348 Industries

EECS 348 Fall 2023 Team Project: Arithmetic Evaluation Program Software Requirements Specifications

Version <1.0>

Revision History

Date	Version	Description	Author
10/10/2023	1.0	First Iteration of the software development requirements specifications document for the EECS 348 Fall 2023 Team Project. This document will highlight all function and systemic requirements for the program.	Dev Patel Jay Chung Ruth Higgason Colin Marett

Table of Contents

Software Requirements Specifications

1. Introduction

The Arithmetic Evaluation Program is a software application that performs complex mathematical calculations. The system shall be able to perform the following operations: addition, subtraction, multiplication, division, exponentiation, and square root. The system shall also be able to handle complex numbers as well as understand rules of operations, operator precedence, parenthesis usage and intercept errors. It shall be designed to be user-friendly and easy to use. The user interface shall be intuitive and allow the user to easily enter and edit mathematical expressions. The system shall be implemented using a modern programming language, C++, and shall be compatible with the latest version of the operator systems. The system shall also be well-documented and include a user manual.

1.1 Purpose

The primary objective of the advanced arithmetic calculator program is to provide users with a comprehensive and efficient solution for performing complex mathematical calculations. The program is designed to handle a wide range of arithmetic operations, including addition, subtraction, multiplication, and division. Additionally, the program incorporates advanced mathematical functions, such as parenthesis handling, precedence understanding and error interception. By offering a user-friendly interface and seamless integration with other software applications, the advanced arithmetic calculator program aims to streamline the process of conducting mathematical computations and enhance the overall productivity of its users.

1.2 Scope

The Scope for the advanced arithmetic calculator program includes the following features and functionalities:

1. Basic Arithmetic Operations: The program must support the execution of fundamental arithmetic operations, such as addition, subtraction, multiplication, and division.
2. Advanced Mathematical Functions: The advanced arithmetic calculator program should also provide support for advanced mathematical functions, such as logarithms, exponentiation, and trigonometric operations.
3. User-Friendly Interface: The program must offer a user-friendly interface that allows users to easily input their calculations and view the results.
4. Error Handling and Validation: The advanced arithmetic calculator program should incorporate robust error handling and validation mechanisms to ensure the accuracy and reliability of the calculations.
5. Compatibility with Other Software Applications: The program should be designed to seamlessly integrate with other software applications and platforms, thereby enhancing the overall productivity and efficiency of its users.
6. Performance and Scalability: The advanced arithmetic calculator program must demonstrate high performance and scalability, ensuring that it can efficiently handle complex calculations and large datasets.

1.3 Definitions, Acronyms, and Abbreviations

SRS - Software Requirements Specification

C++ - Coding Language used to construct the program

PEMDAS - order of precedence - Parentheses, Exponents, Multiplication, Division,

Addition and Subtraction. This describes the arbitrary order in which complex equations are to be computed.

1.4 References

N/A

1.5 Overview

The following will include a brief overview of the overall project, including a description, the individual product perspectives, constraints, functions and characteristics. In section 3 we will elaborate the specific requirements pertaining to the overall system and discuss functional requirements, use cases and user needs. Finally section 4 will give a graphical interpretation of each functional requirement.

2. Overall Description

This section of the SRS describes the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides a background for those requirements, which are defined in detail in Section 3, and makes them easier to understand. Include such items as:

2.1 Product perspective

2.1.1 System Interfaces

N/A

2.1.2 User Interfaces

The user interface for this product will be the command line, where the user will enter input and receive output.

2.1.3 Hardware Interfaces

This product will run on any hardware which has a C++ compiler.

2.1.4 Software Interfaces

The software language used in this project will be C++. The code will be shared via Github.

2.1.5 Communication Interfaces

N/A

2.1.6 Memory Constraints

The size of the program will determine the memory constraints, but it will most likely be under 1MB.

2.1.7 Operations

It will perform basic arithmetic operations in a user-friendly way, such that it incorporates an understanding of PEMDAS, parenthesis handling, error handling, etc.

2.2 Product functions

The product will be an expression parser that will determine if an expression is valid and, if so, will evaluate it.

2.3 User characteristics

N/A

2.4 Constraints

The product cannot run on software that does not have a C++ compiler or hardware that does not support the C++ language.

2.5 Assumptions and dependencies

N/A

2.6 Requirements subsets

N/A

3. Specific Requirements

In this section, we provide detailed software requirements for a calculator or similar program to be developed in C++. The decision between developing a web-based or application-based system will be determined later. Currently, our primary focus is to ensure the functionality of the program.

3.1 Functionality

This section outlines the functional requirements of the program to be developed in C++. Since the choice between a web or application platform has not been finalized, we define the program's functionality independently. Any necessary adjustments related to the chosen platform will be made at a later stage.

3.1.1 <Functional Requirement One>

Users should be able to input numbers and operators, and the program should perform calculations and display results.

3.2 Use-Case Specifications

Given that the specific approach (web or application) is yet to be decided, we prioritize detailing the functional requirements over specifying use-case modeling.

3.3 Supplementary Requirements

This section accommodates other essential requirements, such as performance, user interface, security, non-functional requirements, and development constraints. As the choice of platform is currently pending, this section remains flexible for future adjustments.

4. Classification of Functional Requirements

Type = Essential, Desirable, Optional

Order by type

Functionality	Type
Expression Parsing: create a data structure to represent the parsed input expression’s structure	Essential
Operator Precedence: Implement the logic of precedence according to PEMDAS to the equation	Essential
Parenthesis Handling: Function to identify and evaluate expressions in parenthesis	Essential
Numeric Constants: Recognising numeric constants in the input	Essential
User Interface: making a user-friendly and legible interface for the user to input expressions and show the calculated results	Essential
Error Handling: Robust error handling to handle things like division by zero or invalid expressions	Essential

5. Appendices

Appendix A: Glossary of Terms - Defines key terminology used in the project.

Appendix B: Project Management Plan – Details project scheduling, Risk management and Configuration management strategies.

Appendix C: Requirement Document – Lists functional and non – functional requirements.

Appendix D: Design Document – Describes the project design, data structures and algorithms used.

Appendix E: Test Cases and Plans – Contains test cases and plans to validate the program's functionality.

Appendix F: User Manual – Provides user instructions and examples.

Appendix G: Code Documentation – Includes in code comments for guidance.

Appendix H: Change Requests – Tracks future changes and modifications.

Appendix I: Acknowledgments – Recognizes contribution and support.

Appendix J: References – A list of references and sources used during the project development process.