

---

**<Company Name>**

---

**<EECS 348 - Calculator Application>  
Software Requirements Specifications**

...

**Version <1.0>**

EECS 348 – Calculator Application	Version: <1.0>
Software Requirements Specifications	Date: 10/03/24
<document identifier>	

## Revision History

Date	Version	Description	Author
10/03/24	1.0	Identifying and setting expectations for the desired behaviors of the application.	<Sione, Jett, K, Meg, Dakota, Mohamed, Ash>
10/16/24	1.0	Added finishing touches to all of the sections and agreed upon finalization of the document.	<Sione, Jett, K, Meg, Dakota, Mohamed, Ash>

EECS 348 – Calculator Application	Version: <1.0>
Software Requirements Specifications	Date: 10/03/24
<document identifier>	

# 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	4
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	6
3.1.1	<Functional Requirement One>	6
3.2	Use-Case Specifications	6
3.3	Supplementary Requirements	6
4.	Classification of Functional Requirements	7
5.	Appendices	7

EECS 348 – Calculator Application	Version: <1.0>
Software Requirements Specifications	Date: 10/03/24
<document identifier>	

# Software Requirements Specifications

## 1. Introduction

### 1.1 Purpose

*The purpose of this SRS is to define the functional and non-functional requirements of the calculator application. This document will describe the complete behavior of the external system, including its interfaces, constraints, and operations that are stated in the project plan. It will also serve as a reference for future design, implementation, and testing phases.*

### 1.2 Scope

*This SRS applies to the calculator software application, which will include the following: expressions parsing, operator support, parenthesis handling, error handling, and numeric constants. Moreover, the SRS will cover non-functional features such as user interface, usability, and performance. This document outlines the requirements for both user interface interactions and backend processing.*

### 1.3 Definitions, Acronyms, and Abbreviations

PEMDAS – Parentheses, Exponents, Multiplication, Division, Addition, and Subtraction

### 1.4 References

IEEE830-1998 (*Recommended Practice for Software*)

UPEDU Software Requirements Specification (SRS) Template

C++ programming language

### 1.5 Overview

The following section, section 2, provides an overall description and the it's requirements. Section 3 contains functional requirements and non-functional requirements including use case and constraints.

## 2. Overall Description

**For this project we are making a calculator programmed through C++ that can handle most arithmetic operations, hierarchal operations through PEMDAS, a user-friendly interface, and error handling.**

EECS 348 – Calculator Application	Version: <1.0>
Software Requirements Specifications	Date: 10/03/24
<document identifier>	

## 2.1 Product perspective

### 2.1.1 User Interfaces

*The command line interface will be providing the user where they can input expressions. The system will display the results.*

### 2.1.2 Software Interfaces

*The calculator will be made through C++ libraries for parsing expressions and doing calculations, while also making sure of the PEMDAS rule.*

### 2.1.3 Memory Constraints

Memory constraints will be handled by having an optimized data structure and controlling the recursion to handle large but not too large expressions. The calculator should have a limited expression length.

## 2.2 Product functions

The functions required for this project are the +, -, \*, /, %, and \*\*. We'll need to be able to calculate numeric constants. It must be able to handle grouping and have a hierarchy of arithmetic operations. It also must handle errors.

## 2.3 User characteristics

The calculator will be used by people who have basic knowledge of arithmetic operators.

## 2.4 Constraints

The system will be developed using C++. Efficient recursion to prevent stack overflow.

## 2.5 Assumptions and dependencies

Assumption:

- User: it is assumed that the user is familiar with basic arithmetic operations.
- System: The arithmetic expression evaluator was developed to be run on a system with compilers.
- The format of Input: It will be assumed that all user input will be valid. If the user inputs an invalid expression, the application will handle the Input in the proper way according to the error.

Dependencies:

- Compiler: Our application is dependent on compilers that support C++.

