Kansas Instruments

Compiler Expression Parser

Test Cases

Version 1.3

Revision History

Date	Version	Description	Author
12/09/24	1.0	Add test cases 1-4	Nevan Snider
12/10/24	1.1	Add test cases 5-8	Ayoub Lamrani
12/11/24	1.2	Format document	All
12/12/24	1.3	Add test cases 9-15 and record results	Gaven Behrends

Table of Contents

Test Cases

1. Purpose

This Test Case Specification document for the Compiler Expression Parser defines a list of test cases designed to ensure correct functionality.

2. Table columns

2.1.1 Test ID

Unique identifier for the test case, e.g. TC01, TC02, etc.

2.1.2 Description

A brief description of what capabilities the case intends to test.

2.1.3 Input

The input data that will be used by the test case.

2.1.4 Expected

The expected result of the test case.

2.1.5 Actual

The actual result of execution.

2.1.6 Status

Whether the test is passed by the current program version. (Pass/Fail)

3. Test case specifications

Test ID	**Description**	**Input**	**Expected**	**Actual**	**Status**
TC01	Verify that the calculator correctly identifies precedence of parentheses	(3+(4*2))-5	6	6	Pass
TC02	Verify that the calculator correctly handles double negatives	(-5)**2+(-5)**2	50	50	Pass
TC03	Verify that the calculator handles standard operators correctly	(3-(4+1))*(64/4%3)	-2	-2	Pass
TC04	Verify that the calculator identifies a syntax error	3+2+	Invalid Arg Error	Invalid Arg Error	Pass
TC05	Verify that the calculator handles negatives without parentheses.	-4+2^3	4	4	Pass
TC06	Verify that the calculator identifies division by zero errors.	10/(5-5)	Division by Zero Error	Division by Zero Error	Pass
TC07	Verify that the calculator identifies invalid characters in the input.	3+4*abc	Invalid Arg Error	Invalid Arg Error	Pass
TC08	Verify that the calculator handles a single value as input.	42	42	42	Pass

TC09	Verify that the calculator can handle extraneous parentheses.	(((2+3)))+(((1+2)))	8	8	Pass
TC10	Verify that the calculator handles the modulo operator.	38%6	2	2	Pass
TC11	Verify precedence of exponents over signs.	-4**2	-16	-16	Pass
TC12	Verify proper handling of repeated negations	-(-(-3)) + -(-4)	1	1	Pass
TC13	Verify behavior of negative exponents.	+2 ** -3	0.125	0.125	Pass
TC14	Verify the handling of unmatched parentheses.	2*(4+3-1	Invalid Arg Error	Invalid Arg Error	Pass
TC15	Verify the detection of invalid operator placement.	((2 -) 1 + 3)	Invalid Arg Error	Invalid Arg Error	Pass

4. Environmental needs

4.1.1 Hardware

N/A

4.1.2 Software

N/A

4.1.3 Other

All expressions fitting the requirements specified in 02-Software-Requirements-Specification must be correctly evaluated by the program.

5. Special procedural requirements

N/A

6. Intercase dependencies

N/A