
Math Olympiad

**Arithmetic Expression Evaluator
Software Requirements Specifications**

Version <2.0>

Arithmetic Expression Evaluator	Version: <2.0>
Software Requirements Specifications	Date: <10/20/24>
artifacts/02-Software-Requirements-Spec	

Revision History

Date	Version	Description	Author
<10/11/24>	<1.0>	<Create document and assign portions>	<Whole group>
<10/18/24>	<2.0>	<Revised and compiled document>	<Whole group>

Arithmetic Expression Evaluator	Version: <2.0>
Software Requirements Specifications	Date: <10/20/24>
artifacts/02-Software-Requirements-Spec	

Table of Contents

1.	Introduction	4
1.1	Purpose	4
1.2	Scope	4
1.3	Definitions, Acronyms, and Abbreviations	4
1.4	References	4
1.5	Overview	4
2.	Overall Description	5
2.1	Product perspective	5
2.1.1	System Interfaces	5
2.1.2	User Interfaces	5
2.1.3	Hardware Interfaces	5
2.1.4	Software Interfaces	5
2.1.5	Communication Interfaces	5
2.1.6	Memory Constraints	5
2.1.7	Operations	5
2.2	Product functions	5
2.3	User characteristics	5
2.4	Constraints	5
2.5	Assumptions and dependencies	5
2.6	Requirements subsets	5
3.	Specific Requirements	5
3.1	Functionality	5
3.1.1	Understand and Parse User Input	5
3.1.2	Understand and Process Parentheses	5
3.1.3	Follow the Order of Operations	5
3.1.4	Display Calculation Output or Error	6
3.1.5	Use Previous Calculations as Variables	6
3.2	Use-Case Specifications	6
3.3	Supplementary Requirements	6
4.	Classification of Functional Requirements	6
5.	Appendices	6

Arithmetic Expression Evaluator	Version: <2.0>
Software Requirements Specifications	Date: <10/20/24>
artifacts/02-Software-Requirements-Spec	

Software Requirements Specifications

1. Introduction

The Software Requirements Specifications Document (SRS) contains specifications for the use cases and constraints placed upon the final product (the Arithmetic Expression Evaluator or AEE). These include an overall description of the AEE, an outline of specific requirements including use-case modeling, and classification of functional requirements based on vitality.

1.1 Purpose

The purpose of this document is to organize and prioritize specific goals and use-cases with regards to the AEE.

1.2 Scope

This document applies to the AEE project, which is a program designed to simplify arithmetic expressions for use in a future compiler. The scope of this document does not extend to the future compiler, the specifics of integration into such a compiler, or the production processes involved in either. The scope remains firmly attached to the use cases and broad-stroke concepts regarding the functions and requirements of the AEE. The scope does not extend to the specific programming practices or script structure, only to the requirements of its use.

1.3 Definitions, Acronyms, and Abbreviations

AEE: Arithmetic Expression Evaluator

SRS: Software Requirements Specifications Document

1.4 References

Use Case Diagram: See section 3.2 and the artifacts folder on GitHub.

1.5 Overview

This document is split into three sections after the introduction, consisting of an overall description of the factors and requirements regarding the AEE, a deep-dive into specific design requirements, and a classification of these requirements based on how necessary they are to the final product.

Arithmetic Expression Evaluator	Version: <2.0>
Software Requirements Specifications	Date: <10/20/24>
artifacts/02-Software-Requirements-Spec	

2. Overall Description

2.1 Product perspective

2.1.1 User Interfaces

The software will include an intuitive User Interface consisting of menus, panels, buttons, text boxes, etc., to allow the user to interact with the software.

2.1.2 Software Interfaces

The software will load calculator history from a CSV file. The software will require an API to read the data from the CSV file.

2.1.3 Communication Interfaces

The software will interface with the end user via a text-based interactive command line.

2.1.4 Memory Constraints

The software will require enough memory to evaluate arithmetic expressions, which will depend on the size of the expression.

2.2 Product functions

The software will act as a calculator and evaluate functions adhering to PEMDAS. It will be able to perform addition, subtraction, multiplication, division, exponentials, and be compatible with parenthesis.

2.3 User characteristics

The user will be responsible for interacting with the program / calculator. They will determine what expressions are evaluated.

2.4 Constraints

The software will be constrained / limited by the size and complexity of the expressions it is provided / asked to complete.

2.5 Assumptions and dependencies

This program is being designed for personal computers, therefore we are assuming there is plenty of memory and processing power.

There needs to be a way to interface with the command line.

The user input must have valid syntax, otherwise an error will be displayed.

3. Specific Requirements

3.1 Functionality

All listed subsections contain the primary requirement along with the member primarily responsible. The team member(s) listed are not the only ones solely responsible, but will often take the lead on addressing the requirement.

3.1.1 Understand and Parse User Input - [UI/UX + Technical]

The program must be able to take in a set of numbers and symbols from the user. After taking in the input, the program must be able to pass the requisite information along to different parts of the program in order to compute the result.

3.1.2 Understand and Process Parentheses - [Technical]

The program must be able to properly understand the order of operations with regards to parentheses. Any expression contained within parentheses must be evaluated first, while still following the order of operations within the parentheses. Additionally, the program must be able to support nested parentheses.

