SCIENTIFIC CALCULATOR

HIMANSHU ARYA

NOVEMBER 29, 2025

Abstract

This project presents a menu-driven Scientific Calculator implemented in C programming language. The calculator performs basic arithmetic operations along with scientific functions such as square root, exponentiation, trigonometric functions, and logarithmic calculations. The program uses loops, conditional statements, math library functions, and error handling to create an interactive user experience. This report explains the problem definition, system design, algorithm, implementation, testing, results, and future enhancements.

CONTENTS

---

1. Problem Definition

1.1 Overview

Scientific calculators are widely used tools for performing mathematical operations ranging from basic arithmetic to complex scientific calculations. This project aims to create a command-line scientific calculator using C that lets users select operations from a menu and receive accurate results.

The calculator offers: arithmetic operations, exponent calculations, square roots, trigonometric functions (sin, cos, tan), and logarithmic functions (ln, log10). The program continues running until the user chooses to exit.

1.2 Objectives

The objective of this project is to develop a scientific calculator that:

- Displays a user-friendly menu of operations
- Accepts and validates user inputs
- Performs arithmetic and scientific calculations
- Handles errors such as division by zero and invalid domain inputs
- Allows repeated operations through continuous looping

---

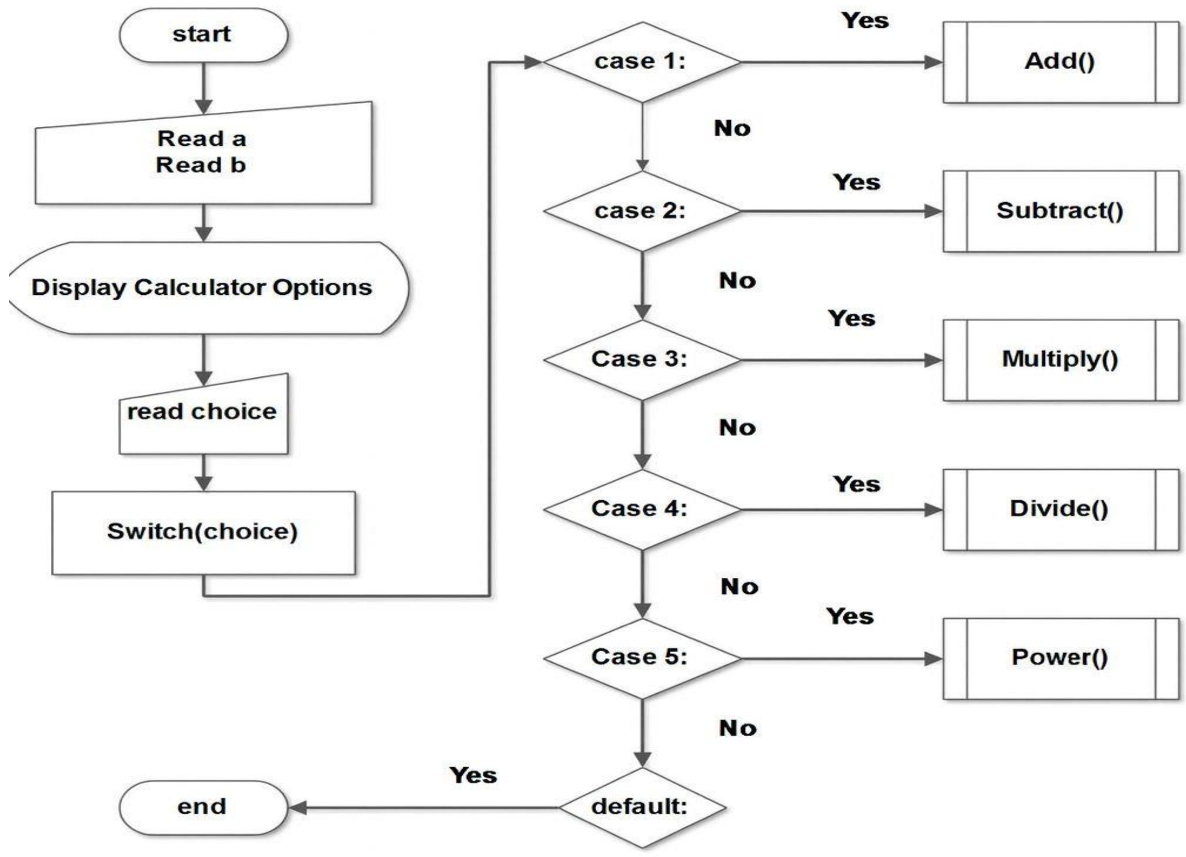
2. System Design

2.1 Algorithm

1. Start program.
2. Display menu of operations.
3. Ask the user for choice.
4. If choice = 0 → Exit program.

5. If operation requires two numbers → Accept num1 and num2.
6. If operation requires one number → Accept num1.
7. Perform calculation using correct math function.
8. Display result.
9. Handle errors (zero division, negative square root, log domain error).
10. Loop back to menu until exit.
11. End program.

2.2 Flowchart



3. Implementation Details

3.1 Key Features

1. Menu-Driven System for easy user interaction.
2. Arithmetic Functions: addition, subtraction, multiplication, division.
3. Scientific Operations: power, square root.
4. Trigonometric Functions using sin(), cos(), and tan().
5. Logarithmic Functions using log() for natural log and log10() for base-10.
6. Error Handling for invalid inputs.
7. Infinite Loop until user chooses exit.

3.2 Code Snippets

Menu Display

```
printf("1. Addition\n");
printf("2. Subtraction\n");
printf("3. Multiplication\n");
printf("4. Division\n");
printf("5. Power (x^y)\n");
printf("6. Square Root\n");
printf("7. Sine (sin x)\n");
printf("8. Cosine (cos x)\n");
printf("9. Tangent (tan x)\n");
printf("10. Logarithm (ln x)\n");
printf("11. Log10 (log10 x)\n");
```

Division with Error Handling

```
if (num2 == 0) {
    printf("Error! Division by zero.\n");
} else {
    result = num1 / num2;
}
```

Square Root Validation

```
if (num1 < 0)
    printf("Error! Negative number.\n");
else
    result = sqrt(num1);
```

Logarithm Validation

```
if (num1 <= 0)
    printf("Error! Log undefined.\n");
else
    printf("ln(%.2lf) = %.4lf\n", num1, log(num1));
```

---

4. Testing & Results

4.1 Test Case 1 — Addition

Input: 4, 6
Expected Output: 10
Result: ✓ Pass

4.2 Test Case 2 — Division by Zero

Input: 5 / 0
Output: Error! Division by zero.
Result: ✓ Pass

4.3 Test Case 3 — Negative Square Root

Input: -9
Output: Error! Negative number.
Result: ✓ Pass

4.4 Test Case 4 — Logarithm Invalid Input

Input: log(-4)
Output: Error! Log undefined.
Result: ✓ Pass

---

5. Conclusion & Future Work

5.1 Conclusion

The scientific calculator successfully performs arithmetic, trigonometric, exponential, and logarithmic operations using C. Error handling ensures the program responds safely to invalid inputs. The menu-driven design creates an easy-to-use interface for users.

5.2 Future Work

Future enhancements may include:

- Adding factorial, permutation, and combination functions
- Converting degrees to radians automatically
- Introducing memory functions (M+, MR, MC)
- Creating a graphical user interface (GUI)

- Adding calculation history

---

6. References

- C Standard Library Documentation (stdio.h, math.h)
- Online educational resources on scientific calculators in C