
KU EECS 348: Software Engineering 1

Code Commanders Software Architecture Document

Version 1.0

Code Commanders	Version: 1.0
Software Architecture Document	Date: 04/14/2024

Revision History

Date	Version	Description	Author
04/14/2024	1.0	Software Architecture initial version	Code Commanders

Code Commanders	Version: 1.0
Software Architecture Document	Date: 04/14/2024

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.	Architectural Representation	4
3.	Architectural Goals and Constraints	4
4.	Logical View	4
4.1	Overview	4
4.2	Architecturally Significant Design Packages	4
5.	Interface Description	5
6.	Quality	5

Code Commanders	Version: 1.0
Software Architecture Document	Date: 04/14/2024

Software Architecture Document

1. Introduction

1.1 Purpose

This document will provide the architectural design phase of the calculator and the details of the architecture used.

1.2 Scope

This document will discuss the design of the calculator on a basic system and class level.

1.3 Definitions, Acronyms, and Abbreviations

The document will utilize the same terminology mentioned in the Project Plan

1.4 References

The document will reference the UML class and package diagrams in their respective folders inside the UML Diagrams folder, the Project Plan for the definitions and the Software Requirement Specifications for the functionality and the constraints for this project to be developed.

1.5 Overview

This document will explain how the architecture is represented in this calculator program, as well as the systematical requirements related to architecture, such as the details and the packages needed to run this program and how the architecture relates to the constraints and non-functional requirements.

2. Architectural Representation

The calculator's architecture will be based on the UML class diagrams and relate architectural parts and their responsibilities.

3. Architectural Goals and Constraints

Our goals for the architecture and the constraints are to utilize the C++ programming language to conduct a calculation between multiple operations and classes. The three classes the calculator module will use include the interface, the tokenizer, and the parser class. These classes will apply how the calculator can handle certain functions and the architecture creates separation so that there is no conflict between distinct functions. We will also need to ensure each function will solve correctly without disturbing any other processes.

4. Logical View

4.1 Overview

This project will utilize the terminal for inputting the functions and will have a calculator truth evaluator module to ensure the operands are valid and will return their expected values.

4.2 Architecturally Significant Design Modules or Packages

The design modules and packages are represented in their respective UML class diagrams located in the project folder.

Code Commanders	Version: 1.0
Software Architecture Document	Date: 04/14/2024

5. Interface Description

The interface will be designed with C++ and be run through the terminal, and it will include text for a user to enter a desired function with various operands and variables. From the calculator's endpoint, we will ensure the user is able to enter their desired function and then this will be sent to the module to determine its validity, in which it will check for any errors or if it can be solved.

6. Quality

By distributing the calculator's functionality between the tokenizer and the evaluator ensures efficient operation of tasks for evaluating a Boolean expression and how the calculator can handle multiple operands and detecting any errors that may occur.