

CSC 413 Project Documentation

Summer 2025

Nawah Bakhtani

922684526

413 01

<https://github.com/NawahB/Calculator>

1. Introduction

This project is a Java-based calculator application designed to evaluate infix mathematical expressions using object-oriented programming principles. It integrates both a backend evaluator and a graphical user interface (GUI) for user interaction.

1.1 Project Overview

The calculator allows users to input arithmetic expressions using standard operators (+, -, *, /, ^) and parentheses. It evaluates the expression with correct operator precedence using stacks and displays the result through a GUI.

1.2 Technical Overview

The application is split into multiple Java classes, each representing a component of the expression evaluation logic. The GUI is built using Java Swing, and the evaluation engine uses two stacks to implement operator precedence and associativity.

1.3 Summary of Work Completed

- Designed and implemented Operand and Operator abstraction.

- Created subclasses for all supported operators: AddOperator, SubtractOperator, MultiplyOperator, DivideOperator, and PowerOperator.
- Built the Evaluator class to process infix expressions using stacks.
- Developed a user-friendly GUI in EvaluatorUI with a hot pink theme and interactive buttons.

2. Development Environment

- Language: Java 17
- IDE: Visual Studio Code
- Build Tools: javac / terminal
- OS Tested: macOS, Windows

3. How to Build/Import Your Project

1. Place all .java files in a single folder.
2. Open the folder in VS Code.
3. Open the terminal and run: `javac *.java`

4. How to Run Your Project

After compilation, run the project using:

```
java EvaluatorUI
```

This will launch the GUI calculator.

6. Assumptions Made

- Input expressions are space-separated like (2 + 3 * 4 instead of 2+3*4).
- Expressions are mathematically valid and do not include unsupported characters.
- Only integer operations are supported (no decimals).

6. Implementation Discussion

The core logic uses the Shunting Yard algorithm idea. The Evaluator class maintains two stacks: one for operands and one for operators. It processes the input string token by token. Operators are pushed based on precedence, and calculations occur when higher precedence is resolved. Each operator subclass handles its respective operation.

6.1 Class Diagram

-Operand: holds an integer value

-Operator (abstract): defines priority() and execute()

-AddOperator, SubtractOperator, etc.: implement Operator

-Evaluator: contains logic to parse and compute expression

-EvaluatorUI: Swing GUI that integrates evaluator backend

7. Project Reflection

This project deepened understanding of Java OOP principles, especially abstraction and class design. It also reinforced algorithmic thinking related to stack-based expression evaluation and GUI construction using Swing.

8. Project Conclusion / Results

The calculator correctly evaluates infix expressions with proper operator precedence and displays results in a themed GUI. The code is modular and ready for future updates like decimals, keyboard input, or history tracking. Overall, it meets all project goals and shows strong use of object-oriented design.