“First Project in Python”

Building a Simple Calculator Using Tkinter

Prepared By:Siliviya

Introduction

•A basic calculator using Python's Tkinter GUI library.

•To create a functional and user-friendly calculator with a graphical interface.

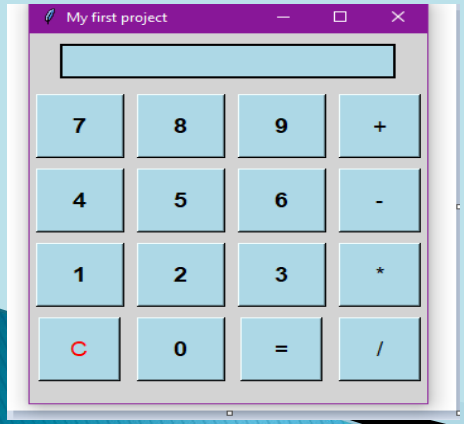
Features of the Calculator

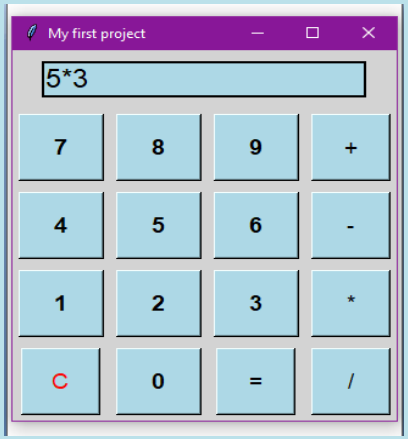
•Basic arithmetic operations (+, -, *, /)

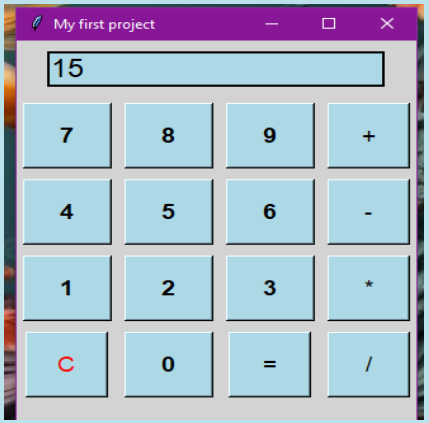
•Clear (C) button to reset input

•Equal (=) button to evaluate expressions

•Styled interface with buttons and entry field







Challenges Faced

- Handling user input errors gracefully (e.g., division by zero, invalid expressions).
- Ensuring a responsive and aesthetically pleasing UI.
- Managing button commands dynamically using lambda functions.
- Facing difficulties in formatting and applying colors effectively.

Solutions Implemented

- Used try-except block to handle errors and display "Error" for invalid inputs.
- Applied grid layout for organized button placement.
- Implemented conditional styling for different button types. Used lambda functions to pass button values dynamically.
- Experimented with different color schemes to improve UI aesthetics.

References:-

"This simple calculator was developed by referring to various YouTube tutorials and leveraging ChatGPT's guidance. YouTube provided visual demonstrations of coding concepts, while ChatGPT helped clarify logic, troubleshoot errors, and optimize the implementation."

Which functions used and why?

Tkinter- It provides tools to create desktop applications with buttons, labels, text fields, menus, and more. Its pre-installed with most Python distributions. (Read about it in Google and chatgpt)

if Statement - Conditional Execution, the if statement is used to execute code only if a certain condition is met.

if-else- Used else block to execute code when the condition is False.

try Statement - **Error Handling** - It is used to handle exceptions (errors) in Python.

Why Use try?
If an error occurs inside try, Python will not crash the program.
Instead, the except block catches the error and handles it.

root – The Main Window

In Tkinter, root represents the main application window. It is the first thing that must be created in any Tkinter program.

Why use root?

It is the parent container that holds all widgets (buttons, labels, entry fields, etc.).
Without root, the GUI window will not appear.
It controls the entire application lifecycle (created first, closed last).

Using for Loop to Create Calculator Buttons

In a GUI calculator, we often need multiple buttons (0–9, operators, etc.). Instead of creating each button separately, we can use a **for loop** to generate them dynamically.

Learning & Improvements

•Deepened understanding of Tkinter widgets and event handling.

•Improved UI design and layout structuring.

•Learned better error handling techniques.

•Future Enhancements: Adding more advanced functions (square root, exponentiation).

Conclusion

•Successfully built a functional calculator with Tkinter.

•Overcame key challenges with structured solutions.

•Scope for further improvements and feature enhancements.

Thank you.