# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**Jnana Sangama, Belagavi - 590018**



**Mini project**

***On***

## “Scientific Calculator CP56”

**By**

**ABHISHEK KUMAR 4MT21AI003**

**PREETHAM 4MT21AI038**

**SAMPREETH 4MT21AI044**

**VAISHNAVI 4MT21AI060**

**MANAS 4MT21AI028**

### 

### Department of Artificial Intelligence & Machine Learning



**DEPARTMENT OF** **ARTIFICIAL INTELLIGENCE & MACHINE LEARNING**

**(*Accredited by NBA*)**

### MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

*Accredited by NAAC with A+ Grade, An ISO 9001: 2015 Certified Institution*

(*A Unit of Rajalaxmi Education Trust®, Mangalore - 575001*)

Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi

*Badaga Mijar, Moodabidri-574225, Karnataka*

2022-23

**ABSTRACT**

SciCal, a C-based software project, provides a command-line interface for precise mathematical calculations. Modeled after efficient pharmaceutical systems, SciCal offers diverse mathematical functions, user access control, and custom expression capabilities, assisting professionals and students in tackling complex equations

### INTRODUCTION

This C program is designed to function as a basic scientific calculator. It allows users to perform various mathematical operations, including addition, subtraction, multiplication, division, exponentiation, and finding the square root of a number. The program interacts with the user by displaying a menu of available operations, prompting for user input, and providing the result of the chosen operation.

**Objectives of Scientific Calculator:**

The objective of this C program for a simple scientific calculator is to create a versatile tool for performing various mathematical operations efficiently. This program serves as a learning exercise for programming and aims to achieve the following goals:

* **Mathematical Operations:** The program enables users to perform common mathematical operations, including addition, subtraction, multiplication, division, exponentiation, and finding square roots.
* **User Interaction:** It provides an interactive user interface where users can choose an operation from a menu and input the necessary numbers for calculation.
* **Error Handling:** The program incorporates error handling to prevent common mathematical errors, such as division by zero or attempting to find the square root of a negative number. It ensures that the program gracefully handles these situations and provides informative error messages.
* **Usability:** By offering a straightforward menu-driven interface, the program aims to be user-friendly and accessible to individuals who need to perform basic mathematical calculations quickly.

**TECHNOLOGIES USED**

1. C Programming Language: The entire program is written in the C programming language.
2. **stdio.h**: This header file is used for standard input and output operations. It provides functions like printf and scanf for console input and output.
3. **math.h** is a standard C library header file that provides mathematical functions and constants for performing various mathematical calculations in C programs.
4. **conio.h** is a non-standard C header file used primarily in older DOS and Windows environments to provide console input and output functions, but it is not supported by modern C compilers, and alternative methods are preferred for console interactions in contemporary programming.

**SYSTEM ARCHITECTURE**

1. **User Interface (UI):**
   * The User Interface layer presents a minimal interface for users to input calculations and view results.
   * It consist of a command-line interface or a basic graphical interface with buttons for digits and operations.
2. **Calculator Logic:**
   * The Calculator Logic layer contains a switch-based structure that evaluates the user's input and performs calculations based on the selected operation.
3. **Error Handling:**
   * Includes error handling within the calculator logic to detect and handle errors such as division by zero or invalid input.
4. **User Input Processing:**
   * This component receives and validates user input, converting it into a format that can be processed by the calculator logic.

**DESIGN AND IMPLEMENTATION**

**Front-End Design :**

• The front-end design is based on a command-line interface, where users can select

options by entering numbers corresponding to their desired actions.

**Back-End Design :**

• The back-end logic is implemented in the C programming language.

• Each module has its functions to accept inputs from user and the type of operation

And perform the required calculations.

