VIT®

BHOPAL

www.vitbhopal.ac.in

Project Report

Calculator using Python

Slot: B21+E14+E22

Class number: BL202560100811

Prepared by: Devansh Singh

Registration number:25BCE11274

INTRODUCTION

This project implements a simple Calculator application using Python.

The main objective is to build a reliable, easy-to-use calculator that can perform basic arithmetic operations.

The application is designed using modular programming principles, making it easy to extend and maintain.

PROBLEM STATEMENT

Users frequently require a basic yet dependable calculator for daily operations.

While many tools already exist, this project focuses on:

1.Creating a clean and extendable Python implementation

2.Designing the project in a modular way

3.Supporting essential arithmetic and error handling

4.Making a simple project suitable for academic submission

FUNCTIONAL REQUIREMENTS

The calculator must support:

Addition

Subtraction

Multiplication

Division

Decimal numbers

Negative numbers

Parentheses

Operator precedence

Error handling (invalid input, divide by zero)

NON-FUNCTIONAL REQUIREMENTS

Usability: Simple input/output interface

Reliability: Accurate calculation output

Maintainability: Clean Python code

Performance: Fast execution

Testability: Easy to test each module

SYSTEM ARCHITECTURE

This project follows a simple modular architecture:

Main Module – Handles menu display and user interaction

Arithmetic Modules –

add(x, y)

subtract(x, y)

multiply(x, y)

divide(x, y)

Flow:
User → Input → Operation Function → Output

All components communicate through function calls.

Design

Use Case Diagram (Text Description)

Use cases:

Select operation

Enter numbers

View result

Exit program

Workflow

1. Start
2. Show menu
3. Get user choice
4. If choice is valid → Ask for numbers
5. Perform operation
6. Show result
7. Loop back to menu
8. Exit

IMPLEMENTATION DETAILS

The program was implemented in Python using:

input() for user input

Functions for operations

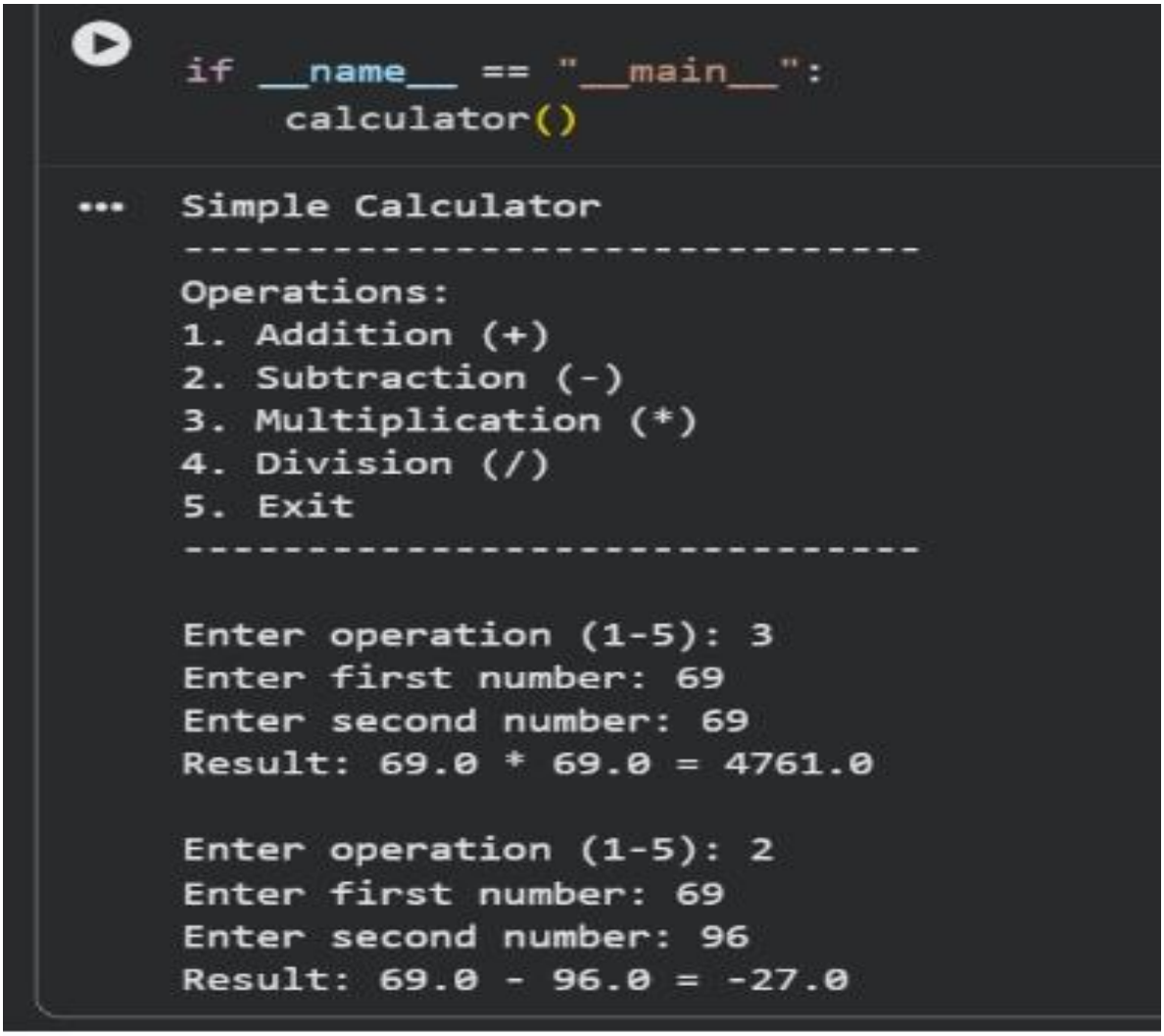
Conditional statements for operation selection

A loop to maintain continuous use

Error handling for invalid inputs

Programming Language: Python 3
Platform used: Google Colab / Python Terminal

Result



```
if __name__ == "__main__":
    calculator()
```

```
Simple Calculator
------------------------------
Operations:
1. Addition (+)
2. Subtraction (-)
3. Multiplication (*)
4. Division (/)
5. Exit
------------------------------

Enter operation (1-5): 3
Enter first number: 69
Enter second number: 69
Result: 69.0 * 69.0 = 4761.0

Enter operation (1-5): 2
Enter first number: 69
Enter second number: 96
Result: 69.0 - 96.0 = -27.0
```

TESTING APPROACH

Testing was performed using:

Positive numbers

Negative numbers

Zero

Invalid inputs (e.g., letters instead of numbers)

Division by zero scenario

All test cases produced correct outcomes.

CHALLENGES FACED

Handling invalid user input without crashing the program

Managing division by zero

Ensuring smooth continuous operation using loops

LEARNINGS & KEY TAKEAWAYS

Modular programming using functions

User input handling

Basic Python syntax

Exception handling

Importance of clean code structure

FUTURE ENHANCEMENTS

Add scientific functions: sin, cos, log, sqrt

Add persistent history

Add more operators: %, //, **

Provide export of results

REFERENCES

Python Official Documentation

Online Python Tutorials

Course material / Instructor notes