Group 22

Arithmetic Parser

Test Case

Version 1.2

Revision History

Date	Version	Description	Author
11/18/2024	1.0	Rough draft of the document.	Janna
12/09/2024	1.1	Continued draft of whole document.	Janna
12/11/2024	1.2	Final draft	Yoseph Ephrem

Table of Contents

Test Case

1. Purpose

This Test Case Specification document for the *Arithmetic Parser* defines a test case for an item that should be tested. The sections of this document are ordered in a specific sequence: Test Case Identifier, Test Item, Input specifications, Output Specifications, Environmental Needs, Special Procedural Requirements, and Intercase Dependencies.

NOTE: for sections 2, 3, 4, and 5: It is OK to use a table like the one proposed in class, also suggested on the project part 5 description.

2. Test case identifier

The Test Case ID for the UI test case is: TC-05.

The Test Case ID for the key-input test case is: TC-06

The Test Case ID for small number handling is TC-07

The Test Case ID for complex expression handling is TC-08

The Test Case ID for a mismatched parenthesis situation is TC-09

3. Test item

TC-05: 4 * (3 + 2) % 7 – 1:

This test case includes the use of numbers on the calculator as well as the following features/operators/symbols: *, (,), +, %, and – (multiplication, parenthesis, modulo, and subtraction). These features work as specified in the Requirements Specification document and their usages are outlined in the user manual document. Both of these documents can be found in our GitHub Repository.

TC-06: (64/4)**3:

This test utilizes parenthesis to carry out PEMDAS order in the operations as well was division and powers. This case is to be tested with direct input from the keyboard but can be evaluated using the user interface buttons (“(64 ÷4)∧2”).

TC-07: 4*4/0:

This test carries out pemdas operations but comes across a error which is division by zero. The input for this case is a string by key input (“4*4/0)

TC-08: (1÷1000x(8+(96÷6+(20x5))))

This tests purpose is to show the parsers ability to calculate very complex equations using pemdas order of operations.

TC-09: (5x8-2÷(8÷4x1 or (5*8-2/(8/4*1

This tests purpose is to show the parsers capabilities for each of the input types, by buttons on display and by keyboard input.

4. Input specifications

TC-05: First, open the executable file provided. To perform this test case input the following in the presented order by pressing the corresponding button on the GUI: “4”, “x”, “(“, “3”, “+”, “2”, “)”, “%”, “7”, “-“, “1”, “=”.

TC-06: To begin open the calculator by running the executable file then type the expression “(64/4)**3” into the calculator and press “=”

TC-07: Open the calculators executable file then type in (4*4/0) into the calculator and press “=”

TC-08: Open the file and press the following buttons in the same order:
“(“,”1”,”÷”,”1000”,”x”,”(“,”8”,”+”,”(“,”9”,”6”,”÷”,”6”,”+”,”(“,”2”,”0”,”x”,”5”,”)”,”)”,”)”,”)”

TC-09: Open the calculator and enter one of the following two input types for the same question.
select the buttons in this order: “(“,”5”,”x”,”8”,”-”,”2”,”÷”,”(“,”8”,”÷”,”4”,”x”,”1”

Or enter the following directly: (5*8-2/(8/4*1

5. Output specifications

TC-05: The output specification for this test case is 5, and when executing this on the arithmetic parser, the correct output is given.

TC-06: The output for this test case is 256 and the correct value is found when using the parser.

TC-07: The output for this test case is undefined

TC-08: The output for this case is 0.124

TC-09: The outputs for this case are the following
Direct key input: nan
Gui button input: 39

6. Environmental needs

6.1.1 Hardware (nothing particular for the arithmetic expression project)

6.1.2 Software (nothing particular for the arithmetic expression project)

6.1.3 Other

There are no other requirements necessary for our project.

7. Special procedural requirements

8. Intercase dependencies