Orange

Arithmetic Expression Evaluator

Test Case

Version 1.0

Revision History

Date	Version	Description	Author
12/02/23	1.0	Create test cases for the program.	Mikaela Navarro

Table of Contents

Test Case

1. Purpose

The purpose of this Test Case document is to verify the correct functioning of the program. This includes testing basic arithmetic operations, handling parentheses, handling errors, and ensuring accurate evaluation of expressions.

2. Test cases

Test Case ID	Test Description	Test Data	Expected Results	Actual Results	Pass/Fail
TC01	Test the addition operator	3 + 4	7	7	Pass
TC02	Test the subtraction operator and parentheses	8 - (5 - 2)	5	5	Pass
TC03	Test the multiplication and division operators	10 * 2 / 5	4	4	Pass
TC04	Test the exponentiation operator	2 ^ 3	8	8	Pass
TC05	Test using a mix of operators	4 * (3+2) % 7 - 1	5	5	Pass
TC06	Test the addition operator with multiple parentheses	(((2 + 3))) + (((1 + 2)))	8	8	Pass
TC07	Test using a mix of operators with multiple parentheses	((5 * 2) - ((3 / 1) + ((4 % 3))))	6	6	Pass
TC08	Test using nested parentheses with the exponent operator	(((2 ^ (1 + 1)) + ((3 - 1) ^ 2)) / ((4 / 2) % 3))	4	4	Pass
TC09	Test an expression using a combination of extraneous and necessary parentheses	(((((5 - 3))) * (((2 + 1))) + ((2 * 3))))	12	12	Pass
TC10	Test the division operator with extraneous parentheses	((9 + 6)) / ((3 * 1) / (((2 + 2))) - 1)	-60	-60	Pass

TC11	Test an expression using combining unary operators with arithmetic operations	+(-2) * (-3) - ((-4) / (+5))	6.8	6.8	Pass
TC12	Test an expression using unary negation and addition in parentheses	-(+1) + (+2)	1	1	Pass
TC13	Test an expression using negation and addition with negated parentheses	-(-(-3)) + (-4) + (+5)	-2	-2	Pass
TC14	Test an expression using unary negation and exponentiation	+2 ^ (-3)	0.125	0.125	Pass
TC15	Test an expression using combining unary operators with parentheses	-(+2) * (+3) - (-4) / (-5)	-6.8	-6.8	Pass
TC16	Test unmatching parentheses	2 * (4 + 3 - 1	Error: Unbalanced Brackets	Error: Unbalanced Brackets	Pass
TC17	Test the operators without operands	* 5 + 2	Error: Missing Operands	Error: Missing Operands	Pass
TC18	Test an incorrect operator usage	4 / 0	Error: Undefined Operation: Division by Zero	Error: Undefined Operation: Division by Zero	Pass
TC19	Test a missing operator	5 (2 + 3)	Error: Missing Operators	Error: Missing Operators	Pass
TC20	Test invalid characters	7 & 3	Error: Invalid Character in Expression	Error: Invalid Character in Expression	Pass
TC21	Test mismatched parentheses	(((3 + 4) - 2) + (1)	Error: Unbalanced Brackets	Error: Unbalanced Brackets	Pass
TC22	Test invalid operator usage	((5 + 2) / (3 * 0))	Error: Undefined Operation: Division by Zero	Error: Undefined Operation: Division by Zero	Pass
TC23	Test invalid operator sequence	((2 -) 1 + 3)	Error: Missing Operators	Error: Missing Operators	Pass

TC24	Test missing operand	((4 * 2) + (-))	Error: Missing Operands	Error: Missing Operands	Pass
TC25	Test invalid characters	((7 * 3) @ 2)	Error: Invalid Character in Expression	Error: Invalid Character in Expression	Pass
TC26	Test decimal numbers	3.14 + 2.5	5.64	5.64	Pass
TC27	Test large numbers	1000000 ^ 2	1e+12	1e+12	Pass
TC28	Test scientific notation with negative exponents	2e3 + 4e-2	2000.04	2000.04	Pass
TC29	Test scientific notation with empty exponent	10e	Error: Invalid Numeric Constant: scientific notation invalid exponent	Error: Invalid Numeric Constant: scientific notation invalid exponent	Pass
TC30	Test scientific notation with exponent operator	3e^2 + 4e-3	Error: Invalid Numeric Constant: scientific notation invalid exponent	Error: Invalid Numeric Constant: scientific notation invalid exponent	Pass
TC31	Test scientific notation with decimal	10.3e2	1030	1030	Pass
TC32	Test scientific notation with decimal exponent	6e2.5	Error: Invalid Numeric Constant: scientific notation invalid exponent	Error: Invalid Numeric Constant: scientific notation invalid exponent	Pass
TC33	Test an expression using different parenthesis types, i.e., [], ()	3 * (4 + [5 - 2]) / (7 * [2 + 1])	1	1	Pass

3. Environmental needs

3.1.1 Hardware

N/A

3.1.2 Software

N/A

3.1.3 Other

The program should be able to run on multiple platforms (e.g., Windows, Linux, macOS) and servers.

4. Special procedural requirements

N/A

5. Intercase dependencies

N/A