Code Commanders

Test Case

Version 1.0

Revision History

Date	Version	Description	Author
05/02/2024	1.0	Test cases	Code Commanders

Table of Contents

Test Case

1. Purpose

The test case document provides the user with a list of possible cases for the calculator module. Every test case will include a case ID, description, the input, the expected output, actual output and the result of the case. The purpose of having test cases is that we need to ensure that the calculator module runs smoothly without bugs and with proper inputs including valid and invalid functions. Valid cases include proper use of operators, order of operations and parenthesis to close off part of a function. Invalid cases may include uninitialized variables, mismatched parenthesis, extraneous variables, and missing operands or operators.

2. Test case identifier

The test case identifiers are included in the supplemental test case table included below and are titled "TC" followed by the number. These identifiers belong to the type of case being tested, signified by their letter symbol described below.

TC#-Test Case (Number)

And each test case identifier will have some of these symbols in at least one case:

& – AND

| - OR

! - NOT

$ - XOR

@ - NAND

3. Test item

The purpose for test case is in the Description column referred in the supplemental test case excel file or the table below and provides the input condition that is being tested in every case.

4. Input specifications

Refer to the test case at the end of this document or in the supplemental excel test case document for list of input specifications. For this testing plan, it will consist of the calculator module for inputting in each function so it can be passed through strings into the module to be evaluated.

5. Output specifications

There are two types of output specifications listed in the test case table documents, which are columns called Expected Output and Actual Output. The Expected Output shows the intended output of each test case such as if an input is valid or an error message if invalid. The Actual Output shows what comes out by executing the code, either a value or an error. The last column, Result, determines if the Expected and Actual outputs match each other and if they do, it is considered a passing case. If they are different, then it will be a failure case.

6. Environmental needs

Running this test plan will require a computer with any IDE installed to run C++ programs.

Case #'s	Description	Input	Expected Output	Actual Output	Result
TC01	Basic AND	T&T	T	T	Pass
TC02	multivariable AND	T&F&T&T	F	F	Pass
TC02	XOR	(T$T)	F	F	Pass
TC03	XOR	(F$T)	T	T	Pass
TC04	NOT/OR	!T|!F	T	T	Pass
TC05	NOT/AND	!(T&T)	F	F	Pass
TC06	Complex	((T&T)&T)|F	T	T	Pass
TC07	Complex	(!F|!T) & (!T)	F	F	Pass
TC08	Complex	(((!T@T)|(T&T))@(T))	F	F	Pass
TC09	missing operands	(T|)	ERROR	ERROR	Pass
TC10	Uninitialized characters	X|Y	ERROR	ERROR	Pass
TC11	Unexpected characters	((T&T)@(T@T)True)]	ERROR	ERROR	Pass
TC12	Invalid characters/ uniniti	y|n	ERROR	ERROR	Pass
TC13	Spacing	T & F (spaced out)	F	T	Fail
TC14	Inconsistent capitalizatio	(T&F)|t	T	ERROR	Fail
TC15	Unidentified operator	T*F	ERROR	ERROR	Pass