



THE OPEN UNIVERSITY OF SRI LANKA

Python

Assignment – 01

Name: - W.L.Hashini Sathsarani

Reg No: - 123587559

Complete Date: - 2025.01.26

Closing date: -2025.01.30

Question

Deadline - 30th of January 2025

Email your answers - hnmun@ou.ac.lk

Assignment Objective

Design and implement a **simple calculator** using Python. The program should have a graphical frontend and be fully functional for basic arithmetic operations. This assignment will help you understand the integration of Python's logic with a simple GUI framework and demonstrate the use of built-in functions for computation.

Assignment Requirements

Functionality

- The calculator must be designed to perform the basic arithmetic operations listed below:
 - Addition (+)
 - Subtraction (-)
 - Multiplication (*)
 - Division (/)
- Make sure that the calculator can handle edge cases properly, such as preventing errors when attempting to divide a number by zero or when encountering invalid inputs.

User Interface (UI)

- Create a graphical user interface (GUI) for the calculator using a simple and accessible framework such as tkinter.
- The GUI should have the followings:
 - A display area that will show the current input from the user as well as the calculated results.
 - Clearly labeled buttons for numbers (0-9) and for the arithmetic operations (+, -, *, /, and =).
 - A "Clear" button to reset the input and clear the calculator display.

Programming Logic

- Use Python's built-in functions to implement the core arithmetic operations for the calculator.
- The program should be written with modularity in mind, meaning each operation (e.g., addition, subtraction, multiplication, and division) should be implemented as a separate function to improve code clarity and reusability.

Behavior

- Users should be able to input numbers and select operations using the buttons provided in the GUI.
- As the user interacts with the calculator, the display should update to reflect the input and show the calculated result when the user presses the = button.
- The program should be capable of handling errors properly, such as displaying an error message when invalid inputs are entered or when division by zero is attempted.

Sample Input and Output

Example 1:

- User clicks 5, +, 3, =
- Display: 8

Example 2:

- User clicks 10, /, 0, =
- Display: Error: Division by Zero

Submission Requirements

1. You are required to submit the complete Python script as a .py file, ensuring it contains all the necessary code for the calculator's functionality.
2. Make sure to include detailed comments throughout your program to explain the purpose and functionality of each part of the code, making it easier to understand and follow.
3. It is important to test your program thoroughly to make sure that it runs without any errors or issues, providing a smooth and error-free experience.

Evaluation Criteria

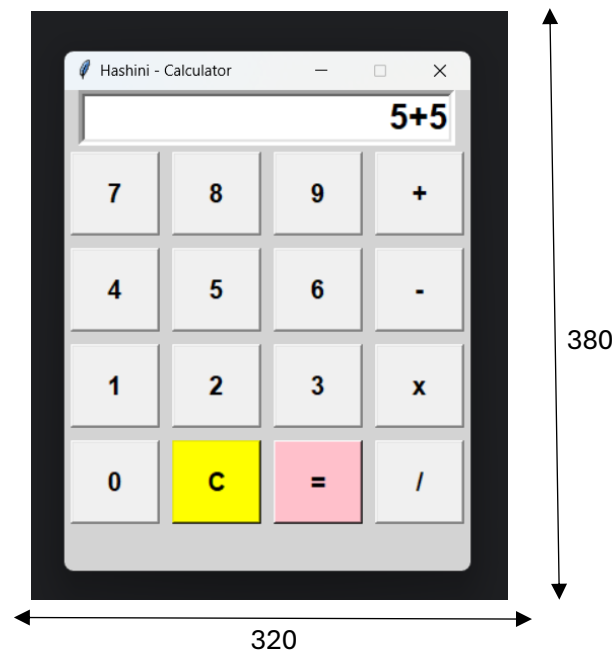
- **Functionality: 50%**
- **Code Quality: 20%**

Optional

- Add advanced operations such as square root, exponentiation, or memory functions.
- Make the UI more visually appealing with colors or themes.

****AI generated answers will not be accepted****

Calculator Design



This is my calculator design. Its height is 380 and its weight is 320. I use light gray color in my calculator background. So, I want to highlight **C** button and **=** button because we mainly use these buttons.

