SURIGAO DEL NORTE STATE UNIVERSITY

SURIGAO CITY

**PROJECT IN INTERMEDIATE**

**PROGRAMMING**

**GUI CALCULATOR**

**BSCS 1A1**

Submitted by:

Jasmin G. Ligad

Submitted to:

Dr. Unife O. Cagas

**DESCRIPTION**

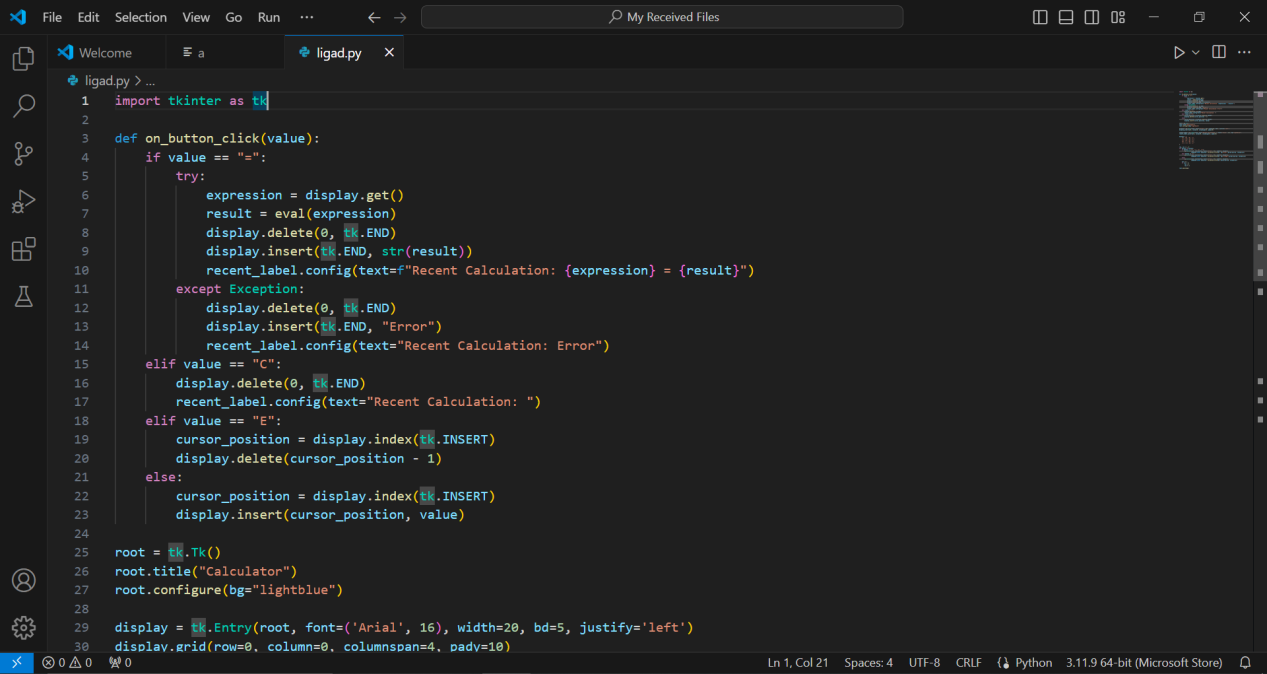
This Python script utilizes the Tkinter library to construct a basic calculator GUI. It features a display area for input and output, along with buttons for digits, arithmetic operations, and control functions like Clear and Equals. The calculator evaluates expressions using Python's eval() function and updates the display accordingly. Each button is linked to a function that handles its specific action, such as appending digits or operations to the input, clearing the display, or performing calculations. Through a grid layout, the buttons are arranged within the window, providing an intuitive user interface for performing arithmetic calculations.  
  
  
**BENEFITS** The code offers a practical solution for basic arithmetic calculations, providing users with a straightforward and efficient way to perform mathematical operations. Its user-friendly graphical interface makes it accessible to a wide range of users, allowing them to input expressions and obtain results with ease. Error handling ensures that exceptions are managed gracefully, preventing the application from crashing and providing feedback to users in case of invalid input or arithmetic errors. Furthermore, the code can be easily customized and serves as an educational tool for individuals learning Python programming or GUI development with Tkinter, offering insights into event-driven programming and widget creation. Overall, the calculator code streamlines the process of arithmetic computation, offering convenience and functionality to users across various contexts.

