Arithmetic Expression Evaluator

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 12/03/2023 | 1.0 | Complete Document | C. Cooper |

Table of Contents

1. Purpose 4

2. Test case identifier 4

3. Input specifications 4

4. Output specifications 4

5. Environmental needs 4

# Purpose

This Test Case Specification document for the Arithmetic Expression Evaluator defines the format of the test cases that are to be used when testing the Arithmetic Expression Evaluator. The test cases themselves can be found in the Test Cases Datasheet. There are ninety test cases in total.

The goal of the test case is to discover any errors in the implementation of the requirements so that they may be fixed before shipping the final product. To this end, each test case will be tested twice, by two independent testers, to ensure that no errors are made. If the system fails any test cases, it will be updated, and the test cases will be re-evaluated to ensure the update fixed the problems without creating any new ones.

# Test case identifier

Each test case is identified with a unique identifier that conveys the aspects of the requirements that it intends to test. It additionally has a longer plain text name, and a purpose declaration which elaborates on what it intends to test.

The identifiers consist of a series of one-to-two-character codes that represent what it intends to test, and then a number to differentiate cases which test the same general aspects.

The identifier codes are:

|  |  |  |
| --- | --- | --- |
| **Character Code** | **Aspect of Requirements** | **Relevant Requirements Section** |
| A | Addition | 3.1.1.1 – Addition |
| Ab | Absolute Value | 3.1.2.2 – Absolute Value |
| B | Binary Operators | 3.1.1 – Binary Operators |
| C | Input Characters | 3.1.6 – User Interface |
| D | Division | 3.1.1.4 – Division |
| E | Exponentiation | 3.1.1.6 – Exponentiation |
| I | Input Integer Values | 3.1.6 – User Interface |
| M | Multiplication | 3.1.1.3 – Multiplication |
| Mo | Modulo | 3.1.1.5 – Modulo |
| N | Negation | 3.1.2.1 – Negation |
| O | Order if Operations | 3.1.4 – Operator Precedence |
| P | Parentheses | 3.1.3 – Parentheses |
| Ps | Parsing | 3.1.6 – User Interface |
| S | Subtraction | 3.1.1.2 – Subtraction |
| U | Unary Operators | 3.1.2 – Unary Operators |
| ! | Errors | 3.1.5 – Errors |

Examples:

PPs!-1 would be the first test case to test parentheses parsing errors.

OU-3 would be the third test case to test unary operator order of precedence.

# Input specifications

Each test case directly specifies the input to be given to the program.

# Output specifications

Each test case specifies the exact output to be given by the evaluator, except in special cases, such as when the program should terminate. In these cases, the desired behavior is specified in brackets.

# Environmental needs

All test cases shall be tested on the latest version of the evaluator. The cases may be tested on any operating system which can compile c++ code.