Code

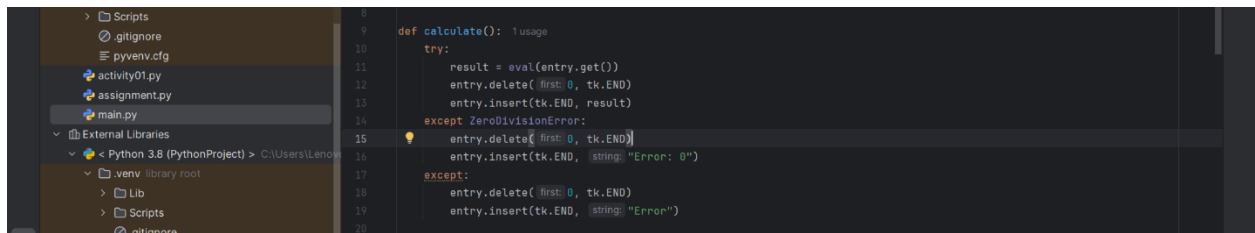
```
PythonProject Version control
Project main.py
1 import tkinter as tk
2
```

This is my calculator start code line. This line imports the **Tkinter** library with alias **tk**.tkinter provides tools to create windows, buttons text boxes...I watch lecturer notes and I use W3 school website.

Functions

```
... _virtualenv.pth
_virtualenv.py
distutils-precedence.pth
pip-23.2.1.virtualenv
setuptools-68.2.0.virtualenv
wheel-0.41.2.virtualenv
Scripts
1
2
3 # Functions
4 def button_click(number): 14 usages
5     current = entry.get()
6     entry.delete(0, tk.END)
7     entry.insert(tk.END, current + str(number))
8
```

This function is used when a number or operator is clicked on the calculator. it takes the number as an argument and appends it to the current value display in the entry box. This code the text box where numbers and results are displayed and clear the entry widget and appends the clicked number or operator to the end of the current value.



```
8
9
10 def calculate():
11     try:
12         result = eval(entry.get())
13         entry.delete(0, tk.END)
14         entry.insert(tk.END, result)
15     except ZeroDivisionError:
16         entry.delete(0, tk.END)
17         entry.insert(tk.END, string="Error: 0")
18     except:
19         entry.delete(0, tk.END)
20         entry.insert(tk.END, string="Error")
```

This function performs the calculation when the “=” button is clicked. This used the eval () function to evaluate the expression entered in the entry widget. If the evaluation is successful and it displays the result in the entry box. If there’s a division by 0 it shows “Error: 0”.



```
20
21 def clear():
22     entry.delete(0, tk.END) # Simply clear the entry box
23
```

This function clears the content of the entry widget when the “C” button is clicked. This C button deletes all the text currently in the entry widget. so this is more useful button code in this code.



```
23
24 # main window ake code aka
25 root = tk.Tk()
26 root.title("Hashini - Calculator")
27 root.config(bg='light gray')
28 root.geometry('320x380+0+0') # Size aka
29 root.resizable(width=False, height=False) # Disable resizing aka
30
```

This is setting up the main window. This section creates the main window of the application.