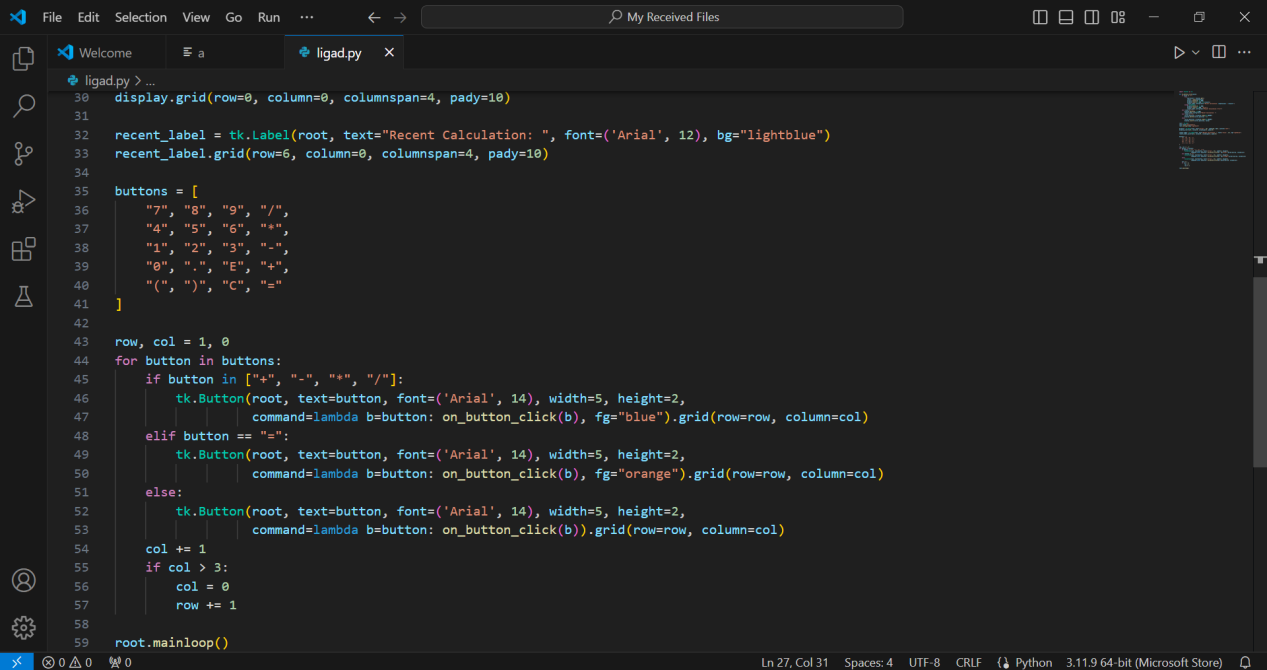
**SIGNIFICANCE OF THE STUDY**The calculator code holds significance in its practicality and adaptability. Its efficiency streamlines arithmetic tasks, ensuring accurate results while saving time and effort. Accessible through a user-friendly graphical interface, it caters to users of varying skill levels, eliminating barriers to mathematical computation. As a learning tool, it offers insights into programming concepts and GUI development, aiding individuals in their educational pursuits. The code's versatility allows for customization and integration into diverse applications, enhancing convenience across academic, professional, and personal contexts. Overall, the calculator code serves as a valuable tool, simplifying arithmetic operations and promoting accessibility and learning in mathematics and programming.

**PURPOSE** The calculator code aims to offer users a quick and efficient way to perform arithmetic calculations, simplifying tasks like addition, subtraction, multiplication, and division. Its user-friendly interface makes it accessible to individuals of all skill levels, providing instant results and eliminating the need for manual computation.

**FEATURES** 1.User-Friendly Interface: It provides a graphical user interface (GUI) with buttons for numbers, operators, and special functions, making it easy to input arithmetic expressions.  
2. Real-Time Calculation: The code instantly evaluates arithmetic expressions as they are inputted, providing immediate results to the user.  
3. Error Handling: It includes error handling mechanisms to manage exceptions that may occur during evaluation, such as division by zero or invalid input.  
4. Clearing and Editing: Users can clear the input field (C button) to start a new calculation, or use the backspace (E button) to edit the current input.  
5. Recent Calculation Display: The code shows the most recent calculation performed, providing users with a reference to their previous inputs and results.  
6. Customization: Developers can easily customize the code to add new features, modify the interface, or integrate it into larger applications according to their requirements.

