Apex Software

Tech Calculator

Test Case

Version 1.0

Revision History

Date	Version	Description	Author
12/11/2024	1.0	Initial drafting of test cases document	Gage Weaver

Table of Contents

Test Case

1.   Purpose

This document serves as the Test Case documentation for *Tech Calculator*. It contains the required test cases from the EECS-Term-Project.pdf document present in the files section of the EECS 348 canvas. Additionally, it contains extra test cases meant to ensure functionality across a wide variety of inputs.

2.   Test case identifier

This section is satisfied by the "Test Case ID" column in the testing table

3.   Test item

This section is satisfied by the "Test Case Details" and "Purpose" column of the testing table

4.   Input specifications

This section is satisfied by the "Inputs" column on the testing table.

5.   Output specifications

The actual output of the program is shown in the "Observed Outputs" column while the expected output from testing is shown in the "Expected Output" column. If these match "Pass" is entered into the "Pass/Fail" column.

6.   Environmental needs

N/A

7.   Special procedural requirements

N/A

8.   Intercase dependencies

N/A

9.   Testing Table

Below is the testing table, which contains all the information gathered while testing the calculator. Satisfying all the requirements for sections 2 through 5.

Test Case ID	Test Case Details	Purpose	Inputs	Expected Output	Observed Outputs	Pass/Fail
TC01	Two Integer Addition (Valid Ex #1)	Show calculator gets valid result for strictly addition with integers	3 + 4	7	7	Pass
TC02	Two Float Addition	Show calculator gets valid result for strictly addition with floats	6.3 + 4.8	11.1	11.1	Pass

TC03	Subtraction with Parentheses subtraction (Valid Ex #2)	Show calculator gets valid result when a subtraction occurs between an integer and the result of another subtraction in parentheses	8 - (5-2)	5	5	Pass
TC04	Multiplication and Division (Valid Ex #3)	Shows the calculator can handle multiplication and division in the same expression using the rule of PEMDAS, where the multiplication and division operations are treated at the same priority from right to left	10 * 2 / 5	4	4	Pass
TC05	Exponents (Valid Ex #4)	Shows the calculator can handle basic exponential expressions(only one expression)	2 ** 3	8	8	Pass
TC06	Intermediate PEMDAS (Valid Ex #5)	Shows the calculator can handle PEMDAS when majority of the operators appear in an inputted expression	4*(3+2)%7 - 1	5	5	Pass

TC07	Addition of two additions in parentheses with additional parentheses (Valid Ex #6)	Show calculator can handle addition with parentheses and handle extra parentheses, given there is still a balanced amount	(((2+3)))+(((1+2)))		8	8 Pass
TC08	5 operators with parentheses (Valid Ex #7)	Shows the calculator can handle all the operations at the same time	((5*2)-((3/1)+((4%3))))		6	6 Pass
TC09	5 operators (No multiplication) with some nested expressions (Valid Ex #8)	Shows the calculator can handle complex expressions with mixes of nested parentheses expressions	(((2**(1+1))+((3-1)**2))/((4/2)%3))		4	4 Pass
TC10	Triple expression with parentheses and multiple types of operators: -, +, *, ** (Valid Ex #9)	Shows the calculator can handle complex expressions with parentheses and multiple types of numeric operations	((((((5-3)))*(((2+1)))+((2**3))))		14	14 Pass
TC11	Quadruple expression with 3 nested expressions (Valid Ex #10)	Shows the calculator can handle complex, multi-expression problems, even when some expressions take input from one of the nested expressions ex: (2+2)-1	((9+6))/((3*1)/((2+2))-1)		-60	-60 Pass

TC12	Signed Integers with +, -, *, / (Valid Ex #11)	Shows the calculator can handle signed integers with the 4 main operators	" +(-2)*(-3)-((-4)/ (5)) "		6.8	6.8 Pass
TC13	Signed Integers with Addition (Valid Ex #12)	Shows the calculator can handle signed integers with only addition	" -(+1)+(+2) "		1	1 Pass
TC14	Signed Integers with Addition and Subtraction (Valid Ex #13)	Shows the calculator can handle signed integers with addition and subtraction	" -(-(-3))+(-4)+(+ 5) "		-2	-2 Pass
TC15	Signed Integers with Exponents (Valid Ex #14)	Shows the calculator can handle signed integers with the exponent operator	" + 2 ** (-3) "		0.125	0.125 Pass
TC16	Signed Integers with Multiplication and Division (Valid Ex #15)	Shows the calculator can handle signed integers with multiplication and division operators only	" -(+2)*(+3)-(-4)/(-5) "		-6.8	-6.8 Pass
TC17	Nested Exponents	Shows the calculator can handle nested exponents, where the order of exponents solved is in reverse. In this test case 2**3 is solved first and gives a value of 8. This makes the final expression 2**8 = 256	2 ** 2 ** 3		256	256 Pass