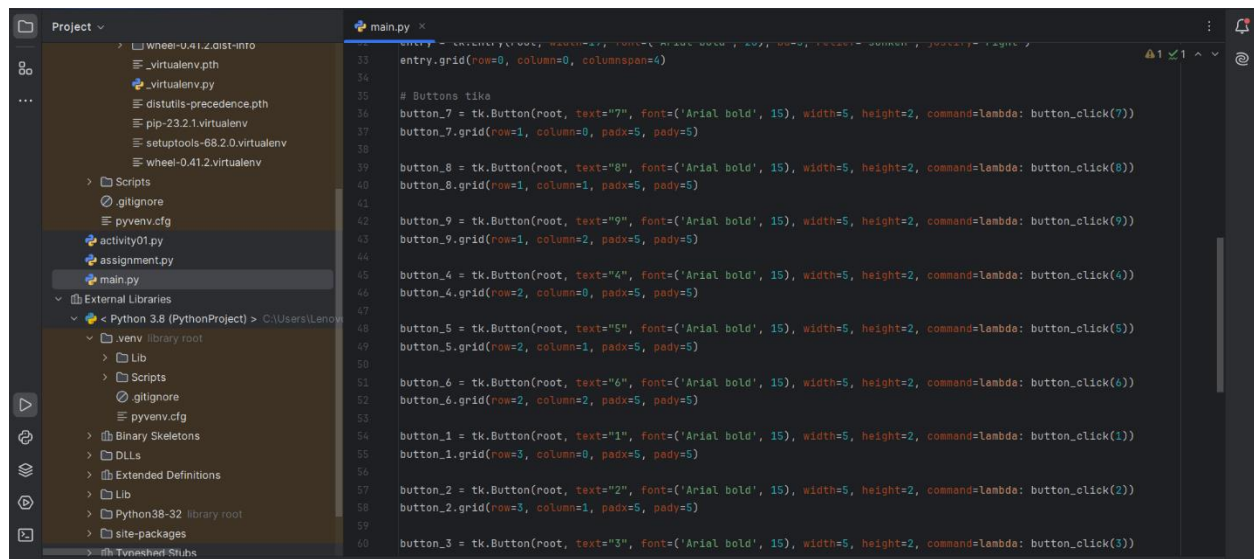
- **root = tk.Tk ()** - initializes the main window.
- **root.title(“Hashini – Calculator”)** – This is sets the title of the window.
- **root.config(bg = ‘light gray’)** – This is set the background color of the window to light gray.
- **root.geometry(‘320x380+0+0’)** – this line is defines the window size(320*380pixels)and the position (top-left corner of the screen)
- **root.resizable(False,False)** – this line this use to disables the ability to resize the window.



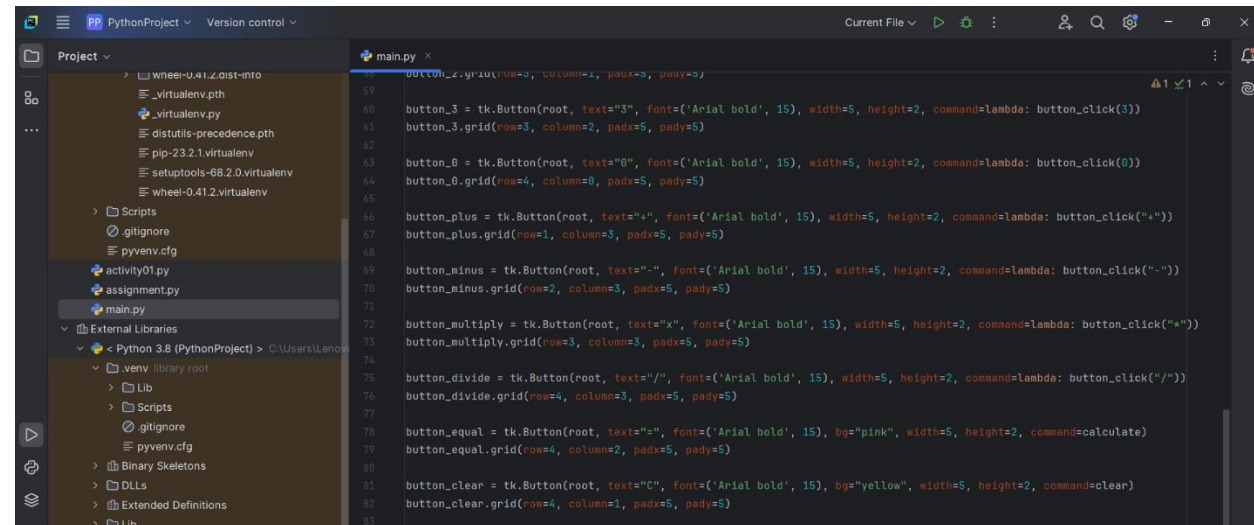
```
30
31 # Display box ake code aka
32 entry = tk.Entry(root, width=19, font=('Arial bold', 20), bd=5, relief="sunken", justify="right")
33 entry.grid(row=0, column=0, columnspan=4)
34
```

This is creating the entry widget. This is the text box where the numbers and results will be displayed. The entry widget is created with a width of 19 characters and a font size of 20. And it has a border (bd=5) and a sunken appearance (relief="sunken"). The text is aligned to the right (justify="right") and it spans across 4 columns in the grid layout (columnspan=4).

