Calculator

March 8, 2025

1 CALCULATOR PROJECT

1.1 NAME:Elham madani sarighiyeh

1.2 DATE: 8/3/2025

1.2.1 GIT HUB REPOSITORY: https://github.com/Elhammadani/CALCULATOR-FINAL-EXAM.git

1.2.2 DESCRIPTION: This codes is about calculator program. Also this code run in jupyter and opening calculator app in operating system (windows). After that i put calculator image on power point.

[23]:
```
from tkinter import *
import math

root = Tk()
root.title("Calculator")
root.geometry("450x600+500+40")
root.resizable(True, True)  # Allow automatic resizing

calc = Frame(root, bd=20, pady=5, bg='gainsboro', relief=RIDGE)
calc.pack(fill=BOTH, expand=True)

class Calc:
    def __init__(self):
        self.total = 0
        self.current = ""
        self.input_value = True
        self.check_sum = False
        self.op = ""
        self.result = False

    def number_enter(self, num):
        self.result = False
        firstnum = txtDisplay.get()
        secondnum = str(num)
        if self.input_value:
            self.current = secondnum
```

```
            self.input_value = False
        else:
            if secondnum == '.' and '.' in firstnum:
                return
            self.current = firstnum + secondnum
        txtDisplay.delete(0, END)
        txtDisplay.insert(0, self.current)

    def operation(self, op):
        self.current = float(self.current)
        if self.check_sum:
            self.valid_function()
        else:
            self.total = self.current
        self.input_value = True
        self.op = op
        self.check_sum = True

    def valid_function(self):
        if self.op == "+":
            self.total += self.current
        elif self.op == "-":
            self.total -= self.current
        elif self.op == "*":
            self.total *= self.current
        elif self.op == "/":
            self.total /= self.current
        txtDisplay.delete(0, END)
        txtDisplay.insert(0, self.total)

    def result_function(self):
        self.current = float(self.current)
        if self.check_sum:
            self.valid_function()
        self.check_sum = False
        self.input_value = True

    def clear_entry(self):
        self.current = ""
        self.total = 0
        txtDisplay.delete(0, END)
        self.input_value = True

    def square(self):
        self.current = float(self.current) ** 2
        txtDisplay.delete(0, END)
        txtDisplay.insert(0, self.current)
```

```
    def square_root(self):
        self.current = math.sqrt(float(self.current))
        txtDisplay.delete(0, END)
        txtDisplay.insert(0, self.current)

    def backspace(self):
        self.current = txtDisplay.get()[:-1]
        txtDisplay.delete(0, END)
        txtDisplay.insert(0, self.current)

    def plus_minus(self):
        self.current = str(-1 * float(txtDisplay.get()))
        txtDisplay.delete(0, END)
        txtDisplay.insert(0, self.current)

added_value = Calc()

# Entry widget
txtDisplay = Entry(calc, font=('arial', 50, 'bold'), bd=20, justify=RIGHT)
txtDisplay.grid(row=0, column=0, columnspan=4, pady=1, sticky=NSEW)

def exit_calc():
    root.destroy()

# Buttons (Reordered for standard calculator layout)
buttons = [
    ('C', 1, 0), ('←', 1, 1), ('+/-', 1, 2), ('/', 1, 3),
    ('7', 2, 0), ('8', 2, 1), ('9', 2, 2), ('*', 2, 3),
    ('4', 3, 0), ('5', 3, 1), ('6', 3, 2), ('-', 3, 3),
    ('1', 4, 0), ('2', 4, 1), ('3', 4, 2), ('+', 4, 3),
    ('0', 5, 0), ('.', 5, 1), ('=', 5, 2), ('Exit', 5, 3),
    ('x²', 6, 0), ('√', 6, 1)
]

def create_buttons():
    for (text, row, col) in buttons:
        if text.isdigit() or text == '.':
            action = lambda x=text: added_value.number_enter(x)
        elif text in ['+', '-', '*', '/']:
            action = lambda x=text: added_value.operation(x)
        elif text == '=':
            action = added_value.result_function
        elif text == 'C':
            action = added_value.clear_entry
        elif text == 'x²':
            action = added_value.square
```

```
        elif text == '√':
            action = added_value.square_root
        elif text == 'Exit':
            action = exit_calc
        elif text == '←':
            action = added_value.backspace
        elif text == '+/-':
            action = added_value.plus_minus
        Button(calc, text=text, font=('arial', 20, 'bold'), height=2, width=9,
               bd=4, command=action).grid(row=row, column=col, pady=5,␣
 ↪sticky=NSEW)
for i in range(7):
    calc.grid_rowconfigure(i, weight=1)
    calc.grid_columnconfigure(i % 4, weight=1)
txtDisplay.grid_columnconfigure(0, weight=1)  # Make display text adjustable
txtDisplay.config(font=('arial', 50, 'bold'))  # Increase text size
create_buttons()
root.mainloop()
```

[ ]: