Multi-Function Command Line Calculator

A versatile mathematical tool for both basic arithmetic and advanced operations, built with C programming for speed, efficiency, and simplicity.

INTRODUCTION

Project Overview

What It Does

The Multi-Function Command Line Calculator performs basic arithmetic operations like addition, subtraction, multiplication, and division, as well as advanced functions including square roots and exponential powers—all directly from your terminal.

Our Goal

Provide developers and students with a fast, lightweight, and user-friendly interface for mathematical computations. Built for near-instant execution and seamless integration into terminal-based workflows.

MOTIVATION

Why I Chose This Project

Learning C Fundamentals

This project allowed me to strengthen my understanding of C programming, including control structures, functions, and library integration with math.h.

Practical Application

I wanted to build something useful that solves real problems—a tool developers actually use in their daily workflow for quick calculations without leaving the terminal.

Portfolio Development

Creating a well-documented project demonstrates my coding skills and problem-solving abilities to potential employers and academic evaluators.

TECHNOLOGY STACK

Tools and Technologies

C Language

Built using the C11 Standard, leveraging the powerful math.h library for advanced mathematical calculations and operations.

GCC Compiler

Compiles the source code efficiently, linking the math library for seamless mathematical function integration.

VS Code

Primary development environment for writing clean code, organizing project structure, and debugging with powerful extensions.

GitHub

Version control system to manage code updates, track changes, and ensure code stability throughout the development process.

ARCHITECTURE

How the Calculator Works

User Menu

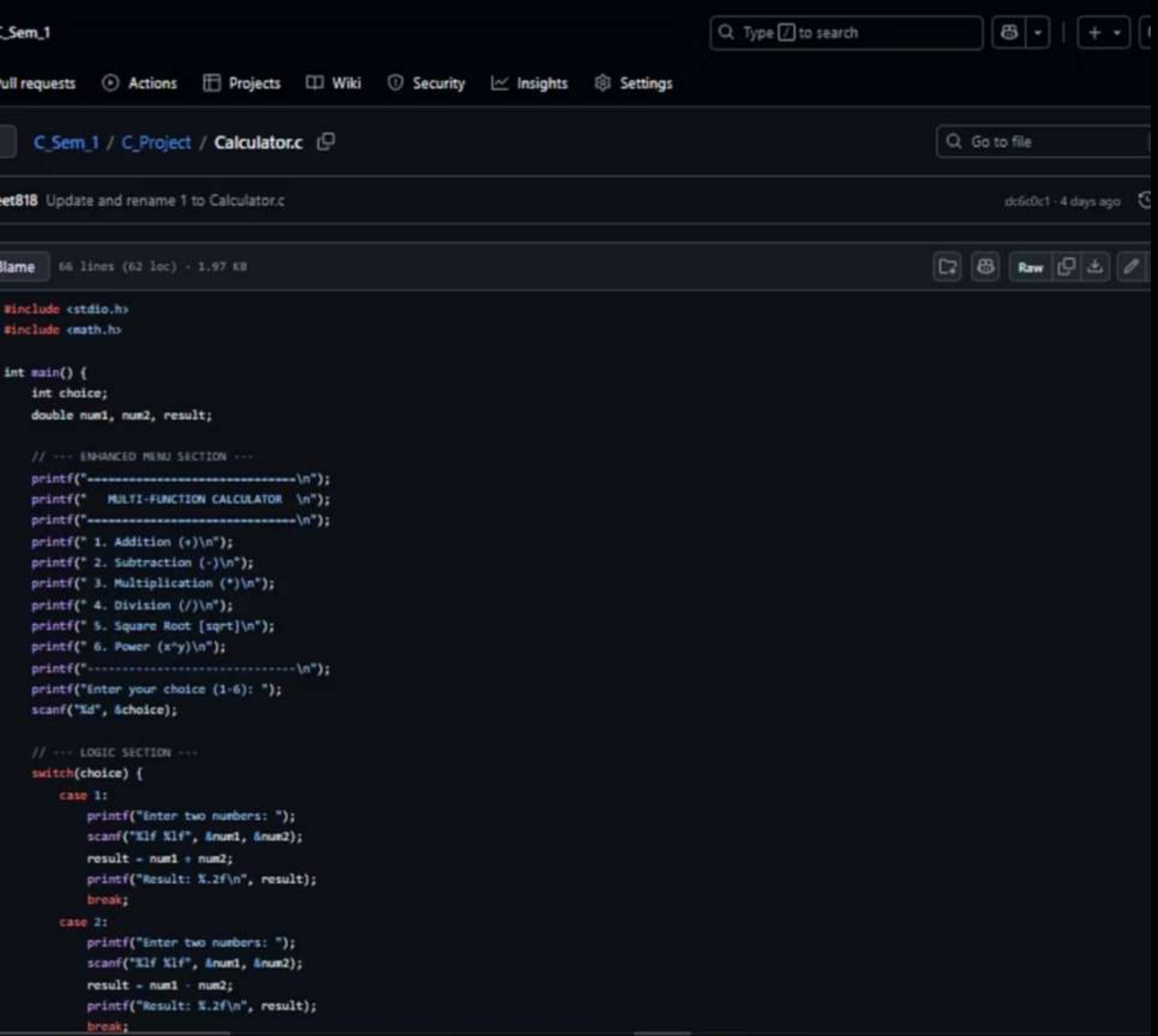
A stylized printf menu presents six distinct mathematical operation options in a clear, intuitive format.