Buttons



```
33 entry.grid(row=0, column=0, columnspan=4)
34
35 # Buttons tika
36 button_7 = tk.Button(root, text="7", font=('Arial bold', 15), width=5, height=2, command=lambda: button_click(7))
37 button_7.grid(row=1, column=0, padx=5, pady=5)
38
39 button_8 = tk.Button(root, text="8", font=('Arial bold', 15), width=5, height=2, command=lambda: button_click(8))
40 button_8.grid(row=1, column=1, padx=5, pady=5)
41
42 button_9 = tk.Button(root, text="9", font=('Arial bold', 15), width=5, height=2, command=lambda: button_click(9))
43 button_9.grid(row=1, column=2, padx=5, pady=5)
44
45 button_4 = tk.Button(root, text="4", font=('Arial bold', 15), width=5, height=2, command=lambda: button_click(4))
46 button_4.grid(row=2, column=0, padx=5, pady=5)
47
48 button_5 = tk.Button(root, text="5", font=('Arial bold', 15), width=5, height=2, command=lambda: button_click(5))
49 button_5.grid(row=2, column=1, padx=5, pady=5)
50
51 button_6 = tk.Button(root, text="6", font=('Arial bold', 15), width=5, height=2, command=lambda: button_click(6))
52 button_6.grid(row=2, column=2, padx=5, pady=5)
53
54 button_1 = tk.Button(root, text="1", font=('Arial bold', 15), width=5, height=2, command=lambda: button_click(1))
55 button_1.grid(row=3, column=0, padx=5, pady=5)
56
57 button_2 = tk.Button(root, text="2", font=('Arial bold', 15), width=5, height=2, command=lambda: button_click(2))
58 button_2.grid(row=3, column=1, padx=5, pady=5)
59
60 button_3 = tk.Button(root, text="3", font=('Arial bold', 15), width=5, height=2, command=lambda: button_click(3))
```



```
59 button_2.grid(row=3, column=1, padx=5, pady=5)
60 button_3 = tk.Button(root, text="3", font=('Arial bold', 15), width=5, height=2, command=lambda: button_click(3))
61 button_3.grid(row=3, column=2, padx=5, pady=5)
62
63 button_0 = tk.Button(root, text="0", font=('Arial bold', 15), width=5, height=2, command=lambda: button_click(0))
64 button_0.grid(row=4, column=0, padx=5, pady=5)
65
66 button_plus = tk.Button(root, text="+", font=('Arial bold', 15), width=5, height=2, command=lambda: button_click("+"))
67 button_plus.grid(row=1, column=3, padx=5, pady=5)
68
69 button_minus = tk.Button(root, text="-", font=('Arial bold', 15), width=5, height=2, command=lambda: button_click("-"))
70 button_minus.grid(row=2, column=3, padx=5, pady=5)
71
72 button_multiply = tk.Button(root, text="x", font=('Arial bold', 15), width=5, height=2, command=lambda: button_click("*"))
73 button_multiply.grid(row=3, column=3, padx=5, pady=5)
74
75 button_divide = tk.Button(root, text="/", font=('Arial bold', 15), width=5, height=2, command=lambda: button_click("/"))
76 button_divide.grid(row=4, column=3, padx=5, pady=5)
77
78 button_equal = tk.Button(root, text="=", font=('Arial bold', 15), bg="pink", width=5, height=2, command=calculate)
79 button_equal.grid(row=4, column=2, padx=5, pady=5)
80
81 button_clear = tk.Button(root, text="C", font=('Arial bold', 15), bg="yellow", width=5, height=2, command=clear)
82 button_clear.grid(row=4, column=1, padx=5, pady=5)
83
```

The calculator button **0 – 9** and **+, -, /** buttons are created using the `tk.button()` method and placed in the grid layout.



```
84 #main eka
85 root.mainloop()
```

This line starts the Tkinter event loop, which keeps the application running and responsive. It waits for user interactions like button clicks.

References

- https://youtu.be/NzSCNjn4_RI?si=rpzf-5VEDWjWPmqx – help this video
- <https://youtu.be/QZPv1y2znZo?si=qV935jazWoJanEw9> - main box made
- <https://youtu.be/6CZB6VTy3Hg?si=ErvOiJQjst8fjhR-> - mainly use video