C++ Calculator

Test Case

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 12/01/2023 | 1.0 | Test Cases for Arithmetic Calculator | Humza Qureshi, Zonaid Prithu |

Table of Contents

1. Purpose 4
2. Test Cases 4
3. Environmental needs 9
   * 1. Hardware 9
     2. Software ( 9
     3. Other 9
4. Special procedural requirements 9
5. Intercase dependencies 9

Test Case

# Purpose

The purpose of test cases in building the C++ calculator is to rigorously validate and ensure the accuracy, reliability, and functionality of the software. Through comprehensive testing, the aim is to confirm correct outcomes for fundamental arithmetic operations, assess the handling of edge cases and potential errors, validate adherence to the order of operations, and scrutinize floating-point precision. Additionally, the testing process will verify the calculator's responsiveness to user inputs, ensure compatibility with various numeric types, and assess overall performance, including the handling of large inputs. Ultimately, the goal is to identify and address potential issues early in development, instilling.

# Test Cases

| **Test-Case ID** | **Test-Case Description** | **Test Data** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| --- | --- | --- | --- | --- | --- |
| Add-1 | Test Addition unary operator | 4+6 | 10 | 10 | Pass |
| Add-2 | Test Addition unary operator | 2+9 | 11 | 11 | Pass |
| SubPar-1 | Test subtraction parenthesis op. | (8-2)-10 | -4 | -4 | Pass |
| SubPar-2 | Test subtraction parenthesis op. | (-4-20) - 7 | -31 | -31 | Pass |
| MultiDiv-1 | Test Multi-Div operators | 10/5\*3 | 6 | 6 | Pass |
| MultiDiv-2 | Test Multi-Div operators | 8\*8/2 | 32 | 32 | Pass |
| Expo-1 | Test Exponential operator | 3^4 | 81 | 81 | Pass |
| Expo-2 | Test Exponential operator | 2^(-2) | 0.25 | 0.25 | Pass |
| Expo-3 | Test Exponents to a degree which is less than one | 4^(0.5) | 2 | 2 | Pass |
| Expo-4 | Test exponents to a degree which is less than one | 4^(2/4) | 2 | 2 | Pass |
| Expo-5 | Test Exponents with a base of 1 | 1^7 | 1 | 1 | Pass |
| MixedOP-1 | Test combined operators and parentheses | (4/(3+2)\*9)%6 | 1.2 | 1.2 |  |
| MixedOP-2 | Test combined operators and parentheses | (4^4)/6 \* (36/6) | 256 | 256 | Pass |
| ExtranPar-1 | Test Extraneous Parentheses  with addition | ((((1+1))))+(((4+6))) | 12 | 12 | Pass |
| ExtranPar-2 | Test Extraneous Parentheses  with addition and Subtraction | ((10-2))-((6+5)) | -3 | -3 | Pass |
| ExtranParDiv-1 | Test Mixed Operations with Extraneous Parentheses | ((3 + 7))/((1 \* 2)/(((5 + -3)))-9) | -1.25 | -1.25 | Pass |
| ExtranParDiv-2 | Test Division with Extraneous Parentheses | (1+1)/(1+1)/((0+1))%2 | 1 | 1 | Pass |
| ExtranParDiv-3 | Test Division with Extraneous Parentheses | ((9 + 6))/((3 \* 1)/(((2 + 2)))-1) | -60 | -60 | Pass |
| MixedOpExtran-1 | Test Mixed Operations with extraneous parentheses | ((10 \* 2) - ((4 / 1) + ((6 % 4)))) | 14 | 14 | Pass |
| MixedOpExtran-2 | Test Mixed Operations with extraneous parentheses | (((4-2))) \* (((3+2)) % 4)/2 | 1 | 1 | Pass |
| NestedParExpo-1 | Test nested parentheses and exponentiation | (((2^(1+1))+((3-1) ^2)) / ((4/2)%3)) | 4 | 4 | Pass |
| NestedParExpo-2 | Test nested parentheses and exponentiation | ((((3+2) ^ (1 + 1)) + ((3 - 1) ^(2+-2))) / ((4 / 2) % 3)) | 13 | 13 | Pass |
| ComboExtranNess-1 | Testing both extraneous and necessary amounts of parenthesis | (((((5+2)) - (6-7)) \* 8)) | 64 | 64 | Pass |
| ComboExtranNess-2 | Testing both extraneous and necessary amounts of parenthesis | ((((13-3)))/2)+(5+2) | 12 | 12 | Pass |
| ComboExtranNess-3 | Testing both extraneous and necessary amounts of parenthesis | (((((5\*2)) - (9-7)) \* 10)) | 80 | 80 | Pass |
| ComboExtranNess-4 | Testing both extraneous and necessary amounts of parenthesis | (((((3\*0)) - ((9+-7))) \* (-10))) | 20 | 20 | Pass |
| ComboExtranNess-5 | Testing both extraneous and necessary amounts of parenthesis | ((((10+10))))+(10) | 30 | 30 | Pass |
| UnaryArth-1 | Test Unary Operators with basic Arithmetic operators | (-(+5)) \* (-(-3)) - ((-4) / (+4)) | -14 | -14 | Pass |