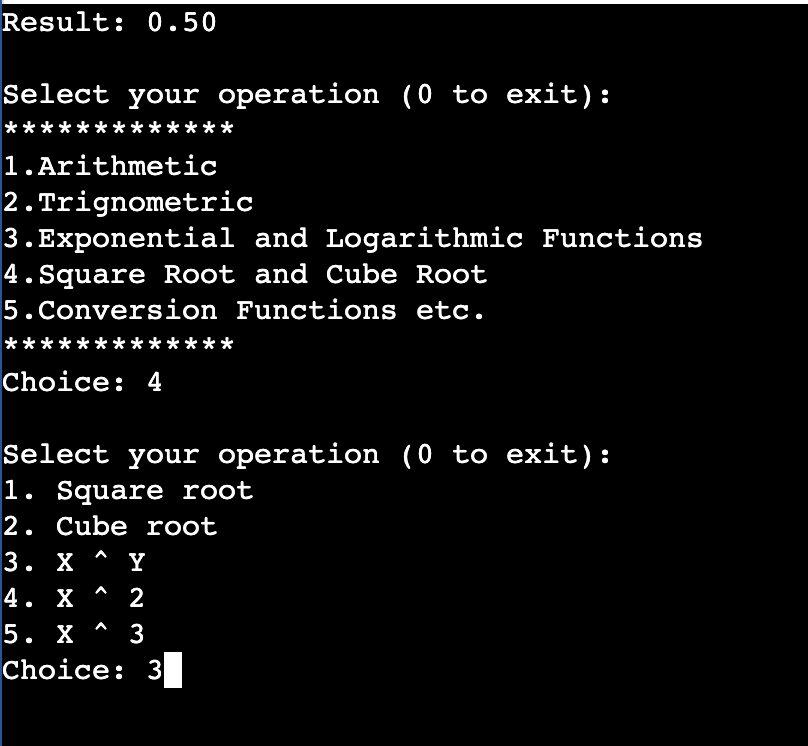
TEST CASES:

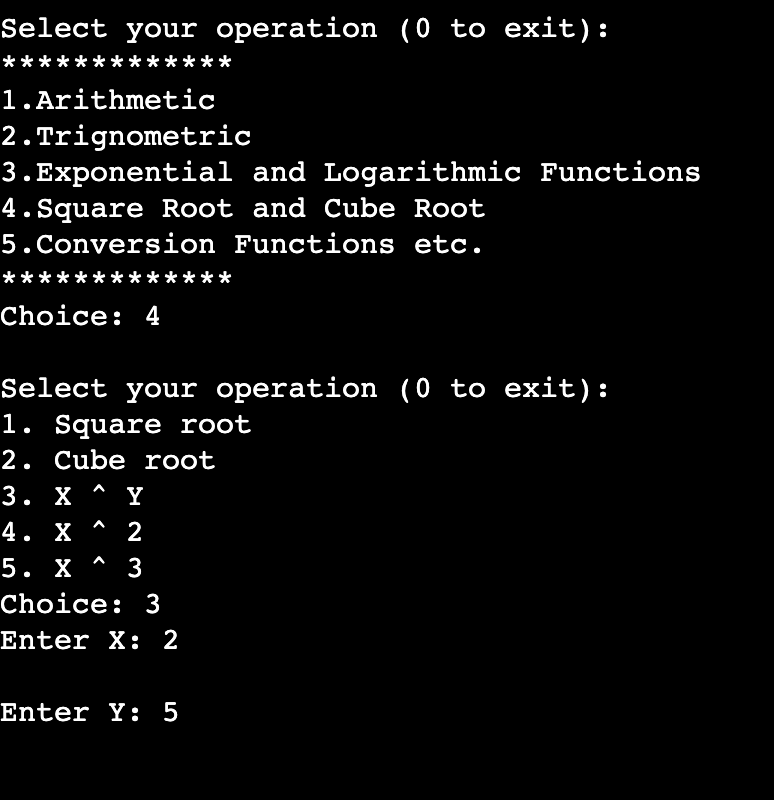
A screen shot of a computer program

Description automatically generated

A black screen with white text

Description automatically generated





**FEATURES AND FUNCTIONALITY**

1. **Basic Arithmetic Operations**: The calculator can perform common arithmetic operations, including addition, subtraction, multiplication, and division.
2. **Trignometric Functions**
3. **Exponential and Logarithmic Functions**
4. **Square Root and Cube Root**
5. **Conversion Functions**
6. **Error Handling**: It includes error checking to handle scenarios like division by zero or invalid input, displaying appropriate error messages.
7. **Simple User Interface**: The user interface is typically minimalistic, with a command-line interface.
8. **Result Display:** The program displays calculation results in a user-friendly format, often with a specified number of decimal places or scientific notation for large or small numbers.
9. **Exit or Quit Option**: The program allows users to exit or quit the calculator when they are done.

**CONCLUSION**

In summary, this project has successfully delivered a basic but functional scientific calculator that fulfills its primary goal of providing a user-friendly tool for everyday mathematical calculations. While it serves as a simple introduction to calculator development, it can be a foundation for further enhancements and customization based on specific user requirements and preferences. This calculator project showcases essential programming concepts, including user input processing, error handling, and calculation logic, which can serve as a valuable learning experience for aspiring developers.

### REFERENCES

🡪 <https://practice.geeksforgeeks.org>

🡪 <https://chat.openai.com>

🡪 <https://stackoverflow.com/>

🡪<https://miro.com/>

🡪<https://draw.io/>