C++ Calculator

Software Requirements Specifications

Version <1.1>

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 10/3/2023 | 1.0 | Completed Sections 1, 2, most of 3, 4, and 5 | Carson Abbott  Owen Deines  Jordan Mcdaniels  Connor Schroeder |
| 10/14/2023 | 1.1 | Finished Section 3. Applied finishing touches | Jordan Mcdaniels  Carson Abbott |
|  |  |  |  |
|  |  |  |  |

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 5

2. Overall Description 6

2.1 Product perspective 6

2.1.1 System Interfaces 6

2.1.2 User Interfaces 6

2.1.3 Hardware Interfaces 6

2.1.4 Software Interfaces 6

2.1.5 Communication Interfaces 6

2.1.6 Memory Constraints 6

2.1.7 Operations 6

2.2 Product functions 6

2.3 User characteristics 6

2.4 Constraints 6

2.5 Assumptions and dependencies 6

2.6 Requirements subsets 6

3.Specific Requirements 7

3.1 Functionality 7

3.1.1 Input Reception 7

3.1.2 Char Error Handling 7

3.1.3 Math Error Handling 7

3.1.4 Operation Computation 7

3.1.5 Parentheses Stack 7

3.1.6 Displaying Results 7

3.2 Use-Case Specifications 7

3.3 Supplementary Requirements 7

4. Classification of Functional Requirements 8

5. Appendices 9

Software Requirements Specifications

# Introduction

This Software Requirements Specifications (SRS) document will give an overview of the C++ Calculator and its design process. The C++ Calculator will be a simple calculator that takes in mathematical expressions and outputs the result all through the command line. This SRS also outlines the many different project deliverables that will need to be completed in order to make the C++ Calculator functional (i.e, addition, order of operations).

## Purpose

The purpose of this SRS is to explain the functionality of a simple calculator which will run on the command line. The program takes a basic algebraic expression as a string and prints its result following calculation. The program must be able to calculate expressions with parentheses, exponents, multiplication, division, addition, and subtraction. Additionally, the program must be finished by the end of the semester.

## Scope

The User should be able to insert an algebraic expression into their command line.

The User should get the value of the algebraic expression back.

## Definitions, Acronyms, and Abbreviations

VCS - Version Control System.

Deliverables - program functionalities.

PEMDAS - Order of Operations: Parentheses → Exponents → Multiplication/Division/Modulo → Addition/Subtraction.

## References

|  |  |  |  |
| --- | --- | --- | --- |
| **Title** | **Report Number** | **Date** | **Publishing Organization** |
| Project Management Plan | 1 | 10/3/2023 | Rock Chalk Instruments |
|  |  |  |  |
|  |  |  |  |

## Overview

The rest of this document describes the C++ calculator’s functionalities, characteristics, requirements, constraints, assumptions, dependencies, and use-case specifications. Most of this will be organized into an outlined format, while the deliverables will be displayed using a table.

# Overall Description

## Product perspective

The goal of this product is to make it easier for the consumer to calculate mathematical expressions. This saves the user both time and effort.

### System Interfaces

N/A

### User Interfaces

The user will interact with the product entirely through a Command Line Interface.

### Hardware Interfaces

N/A

### Software Interfaces

The product will be programmed in C.

### Communication Interfaces

N/A

### Memory Constraints

The product will be constrained by the amount of memory available on the user’s system.

### Operations

N/A

## Product functions

The main product function is to return the answer to a given mathematical expression.

## User characteristics

Users should be able to download the C++ Calculator on whatever OS they’re using and run it without issues.

## Constraints

Constraints for this project include a two and half month time limit and that all programming must be done in C++.

## Assumptions and dependencies

It is assumed that all project members will be available to meet in-person, at a time that works with everyone’s schedule (Appendix A). In order for this project to be completed efficiently and on time, this project is dependent on GitHub and its underlying VCS.