Control Flow

A switch-case structure efficiently routes the user's choice to the corresponding logic block for processing.

Computation

The selected operation executes using C's math library, delivering precise results formatted to two decimal places.



CODE IMPLEMENTATION

Code Explanation

Below is the complete implementation showcasing the calculator's core logic, including user interface, operation handling, and mathematical computations. The code demonstrates clean structure and efficient use of C programming concepts.

View Full Code on GitHu

```
42          case 4:
43              printf("Enter dividend and divisor: ");
44              scanf("%lf %lf", &num1, &num2);
45              result = num1 / num2;
46              printf("Result: %.2f\n", result);
47              break;
48          case 5:
49              printf("Enter number: ");
50              scanf("%lf", &num1);
51              result = sqrt(num1);
52              printf("Square Root: %.2f\n", result);
53              break;
54          case 6:
55              printf("Enter base and exponent: ");
56              scanf("%lf %lf", &num1, &num2);
57              result = pow(num1, num2);
58              printf("Result: %.2f\n", result);
59              break;
60          default:
61              printf("Invalid Option selected!\n");
62      }
63
64      printf("----------------------------\n");
65      return 0;
66  }
```

Code Walkthrough

01

Header Inclusion

The program includes stdio.h for input/output operations and math.h for advanced mathematical functions like sqrt() and pow().

02

Menu Display

A formatted menu presents all six operations with clear numbering, allowing users to select their desired calculation type.

03

Input Handling

The program captures user choice via scanf() and validates the input to ensure it falls within the acceptable range.

04

Switch-Case Logic

Based on the user's selection, the appropriate case executes, prompting for operands and performing the requested calculation.

05

Result Display

Results are formatted using %.2f precision and displayed with clear labels, ensuring professional output presentation.

DESIGN EXCELLENCE

Key Features

1

Format Specifiers

Uses %.2f precision to limit output to two decimal places, ensuring clean and professional result presentation for all calculations.

2

Error Handling

A default case gracefully handles incorrect inputs, preventing program crashes and providing helpful feedback to guide the user.

FUTURE VISION

Future Scope and Enhancements

Planned Improvements

- **Trigonometric Functions:** Add sin, cos, tan, and their inverse operations for expanded mathematical capability
- **History Feature:** Implement calculation history storage to review previous operations and results
- **Expression Parser:** Support complex mathematical expressions like "(5 + 3) * 2" instead of single operations
- **Scientific Notation:** Handle very large and very small numbers using exponential notation
- **Unit Conversion:** Integrate common conversions for temperature, distance, and weight

Thank You!

Multi-Function Command Line Calculator

Thank you for reviewing this project. This calculator demonstrates fundamental C programming concepts while providing a practical tool for developers and students. Your feedback and contributions are welcome!

<1ms	6	100%
Lightning Speed	Operations	Reliability
Near-instant execution due to compiled C	Complete math toolkit from basic to advanced	Robust error handling prevents crashes

View Full Code on GitHu