

---

**EECS 348 Group 14**

---

**Arithmetic Expression Evaluator in C++  
Software Requirements Specifications**

**Version <1.0>**

Arithmetic Expression Evaluator in C++	Version: 1.0
Software Requirements Specifications	Date: 28/09/23
upedu srs	

## Revision History

Date	Version	Description	Author
28/09/23	1.0	First draft	Vincent Dick

Arithmetic Expression Evaluator in C++	Version: 1.0
Software Requirements Specifications	Date: 28/09/23
upedu_srs	

## 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	<Functional Requirement One>		6
3.2	Use-Case Specifications	6	
3.3	Supplementary Requirements	6	
4.	Classification of Functional Requirements	6	
5.	Appendices	6	

Arithmetic Expression Evaluator in C++	Version: 1.0
Software Requirements Specifications	Date: 28/09/23
upedu_srs	

# Software Requirements Specifications

## 1. Introduction

### 1.1 Purpose

The purpose of the SRS is to detail all of the requirements needed in our product to produce a functional final product. In doing this, we will outline the desired behaviors of our calculator. We will also provide all of the nonfunctional requirements, the constraints, and other factors we run into when working on this project.

### 1.2 Scope

This document applies to the AEEC, which serves as a calculator that takes in an arithmetic expression and evaluates it while following PEMDAS rules.

### 1.3 Definitions, Acronyms, and Abbreviations

- AEEC (Arithmetic Expression Evaluator in C++)
- PEMDAS (Parentheses, exponents, multiplication, division, addition, subtraction)
- SRS (Software Requirements Specification)

### 1.4 References

N/A

### 1.5 Overview

The rest of this document contains an overall description of the AEEC software, providing general information about the software and its background (section 2), as well as the system requirements and functionality of the software (section 3).

## 2. Overall Description

### 2.1 Product perspective

The evaluation of arithmetic expressions is an integral part of many academic, personal, and professional pursuits. Evaluating arithmetic expressions by hand may be time consuming and error prone. A tool such as the AEEC can decrease calculation times and improve the precision of results.

#### 2.1.1 System Interfaces

The AEEC is an independent system consisting of two components: the expression interface and the evaluation engine. The expression interface receives input and passes it to the evaluation engine. The evaluation engine performs the specified operations on the given operands and returns the result to the expression interface. The expression interface then displays the result in a readable format.

#### 2.1.2 User Interfaces

The expression interface provides a user interface through the terminal/command line of the host machine.

#### 2.1.3 Hardware Interfaces

The AEEC will execute on a general purpose computer.

#### 2.1.4 Software Interfaces

The AEEC will be an independent C++ program; the only software interfaces necessary are a C++ compiler and the C++ standard library.

#### 2.1.5 Communication Interfaces

The AEEC requires no communication interfaces.

#### 2.1.6 Memory Constraints

The memory constraints of the AEEC will be determined by the RAM of the host machine.

Arithmetic Expression Evaluator in C++	Version: 1.0
Software Requirements Specifications	Date: 28/09/23
upedu srs	

### 2.1.7 Operations

The operations of the AEEC will follow the established and intuitive style of modern calculators. This means that the AEEC will receive expressions written in a conventional format, evaluate them using PEMDAS ordering, and output the results in a conventional format.

## 2.2 Product functions

The main function of the AEEC is evaluating arithmetic expressions using PEMDAS ordering.

An arithmetic expression is an unsolved series of operators and operands (numbers and symbols) that follows conventional mathematical style. The AEEC will receive these arithmetic expressions through the expression interface.

PEMDAS ordering defines the precedence of operations from highest precedence to lowest precedence: parentheses, exponents, multiplication, division, addition, and subtraction. The evaluation of an arithmetic expression consists of resolving each operation with the given operands until a single numerical result is achieved. The AEEC performs the evaluation of expressions in the evaluation engine.

The final function of the AEEC is displaying the result of the expression through the expression interface.

## 2.3 User characteristics

Users are anyone who requires the quick and precise evaluation of an arithmetic expression. Users do not need to be knowledgeable of the AEEC itself, but must understand the input expression and output result to use the AEEC effectively.

## 2.4 Constraints

The AEEC should be able to handle invalid inputs and ensure the output of precise results as quickly as possible.

## 2.5 Assumptions and dependencies

There are no specific assumptions regarding the AEEC. Dependencies of the AEEC include a C++ compiler, the C++ standard library, and a host machine.

## 2.6 Requirements subsets

The requirements subsets are software requirements (detailed in 2.1, 2.2, 2.4, and 2.5) and user requirements (detailed in 2.3).

