EECS 348 Group 14

Arithmetic Expression Evaluator in C++

Test Case

Version <1.0>

Revision History

Date	Version	Description	Author
<30/NOV/23>	1.0	First Test Case Draft	Wesley McDougal

Table of Contents

Test Case

1. Purpose

This Test Case Specification document for the Arithmetic Expression Evaluator in C++ (AEEC) defines a test case for an item that should be tested. This document includes a table of all the test cases, including for every test case:

- the ID of the test case
- the description of the test case
- the data being tested
- the expected result of the data
- the actual result of the data

The Test Case Procedure table consists of 3 sections:

- Valid Expressions Test Cases
- Invalid Expressions Test Cases
- Failed Expressions Test Cases

The Valid Expressions Test Cases determine whether the known correct arithmetic operations will output the desired results, while the Invalid Expressions Test Cases determine whether the known incorrect arithmetic operations are perceived by the AEEC and output the desired errors. The Failed Expressions Test Cases are a collection of failed test cases found in both the Valid and Invalid Expressions Test Cases so that these test cases are tracked and resolved of errors.

2. Test Case Procedure table

Test Case ID	Test Case Description	Test Data	Expected Results	Actual Results	Pass/Fail
Valid Expressions Test Cases					
VE#01	Test Binary Addition Operation	3 + 4	7	7	Pass
VE#02	Test Binary Modulo Operation	10.7 % 6	4.7	4.7	Pass
VE#03	Test Subtraction with Parentheses	8 - (5 - 2)	5	5	Pass
VE#04	Test Exponentiation with Parentheses	((2 ^ 2) ^ 3)	64	64	Pass
VE#05	Test Multiplication and Division	10 * 2 / 5	4	4	Pass
VE#06	Test Addition, Division and Modulo	23 + 50 / 5 % 3	24	24	Pass

Test Case ID	Test Case Description	Test Data	Expected Results	Actual Results	Pass/Fail
VE#07	Test Exponentiation	2 ^ 3	8	8	Pass
VE#08	Test Mixed Operators	4 * (3 + 2) % 7 - 1	5	5	Pass
VE#09	Test Mixed Operators	(6 % 4) + 9 * 4	38	38	Pass
VE#10	Test PEMDAS Ordering	6 + 4 - 18 * 2 / 6 % 2 ^ 2	8	8	Pass
VE#11	Test Complex Addition with Extraneous Parentheses	(((2 + 3))) + (((1 + 2)))	8	8	Pass
VE#12	Test Multiplication and Division with Extraneous Parentheses	((((4 * 4)) / 2)) * ((3))	24	24	Pass
VE#13	Text Mixed Operators with Extraneous Parentheses	((5 * 2) - ((3 / 1) + ((4 % 3))))	6	6	Pass
VE#14	Test Mixed Operators with Extraneous Parentheses	(5 % 2) * (((23))) - ((3 + 10))	10	10	Pass
VE#15	Test Nested Parentheses with Exponents	(((2 ^ (1 + 1)) + ((3 - 1) ^ 2))) / ((4 / 2 %3))	4	4	Pass
VE#16	Test Modulo on Nested Parentheses with Expressions	(25 % (4 * 4)) + (31 % (8 + 3))	18	18	Pass
VE#17	Test Combination of Extraneous and Necessary Parentheses	(((((5 - 3))) * (((2 + 1))) + ((2 * 3))))	12	12	Pass
VE#18	Test Extraneous Operand Parentheses and Necessary Expression Parentheses	(((5)) + (3)) * (4 - (((2))))	16	16	Pass
VE#19	Test Extraneous Parentheses with Division	((9 + 6)) / ((3 * 1) / (((2 + 2))) - 1)	-60	-60	Pass
VE#20	Test Combining Unary Operators with Arithmetic Operations	+(-2) * (-3) - ((-4) / (+5))	6.8	6.8	Pass
VE#2	Test Unary Negation with	-24 / -8	3	double free	FAIL