**CODE**

****

****

**FUNCTIONALITIES**

import tkinter as tk

def on\_button\_click(value):

    if value == "=":

        try:

            expression = display.get()

            result = eval(expression)

            display.delete(0, tk.END)

            display.insert(tk.END, str(result))

            recent\_label.config(text=f"Recent Calculation: {expression} = {result}")

        except Exception:

            display.delete(0, tk.END)

            display.insert(tk.END, "Error")

            recent\_label.config(text="Recent Calculation: Error")

    elif value == "C":

        display.delete(0, tk.END)

        recent\_label.config(text="Recent Calculation: ")

    elif value == "E":

        cursor\_position = display.index(tk.INSERT)

        display.delete(cursor\_position - 1)

    else:

        cursor\_position = display.index(tk.INSERT)

        display.insert(cursor\_position, value)

The ‘**on\_button\_click(value**)’ function manages user interactions in the calculator interface. It evaluates arithmetic expressions when the "=" button is clicked, clears the display when the "C" button is clicked, deletes the last character when the "E" button is clicked, and inserts characters into the display for other buttons. It handles errors during evaluation and updates the recent calculation label accordingly.

root = tk.Tk()

root.title("Calculator")

root.configure(bg="lightblue")

This segment of code initializes the Tkinter window, setting its title to "Calculator" and configuring its background color to light blue. It creates the main window where the calculator interface will be displayed.

display = tk.Entry(root, font=('Arial', 16), width=20, bd=5, justify='left')

display.grid(row=0, column=0, columnspan=4, pady=10)

This code segment creates the display area of the calculator interface using a Tkinter Entry widget. It specifies the font, width, border size, and text alignment for the display. The display is then positioned in the Tkinter grid layout to occupy the first row and span across all four columns with vertical padding applied.

recent\_label = tk.Label(root, text="Recent Calculation: ", font=('Arial', 12), bg="lightblue")

recent\_label.grid(row=6, column=0, columnspan=4, pady=10)

This code snippet creates a Label widget named ‘**recent\_label**’ within the Tkinter window ‘**root**’. The label displays the text "Recent Calculation: " using the Arial font at size 12 and is styled with a light blue background. It is positioned in the Tkinter grid layout at row 6 and column 0, spanning across all four columns to align with the display area. Vertical padding is applied to create spacing between the label and other widgets

buttons = [

    "7", "8", "9", "/",

    "4", "5", "6", "\*",

    "1", "2", "3", "-",

    "0", ".", "E", "+",

    "(", ")", "C", "="

]

This code segment defines a list named ‘**buttons**’, which contains labels for various buttons in the calculator interface. These labels include digits, arithmetic operators, special functions, and additional characters like decimal point and parentheses. The buttons are organized in a specific order to facilitate easy input and calculation within the calculator interface..

row, col = 1, 0

for button in buttons:

    if button in ["+", "-", "\*", "/"]:

        tk.Button(root, text=button, font=('Arial', 14), width=5, height=2,

                  command=lambda b=button: on\_button\_click(b), fg="blue").grid(row=row, column=col)

    elif button == "=":

        tk.Button(root, text=button, font=('Arial', 14), width=5, height=2,

                  command=lambda b=button: on\_button\_click(b), fg="orange").grid(row=row, column=col)

    else:

        tk.Button(root, text=button, font=('Arial', 14), width=5, height=2,

                  command=lambda b=button: on\_button\_click(b)).grid(row=row, column=col)

    col += 1

    if col > 3:

        col = 0

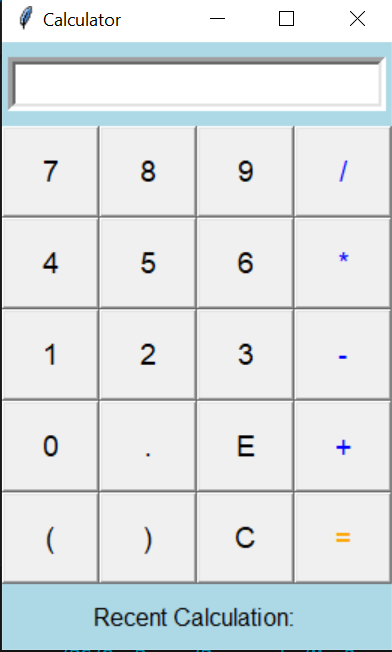
        row += 1

This code snippet dynamically creates Tkinter Button widgets for each button label in the ‘**buttons’** list. It arranges the buttons in a grid layout, with each row containing four buttons. Arithmetic operator buttons are colored blue, the "=" button is colored orange, and all other buttons have default colors. The ‘**on\_button\_click**’ function is assigned as the command for each button, allowing them to perform specific actions when clicked. The ‘**row**’ and ‘**col’** variables are used to manage the grid layout, ensuring proper positioning of the buttons.

root.mainloop()

The ‘**root.mainloop()**’ function initiates the Tkinter event loop, ensuring the responsiveness of the graphical user interface (GUI) by continuously listening for and handling user interactions, such as button clicks or window resizing.

**OUTPUT**

****

**SOFTWARE USED:**

-Visual Studio Code

**HARDWARE USED:**

**-**Laptop

**REFERENCE**

**<https://chatgpt.com/>**

**<https://www.geeksforgeeks.org/python-simple-gui-calculator-using-tkinter/>**