# 3. Specific Requirements

## 3.1 Functionality

### 3.1.1 Expression Interface

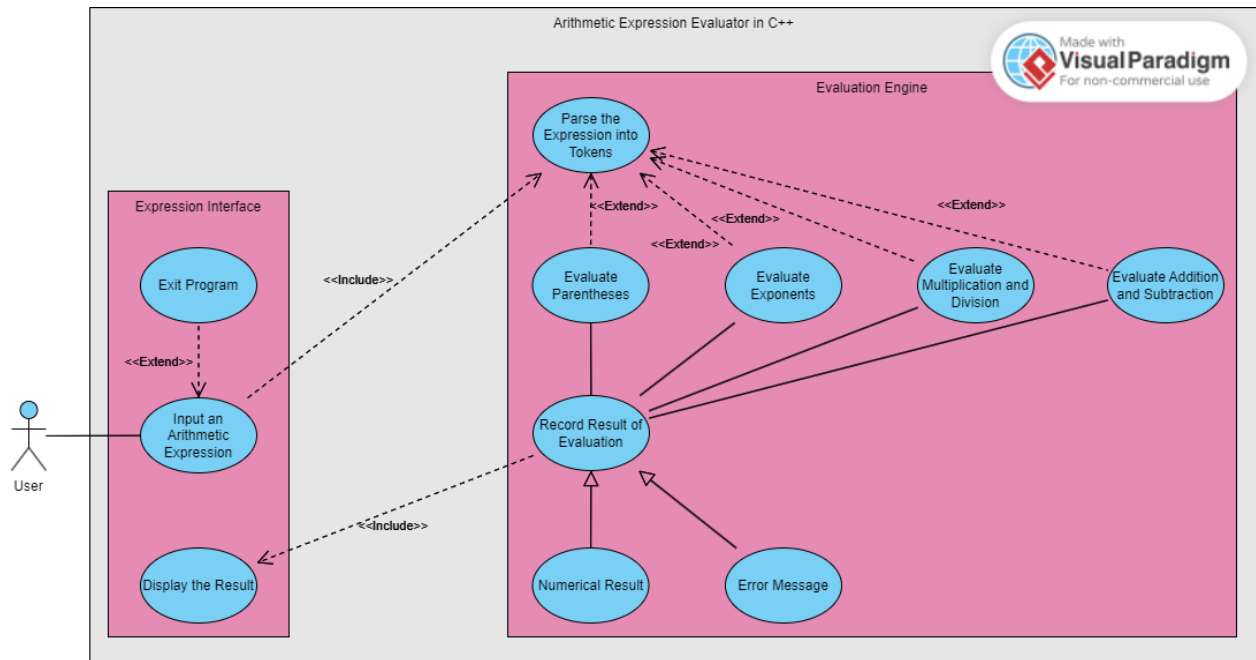
- 3.1.1.1 Prompts the user to enter an arithmetic expression through the terminal/command line.
- 3.1.1.2 Accepts a string representing a valid arithmetic expression containing operators and operands.
- 3.1.1.3 Sends the arithmetic expression to the evaluation engine and waits for a response.
- 3.1.1.4 Receives a response from the evaluation engine representing the result of the expression or an error.

### 3.1.2 Evaluation Engine

- 3.1.2.1 Accepts a string representing a valid arithmetic expression from the expression interface.
- 3.1.2.2 Parses input into tokens of two categories: operators and operands.
- 3.1.2.3 Evaluates the tokens in PEMDAS order to achieve a final result.
- 3.1.2.4 Sends the result of the evaluation to the expression interface, whether it be a number or error message.

Arithmetic Expression Evaluator in C++	Version: 1.0
Software Requirements Specifications	Date: 28/09/23
upedu_srs	

### 3.2 Use-Case Specifications



### 3.3 Supplementary Requirements

Performance: perform PEMDAS operations within a reasonable response time based on the complexity of the PEMDAS expression (i.e. parenthesis nests).

Usability: have a clear and concise user interface, and the errors that were found are to be user-friendly and informative as of why there is an error.

Accuracy: floating point numbers are precise to 10 decimal places; arbitrary variables are not accepted.

Development constraints: coded in the C++ programming language.

OS compatibility: capable of running in any OS (i.e. Linux, Mac, Windows)

## 4. Classification of Functional Requirements

Functionality	Type
3.1.1.1 Prompts the user to enter an arithmetic expression through the terminal/command line.	Essential
3.1.1.2 Accepts a string representing a valid arithmetic expression containing operators and operands.	Essential
3.1.1.3 Sends the arithmetic expression to the evaluation engine and waits for a response.	Essential
3.1.1.4 Receives a response from the evaluation engine representing the result of the expression or an error.	Essential
3.1.2.1 Accepts a string representing a valid arithmetic expression from the expression interface.	Essential
3.1.2.2 Parses input into tokens of two categories: operators and operands.	Essential

Arithmetic Expression Evaluator in C++	Version: 1.0
Software Requirements Specifications	Date: 28/09/23
upedu srs	

3.1.2.3 Evaluates the tokens in PEMDAS order to achieve a final result.	Essential
3.1.2.4 Sends the result of the evaluation to the expression interface, whether it be a number or error message.	Essential

## 5. Appendices

N/A