%3" | 1 | 1 | Pass |
| Modulus-6 | TrailingWhitespace | "7%3 " | 1 | 1 | Pass |
| Modulus-7 | OperatorLeadingAndTrailingWhitespaceAndExpressionLeadingAndTrailingWhitespace | " 7 % 3 " | 1 | 1 | Pass |
| Modulus-8 | NegativeModulusPositive | "-7%3" | 2 | 2 | Pass |
| Modulus-9 | PositiveModulusNegative | "7%-3" | 1 | 1 | Pass |
| Modulus-10 | NegativeModulusNegative | "-7%-3" | 2 | 2 | Pass |
| Modulus-11 | DivisionByZero | "10%0" | std::runtime\_error | std::runtime\_error | Pass |
| FullExpression-1 | MultiplicationPrecedence | "2 + 3 \* 4 - 1" | 13 | 13 | Pass |
| FullExpression-2 | TwoParenthesisPrecedence | "(2 + 3) \* (4 - 1)" | 15 | 15 | Pass |
| FullExpression-3 | OneParenthesisPrecedence | "(2 + 3) \* 4 - 1" | 19 | 19 | Pass |
| FullExpression-4 | ExponentialPrecedence | "2 + 2 ^ 2" | 6 | 6 | Pass |
| FullExpression-5 | ExponentialParenthesis | "(2 + 2) ^ 2" | 16 | 16 | Pass |
| FullExpression-6 | ParenthesisExponentialPrecedence | "1 + 2 \* (2 + 3)^2" | 51 | 51 | Pass |
| FullExpression-7 | ExponentialPrecedenceOverAdd | "( 2 + 3) ^ 2 + 1" | 26 | 26 | Pass |
| FullExpression-8 | ParenthesisPrecedenceOverExponential | "( 2 + 3) ^ (2 + 1)" | 125 | 125 | Pass |
| FullExpression-9 | LargeExpression | "3 + 2 - 1 / 2 \* 4 ^ 2 + 1" | -2 | -2 | Pass |
| FullExpression-10 | LargeExpressionParenthesis | "3 \*( 1 + 2) - (( 4 + 4) \* 3) / 2" | -3 | -3 | Pass |
| FullExpression-11 | NegativeExponent | "2^ -2" | 0.25 | 0.25 | Pass |
| FullExpression-12 | ModPrecedence | "1 + 3 % 2" | 2 | 2 | Pass |
| FullExpression-13 | ModPrecedenceParenthesis | "(1 + 3) % 2" | 0 | 0 | Pass |
| FullExpression-14 | EnclosedEmptyParenthesisAfter | "(1 + 1 ())" | std::runtime\_error | std::runtime\_error | Pass |
| FullExpression-15 | EmptyParenthesisAfter | "1 + 1 ()" | std::runtime\_error | std::runtime\_error | Pass |
| FullExpression-16 | EnclosedEmptyParenthesisBefore | "(() 2 - 1)" | std::runtime\_error | std::runtime\_error | Pass |
| FullExpression-17 | EmptyParenthesisBefore | "() 2 - 1" | std::runtime\_error | std::runtime\_error | Pass |
| FullExpression-18 | MismatchedParenthesisBefore | "(1 + 1" | std::runtime\_error | std::runtime\_error | Pass |
| FullExpression-19 | MismatchedParenthesisBeforeSpaced | "(1 + 1" | std::runtime\_error | std::runtime\_error | Pass |
| FullExpression-20 | MismatchedParenthesisAfter | "1 + 1)" | std::runtime\_error | std::runtime\_error | Pass |
| FullExpression-21 | MismatchedParenthesisAfterSpaced | "1 + 1 )" | std::runtime\_error | std::runtime\_error | Pass |