## Requirements subsets

The two main categories that requirements will fall into will be the deliverables necessary for the user to input and the deliverables necessary for the Calculator to output.

# Specific Requirements

See Appendix B for a Use Case Model Diagram.

## Functionality

### Input Reception

The program will scan what the user inputs and interpret it as a string.

### Char Error Handling

The program will throw an error if the user inputs any characters besides rational numbers, "(", ")", "^", "\*", "/", "+", "-", or empty space.

3.1.3 *Math Error Handling*

The program will throw an error if the inputted expression is not computable (dividing by 0, improper parenthesis placement, no numbers between operators, etc.).

3.1.4 *Operation Computation*

*The program will look for an operator – +, -, \*, /, ^, or % – in the expression, take the chunk containing the operator surrounded by two numbers, perform the operation on the chunk, and replace the chunk from the expression with the result.*

*3.1.5 Parentheses Stack*

When the program initially reads the inputted expression for parentheses, the indexes of all opening parentheses will be added to a stack, and when a closing parenthesis is detected, the slice from the most recent opening parenthesis to the detected closing parenthesis will be taken and computed.

*3.1.6 Displaying Results*

When the program has finished calculations and no longer detects any operators, a message will be printed to the command line saying "The final result is: %f", where %f is the result of the expression as a rational number.

## Use-Case Specifications

* When the “+” operator is present an addition operation will occur.
* When the “-” operator is present a subtraction operation will occur.
* When the “\*” operator is present a multiplication operation will occur.
* When the “/” operator is present a division operation will occur.
* When the “%” operator is present a modulo operation will occur
* When the “^” operator is present a exponentiation operation will occur
* Input equations will follow the rules of PEMDAS

## Supplementary Requirements

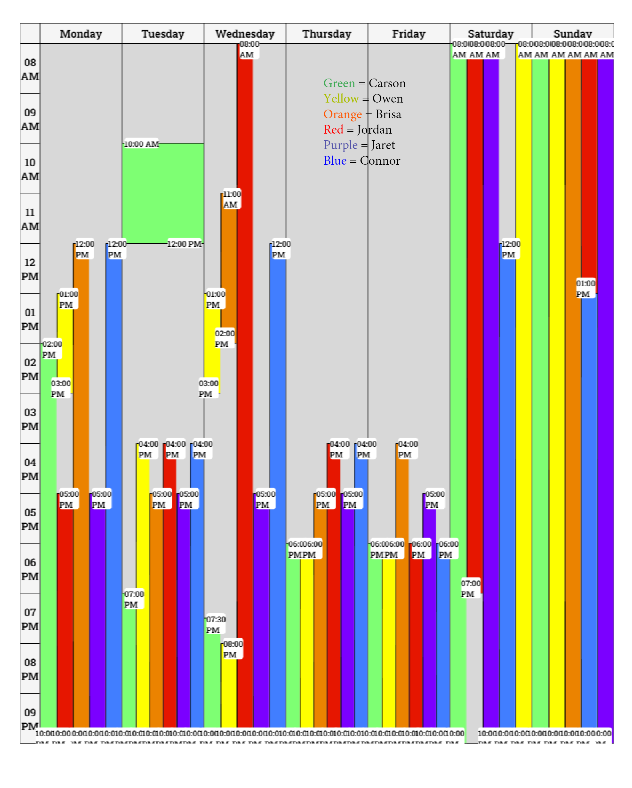
* Can be used on all operating systems
* The application must run in a timely fashion
* Must be done by due date
* N/A Costs
* \*\*

# Classification of Functional Requirements

|  |  |
| --- | --- |
| **Functionality** | **Type** |
| <Input Reception> | Essential |
| <Char Error Handling> | Desirable |
| <Math Error Handling> | Desirable |
| <Operation Computation> | Essential |
| <Parentheses Stack> | Essential |
| <Displaying Results> | Essential |

# Appendices

# Appendix A: Team Schedules



# Appendix B: Use Case Model