| UnaryArth-2 | Test Unary Operators with basic Arithmetic operators | -(-5+(-8)) \* 5/5 | 13 | 13 | Pass |
| UnaryArth-3 | Test Unary Operators with basic Arithmetic operators | -(+2) + (+2) | 0 | 0 | Pass |
| UnaryArth-4 | Test Unary Operators with basic Arithmetic operators | -(-(-10)) + (-5) + (+5) | -10 | -10 | Pass |
| UnaryArth-6 | Test Unary Operators with basic Arithmetic operators | -(+2) \* (+3) - (-5) / (-5) | -7 | -7 | Pass |
| Invalid-1 | Test unmatched parenthesis | 2 \* (4 + 3 - 1 | ERROR STATE | ERROR STATE | Pass |
| Invalid-2 | Test Unmatched parenthesis | ((3\*3)+8 | ERROR STATE | ERROR STATE | Pass |
| Invalid-3 | Test op. without operands | \*6+2 | ERROR STATE | ERROR STATE | Pass |
| Invalid-4 | Test op. without operands | (6+2) + 16/2/ | ERROR STATE | ERROR STATE | Pass |
| Invalid-5 | Test invalid arithmetic usage | 10/0 | ERROR STATE | ERROR STATE | Pass |
| Invalid-6 | Test invalid arithmetic usage | 100/(100+-100) | ERROR STATE | ERROR STATE | Pass |
| Invalid-7 | Test missing operators between numbers/functions | (7+8) 5 | ERROR STATE | ERROR STATE | Pass |
| Invalid-8 | Test missing operator between numbers/functions | 5(6+9) | ERROR STATE | ERROR STATE | Pass |
| Invalid-9 | Test invalid characters | 7&7 | ERROR STATE | ERROR STATE | Pass |
| Invalid-10 | Test Invalid characters | 7@7 | ERROR STATE | ERROR STATE | Pass |
| Invalid-11 | Test Invalid characters | (7+7)+(7@2) | ERROR STATE | ERROR STATE | Pass |
| Invalid-12 | Test Mismatch parenthesis | ((((4+2))+2) | ERROR STATE | ERROR STATE | Pass |
| Invalid-13 | Test Mismatch parenthesis | ((10\*2)+2\*(2+2) | ERROR STATE | ERROR STATE | Pass |
| Invalid-14 | Test Mismatch parenthesis | (((80+2))+70))) | ERROR STATE | ERROR STATE | Pass |
| Invalid-15 | Test mismatch parenthesis | (40+30)) | ERROR STATE | ERROR STATE | Pass |
| Invalid-16 | Test Invalid Sequence of op. | ((30-)2+3) | ERROR STATE | ERROR STATE | Pass |
| Invalid-17 | Test Invalid sequence of op. | (5 \*)+(54) | ERROR STATE | ERROR STATE | Pass |
| Invalid-18 | Test invalid sequence of op. | (5%5)+(5/) | ERROR STATE | ERROR STATE | Pass |
| Invalid-20 | Test Missing operands | (9-0)\*() | ERROR STATE | ERROR STATE | Pass |
| Invalid-21 | Test Missing operands | (0++)\*(33) | ERROR STATE | ERROR STATE | Pass |
| Invalid-22 | Test empty parenthesis | ()+() | ERROR STATE | ERROR STATE | Pass |
| Invalid-23 | Test inputs with characters and integers | helloworld007 | ERROR STATE | ERROR STATE | Pass |
| Invalid-24 | Test inputs with operators, characters, and integers | helloworld(7+7) | ERROR STATE | ERROR STATE | Pass |
| Invalid-25 | Test inputs without operators inside parenthesis | (7+7(8)) | ERROR STATE | ERROR STATE | Fail |
| Invalid-26 | Test unsupported operators (factorial) | (4)! | ERROR STATE | ERROR STATE | Pass |
| Invalid-27 | Test unsupported operators (matrix multiplication) | [1,2,3]\*[2,3,5] | ERROR STATE | ERROR STATE | Pass |
| Invalid-28 | Test unsupported operators (trigonometry) | sin(90) | ERROR STATE | ERROR STATE | Pass |
| Invalid-29 | Testing invalid characters | 7\_7 | ERROR STATE | ERROR STATE | Pass |
| Invalid-30 | Testing invalid characters | 9\9 | ERROR STATE | ERROR STATE | Pass |
| Invalid-31 | Testing equal operator | 1+1=2 | ERROR STATE | ERROR STATE | Pass |
| Invalid-32 | Testing invalid characters | 0|0|0 | ERROR STATE | ERROR STATE | Pass |
| Unconventional-1 | Test inputs with excess space | 7+                     7 | 14 | 14 | Pass |
| Unconventional-2 | Test inputs with excess space | (7+7             ) | 14 | 14 | Pass |
| Unconventional-3 | Test inputs with excess exponents and multiplication | 2^2\*2^2\*2^2\*2 | 128 | 128 | Pass |
| Unconventional-4 | Test inputs with excess exponents and multiplication | (4^2)^2 | 256 | 256 | Pass |

# **Environmental needs**

### Hardware

N/A

### Software (

N/A

### Other

N/A

# Special procedural requirements

As it relates to the test-cases above, there are no special test-cases or special set-ups that need to be created, compiled, or run in order for the cases to be tested by the test machine, user machine, or any other machine.

# Intercase dependencies

N/A