**Arithmetic Expression Evaluator**

**Software Requirements Specifications**

Version <1.0>

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 10/13/2023 | 1.0 | Filled out this document as a group  (Liam, Burke, Anjay, Michael) | TechFusion Innovations |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Table of Contents**

1. Introduction [4](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.gjdgxs)

1.1 Purpose [4](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.30j0zll)

1.2 Scope [4](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.1fob9te)

1.3 Definitions, Acronyms, and Abbreviations [4](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.3znysh7)

1.4 References [4](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.2et92p0)

1.5 Overview [4](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.tyjcwt)

2. Overall Description [5](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.3dy6vkm)

2.1 Product perspective [5](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.1t3h5sf)

2.1.1 System Interfaces [5](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.4d34og8)

2.1.2 User Interfaces [5](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.2s8eyo1)

2.1.3 Hardware Interfaces [5](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.17dp8vu)

2.1.4 Software Interfaces [5](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.3rdcrjn)

2.1.5 Communication Interfaces [5](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.26in1rg)

2.1.6 Memory Constraints [5](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.lnxbz9)

2.1.7 Operations [5](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.35nkun2)

2.2 Product functions [5](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.1ksv4uv)

2.3 User characteristics [5](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.44sinio)

2.4 Constraints [5](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.2jxsxqh)

2.5 Assumptions and dependencies [5](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.z337ya)

2.6 Requirements subsets [5](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.3j2qqm3)

3. Specific Requirements [5](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.1y810tw)

3.1 Functionality [5](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.4i7ojhp)

3.1.1 <Functional Requirement One> [6](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.2xcytpi)

3.2 Use-Case Specifications [6](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.1ci93xb)

3.3 Supplementary Requirements [6](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.3whwml4)

4. Classification of Functional Requirements [6](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.2bn6wsx)

5. Appendices [6](https://docs.google.com/document/d/1119Pe7ZDHT4IRM2YvPNAaXAZ1c2oBEqC3bf9MeM0RSc/edit#heading=h.qsh70q)

**Software Requirements Specifications**

# Introduction

## Purpose

## The purpose of this software Requirements Specifications for the Arithmetic Expression Evaluator is to parse and evaluate arithmetic expressions and output the correct expressions evaluated according to PEMDAS.

## Scope

## The scope of the arithmetic expression evaluator is to take in any user inputs and filter it through the program's subsystems. After this process it will output a usable correct expression evaluated according to PEMDAS.

## Definitions, Acronyms, and Abbreviations

## SDLC - Software Development Life Cycle

## API - Application Programming Interface

## SDK - Software Development Kit

## GUI - Graphical User Interface

## TDD - Test Driven Development

## DB - Database

## Derivable - Tangible or intangible item that is delivered to a client or stakeholder as part of a project.

## Sprint - A "sprint" is a set period during which specific work has to be completed and made ready for review. It's essentially a short, focused phase of development.

## Bug: An error, flaw, or unintended result in software.

## Deployment: The process of releasing a new feature, app, or update to the available infrastructure.

## Push/Pull: Sending or receiving changes to/from a remote repository in a version control system.

## Commit: A set of changes or edits saved to a version control system.

## Branch: A separate line of development created in a version control system, which can later be merged back with the main line.

## Framework: A platform for developing software applications. It provides a foundation on which software developers can build programs for a specific platform.

## References

Our Github

EECS 348: Term Project in C++

## Overview

This document contains the plan to make the Arithmetic Expression Evaluator and is organized as such.

# Overall Description

## Product perspective

### **System Interfaces**

### **User Interfaces**

### **Hardware Interfaces**

### **Software Interfaces**

### **Communication Interfaces**

### **Memory Constraints**

### **Operations**

## Product functions

## User characteristics

## Constraints

## Assumptions and dependencies

## Requirements subsets

# Specific Requirements

## Functionality

* + 1. *Handling the input from the user*

This system prompts the user for input through the console then waits for the input. Once the user inputs the string this system will format it in a way that the verifying system can evaluate it for errors.

### **Verifying the input to be complete**

This system will check the input to make sure that there is no letters or non arithmetic characters

1. *Parenthesis Handler*

Will take in the input and evaluate the parenthesis and do the required arithmetic within the parenthesis

### **Arithmetic Evaluator**

System to output the correct evaluate system

## Use-Case Specifications

User interact with command console and input arithmetic expression

**Use Case 1:**

Main Flow:

1.User inputs a simple arithmetic expression (e.g., 5 + 3).

2.The system validates the expression.

3.The system evaluates the expression.

4. The system displays the result to the user (e.g., 8).

Extensions:

2a. If the input expression is invalid, the system displays an error message.

3a. If the evaluation encounters an error (like divide by zero), the system displays an appropriate error message.

Postconditions: The correct result is displayed and the system is ready for another input.

## Supplementary Requirements

Non-functional requirements:

Ability for the program to accept extra parenthesis, use of symbols with no purpose, invalid characters, missing operator

Big O: The evaluator should return results in less than X milliseconds for standard arithmetic expressions.

Error Handling: The system should detect and handle divide-by-zero scenarios gracefully. The system should handle and inform the user of mathematical errors, such as taking the root of a negative number.

Testing: A comprehensive testing strategy should be in place, covering unit tests, integration tests, and end-to-end tests. Performance and stress testing should be conducted to ensure the system can handle peak loads.

Documentation: End-user documentation should be available, explaining how to use the arithmetic evaluator. Developer documentation should be detailed, including the system's architecture, data flow, and more.

# Classification of Functional Requirements

|  |  |
| --- | --- |
| **Functionality** | **Type** |
| Handling the input from the user | Essential |
| Verifying the input to be complete | Essential |
| Parenthesis Handler | Desirable |
| Arithmetic Evaluator | Essential |

# Appendices

None to note.

# Appendix A: Glossary of Terms:

Arithmetic Expression: A mathematical sentence consisting of two numbers, a mathematical operation, and a result.

**Appendix B: Error Messages and Troubleshooting Guide**

**Appendix C: Credits/ Acknowledgements**