3.1.3 Follow the Order of Operations - [Technical]

The program must be able to accurately and consistently follow the order of operations. This requirement

Arithmetic Expression Evaluator	Version: <2.0>
Software Requirements Specifications	Date: <10/20/24>
artifacts/02-Software-Requirements-Spec	

must go hand in hand with the handling of parentheses, as that will necessitate additional checks to ensure the proper order is being followed.

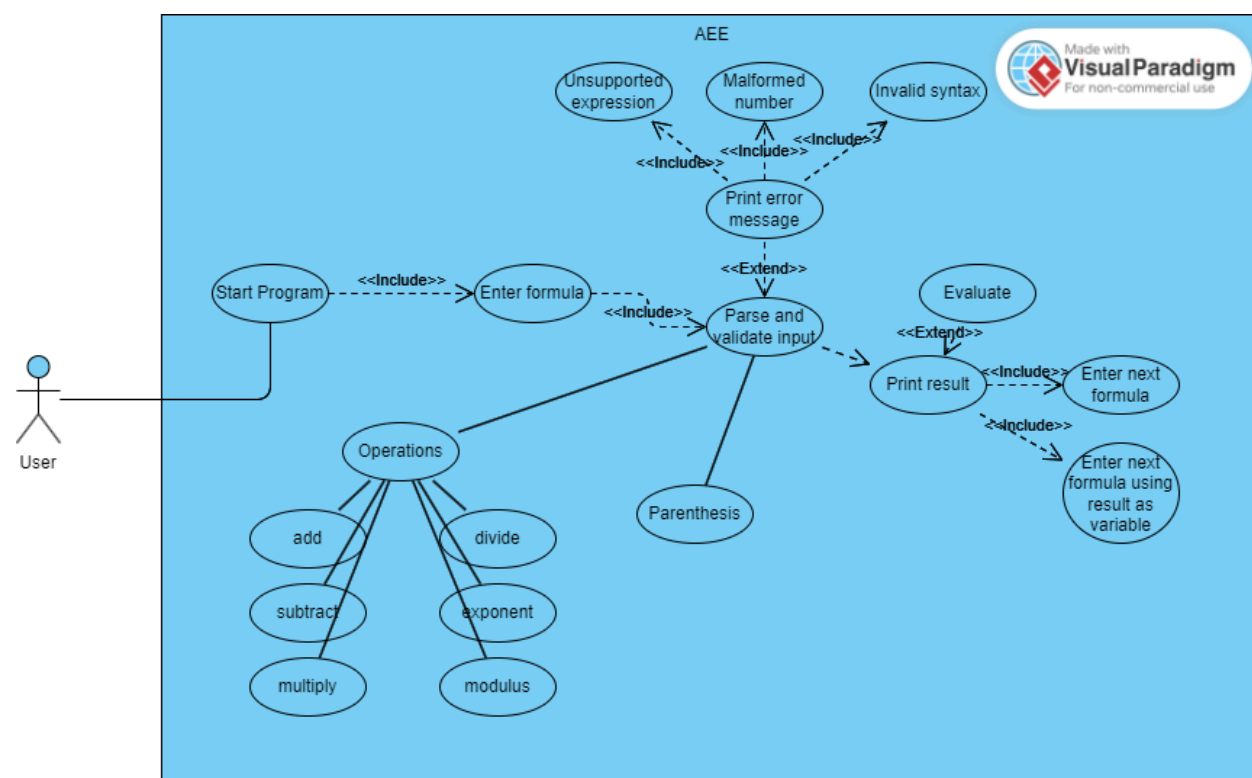
3.1.4 Display Calculation Output or Error - [UI/UX + Technical]

The program must be able to properly display either the result of the inputted equation or a valid and helpful error message. Any error message displayed by the application must be specific so that the end user can resolve the issue.

3.1.5 Use Previous Calculations as Variables - [?]

Optional: Allow previously calculated answers to be used in future calculations as variables (i.e. x). If this functionality is to be implemented, a more specific implementation must be chosen. Will the program only be able to remember the previous answer, or will you be able to store an arbitrary amount of previously calculated answers to use in your future calculations?

3.2 Use-Case Specifications



3.3 Supplementary Requirements

- The program must be written in C++.
- The structure of the code must allow for new functionality to be implemented with minimal editing of the existing codebase.
- The code must be clearly documented to facilitate future development and debugging.
- The program must not take more than several seconds to complete.
- The user interface must be intuitive and readable.
- The program must be optimized to avoid excessive memory usage.

4. Classification of Functional Requirements

Arithmetic Expression Evaluator	Version: <2.0>
Software Requirements Specifications	Date: <10/20/24>
artifacts/02-Software-Requirements-Spec	

Functionality	Type
The program must support and correctly parse user input.	Essential
The system must support parenthesis and correctly evaluate expressions containing them.	Essential
The program must be able to follow PEMDAS order of operations	Essential
The program must be able to output either the result or a helpful error message.	Essential
The program can allow the user to incorporate the previous result into the next operation via variable.	Optional

5. Appendices

- *02a-Use-Case-Diagram - See artifacts/ folder for document. Also included as section 3.2 in SRS.*