Test Case ID	Test Case Description	Test Data	Expected Results	Actual Results	Pass/Fail
1	Division			Aborted	
VE#22	Test Unary Negation with Expression	-(7 * 6)	-42	-42	Pass
VE#23	Test Unary Negation and Addition in Parentheses	-(+1) + (+2)	1	1	Pass
VE#24	Test Unary Addition of Negative Expression	+(2 - 9)	-7	7	FAIL
VE#25	Test Negation and Addition with Negated Parentheses	-(-(-3)) + (-4) + (+5)	-2	-2	Pass
VE#26	Test Alternating Unary Addition and Negation	-(+(-3) + 5)	-2	-8	FAIL
VE#27	Test Unary Negation and Exponentiation	+2 ^ (-3)	0.125	0.125	Pass
VE#28	Test Exponentiation of Number Less Than One	0.4 ^ 3	0.064	0.064	Pass
VE#29	Test Combining Unary Operators with Parentheses	-(+2) * (+3) - (-4) / (-5)	-6.8	-6.8	Pass
VE#30	Test Combining Unary and Binary Without Parentheses	-2 - -3 + +5 - +1	5	double free Aborted	FAIL
Invalid Expressions Test Cases					
IE#01	Test Unmatched Parenthesis with Left Parenthesis	2 * (4 + 3 - 1	Unbalanced Parentheses	Unbalanced Parentheses	Pass
IE#02	Test Unmatched Parenthesis With Right Parenthesis	4 * 7 + 9)	Unbalanced Parentheses	Unbalanced Parentheses	Pass
IE#03	Test Operators Without Operands on the Left Hand Side	* 5 + 2	Too Many Operators	Too Many Operators	Pass
IE#04	Test Operators Without Operands on the Right Hand Side	8 + 3 *	Too Many Operators	Too Many Operators	Pass
IE#05	Test Division by 0	4 / 0	Error: Division by	Error: Division by	Pass

Test Case ID	Test Case Description	Test Data	Expected Results	Actual Results	Pass/Fail
			Zero	Zero	
IE#06	Test Modulo by Zero	4 % 0	Error: Division by Zero	Error: Division by Zero	Pass
IE#07	Test Missing Operator Outside Parentheses (Left)	5 (2 + 3)	Improper Parentheses Usage	Improper Parentheses Usage	Pass
IE#08	Test Missing Operator Outside Parentheses (Right)	(8 + 3) 7	Improper Parentheses Usage	Improper Parentheses Usage	Pass
IE#09	Test invalid ‘& ’character	7 & 3	Invalid Character ‘&’	Invalid Character ‘&’	Pass
IE#10	Test invalid ‘$’ character	6 $ 2	Invalid Character ‘$’	Invalid Character ‘$’	Pass
IE#11	Test Left Mismatched Parenthesis	(((3 + 4) - 2) + (1)	Unbalanced Parentheses	Unbalanced Parentheses	Pass
IE#12	Test Right Mismatched Parenthesis	(1) + (2- (3 + 4)))	Unbalanced Parentheses	Unbalanced Parentheses	Pass
IE#13	Test Modulo by Zero with a Expression	((5 + 2) % (3 * 0))	Error: Division by Zero	Error: Division by Zero	Pass
IE#14	Test Division by Zero with a Expression	((5 + 2) / (3 * 0))	Error: Division by Zero	Error: Division by Zero	Pass
IE#15	Test Invalid Subtraction Operator Placement	((2 -)1 + 3)	Improper Parentheses Usage	Improper Parentheses Usage	Pass
IE#16	Test Invalid Addition Operator Placement	((1 + 3) + 2 +)	Improper Parentheses Usage	Improper Parentheses Usage	Pass
IE#17	Test missing operand at the end	(4 * 2) + 2 -	Too Many Operators	Too Many Operators	Pass

Test Case ID	Test Case Description	Test Data	Expected Results	Actual Results	Pass/Fail
IE#18	Test missing an operand at the start	((+) + (4 / 5))	Improper Parentheses Usage	Improper Parentheses Usage	Pass
IE#19	Test invalid '@'character	((7 * 3) @ 2)	Invalid Character '@'	Invalid Character '@'	Pass
IE#20	Test invalid '!' character	((4 ! 3) * 1)	Invalid Character '!'	Invalid Character '!'	Pass
Failed Expressions Test Cases					
VE#21	Test Unary Negation with Division	-24 / -8	3	3	Pass
VE#24	Test Unary Addition of Negative Expression	+(2 - 9)	-7	-7	Pass
VE#26	Test Alternating Unary Addition and Negation	-(+(-3) + 5)	-2	-2	Pass
VE#30	Test Combining Unary and Binary Without Parentheses	-2 - -3 + +5 - +1	5	5	Pass

6. Environmental Needs

6.1.3 Other

The AEEC calculator is designed to be computationally efficient and optimized for performance when calculating difficult and lengthy expressions. This program is designed to also have a minimal impact on system operations, ensuring that the program does not force unnecessary strain on the user's device. The program also has optimal resource management, ensuring that it is better managing allocated memory and resources to contribute in a more sustainable computing environment. The user should consider closing the AEEC program to better conserve system resources and device energy.