## 3. Specific Requirements

Functional requirements:

1. The system must accept any arithmetic expressions for an input, and handling the operators (+, -, \*, /, %, \*\*).
2. It must handle PEMDAS which is the order of operations that stands for parenthesis, exponents, multiplication, division, addition, and subtraction.
3. The system should recognize integer and float.
4. There must be an input for the user and an output which displays the results of the input
5. The system must handle errors and give proper messages to what's wrong with the input. For example: dividing by 0, unmatched parenthesis, invalid characters, etc.

Non-functional requirements:

EECS 348 – Calculator Application	Version: <1.0>
Software Requirements Specifications	Date: 10/03/24
<document identifier>	

1. The system must be able to process the input and return a result in under 2 seconds.
2. The system should be instinctive to have the users provide any input with ease.
3. The system should be able to handle any big inputs and avoid crashing
4. The code should be easy to read and understand, so the it can be modified in the future.

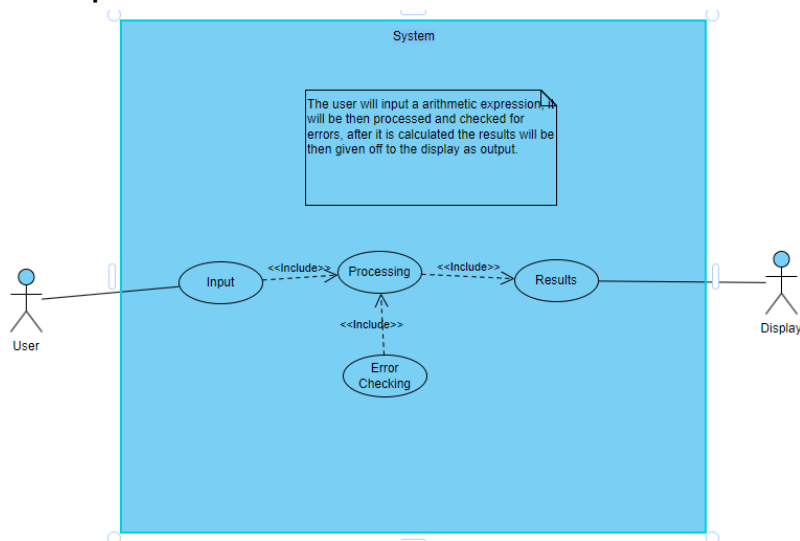
### 3.1 Functionality

The software will be able to do fundamental functionalities that are needed for parsing and calculating arithmetic expressions. It will interpret the inputted expression, such as addition, subtraction, multiplication, division, modulo, and exponentiation. The software will also handle arithmetic expressions with parentheses efficiently, ensuring that they are processed correctly to maintain the established order of operation.

#### 3.1.1 <Functional Requirement One>

The system must accept user input in the form of arithmetic expressions and handle the following operators: addition(+), subtraction(-), multiplication(\*), division(/), modulo(%), and exponentiation(\*\*). It must correctly interpret and apply the order of operations based on the PEMDAS rule, ensuring that parentheses, exponents, and the hierarchy of arithmetic operations are processed accurately. The system will accept both integer and floating-point numbers, enabling it to handle a wide a range of numeric inputs. Additionally, the interface will allow users to input expressions easily, and the system will process and display the results promptly. Errors such as division by zero, mismatched parentheses, and invalid characters must be properly detected and result informative error messages.

### 3.2 Use-Case Specifications



### 3.3 Supplementary Requirements

#### Performance/Usability Requirements

- able to handle large numbers/ equations without significant delay
- quickly respond to user input at all stages
- have a clear and use-friendly command-line interface
- error messages are informative (ex. “unfinished parenthesis”)
- program should allow for consecutive calculations without restarting

#### Development Constraints

- program must be implemented in C(+)
- program should only use standard libraries with no external aid
- code should be well documented for easy editing and navigation

EECS 348 – Calculator Application	Version: <1.0>
Software Requirements Specifications	Date: 10/03/24
<document identifier>	

#### 4. Classification of Functional Requirements

Functionality	Type
Addition +	Essential
Subtraction -	Essential
Multiplication *	Essential
Division /	Essential
Modulo %	Essential
Exponentiation **	Essential
PEMDAS	Desirable
Parentheses	Desirable

#### 5. Appendices

There are no appendices included in this document