TC18	Exponents and parentheses	Shows the calculator follows PEMDAS for parentheses and exponents only situations. In this case the 2**2 should be handled first	(2**2)**3	64	64	Pass
TC19	Divide by 0	Shows the calculator can handle the error case where a division operation includes a 0 as part of the argument	64/0	Error: Division by zero.	Error: Division by zero.	Pass
TC20	Modulo by 0	Shows the calculator can handle the error case where a modulo operation includes a 0 as part of the argument	17%0	Error: Modulus by zero.	Error: Modulus by zero.	Pass
TC21	Too many opening parentheses	Shows the calculator can handle the error case where an imbalanced amount of open and closed parentheses occurs	(((((4*5)	Error: Too many opening parentheses	Error: too many opening parentheses	Pass

TC22	Too many closing parentheses	There are 4 more parentheses on the right side than there are on the left side. This is an imbalance in the number of opening and closing parentheses, meaning the expression is invalid.	(4*5)))))	Error: Too many closing parentheses	Error: Too many closing parentheses	Pass
TC23	Operators Without operands (Invalid Ex #2)	The * operator has no operand to the left of it, making it an invalid expression	*5+2	Cannot start expression with operator	Error: Cannot start expression with operator	Pass
TC24	Missing operator (Invalid Ex #4)	There is no operator between the 5 and opening parentheses '(', making this an invalid expression	5(2+3)	Opening parentheses after a number (implicit multiplication not allowed)	Error: Opening parentheses after a number (implicit multiplication not allowed)	Pass
TC25	Invalid Characters (Invalid Ex #5)	The '&' character is not a valid operator or operand in this calculator, making this an invalid expression	7&3	Invalid character in expression	Error: Invalid character in expression	Pass

TC26	Divide by 0/Invalid Operator Usage (Invalid Ex #7)	This expression reduces down to a division by 0 because 3*0 becomes the denominator of the division operation and 3*0 is itself 0. This produces a division by 0 error, making this expression invalid	((5+2)/(3*0))	Error: Division by zero.	Error: Division by zero.	Pass
TC27	Invalid Operator Sequence (Invalid Ex #8)	There is no operand between the - operator and the closing parentheses ')'. This means the expression is invalid	((2-)1+3)	Closing parentheses after an operator.	Error: Closing parentheses after an operator.	Pass
TC28	Missing operand (Invalid Ex #9)	There is no operand on either side of the - operator in the second sub-expression. This makes it an invalid sub-expression and therefore the entire expression is invalid	((4*2)+(-))	Closing parentheses after an operator.	Error: Closing parentheses after an operator.	Pass
TC29	Rounding Check	Shows the calculator can handle numbers with more decimal points than what is outputted. This means the calculator properly rounds off an answer to 5 decimal places	14/51	0.27451	0.27451	Pass

TC30	Invalid Characters (Invalid Ex #10)	The '^' character is not a valid operator as the proper exponent operator in this calculator is '**'. Therefore this expression is invalid	((7*3)^2)	Invalid character in expression	Error: Invalid character in expression	Pass
TC31	help command	Outputs instructions to the user on how to properly use the calculator	help	user manual	user manual	Pass
TC32	exit command	Ends the program when the user is finished using the calculator	exit	Now Exiting. Goodbye!	Now Exiting. Goodbye!	Pass
TC33	Incorrect Operator Usage (Invalid Ex #3)	Division by zero produces an undefined result, therefore this expression is invalid and an error should be output to the user	4/0	Error: Division by zero.	Error: Division by zero.	Pass
TC34	Mismatched parentheses (Invalid Ex #6)	There are more opening parentheses in this statement than closing parentheses, therefore the expression is invalid and an error output should be sent to the user	(((3+4)-2)+(1)	Error: Too many opening parentheses	Error: Too many opening parentheses	Pass

| TC35 | Unmatched parentheses (Invalid Ex #1) | There are no closing parentheses to match the opening parentheses in this expression. Therefore this is an invalid expression | 2*(4+3-1 | Error: Too many opening parentheses | Error: Too many opening parentheses | Pass |