Republic of the Philippines
CAVITE STATE UNIVERSITY
Don Severino delas Alas Campus
Indang, Cavite

OBJECT-ORIENTED PROGRAMMING

Laboratory Activity No. 4
The Grid Manager

Submitted by:
de Guzman, Kinlie Venice L. – Leader
Causaren, Jan Rovick M. – Member
Lorica, Landon S. – Member
Quijano, Colleen M. – Member
Rebulado, Sarah O. – Member
Umipig, Ian Rafael - Member
Valenzuela, Mat Lauren Keiser R. – Member

Thurs 1:00pm - 4:00pm / BSCpE 1-2

Date Submitted
19-05-2022

Submitted to:
Engr. Maria Rizette H. Sayo

I. Objectives

The goal of this activity is the ff:

- By using Python GUI, create a standard calculator program by using a grid manager.
- The created calculator program using grid manager should have a textbox to where the values and operation be inserted. The Standard Calculator can have a numpad with symbols for operations, equalization, and to note decimal on it.
- The created calculator program using grid manager should contain the four operations and the result should still be exact even though the user pressed the specific operation repeatedly.

II. Methods

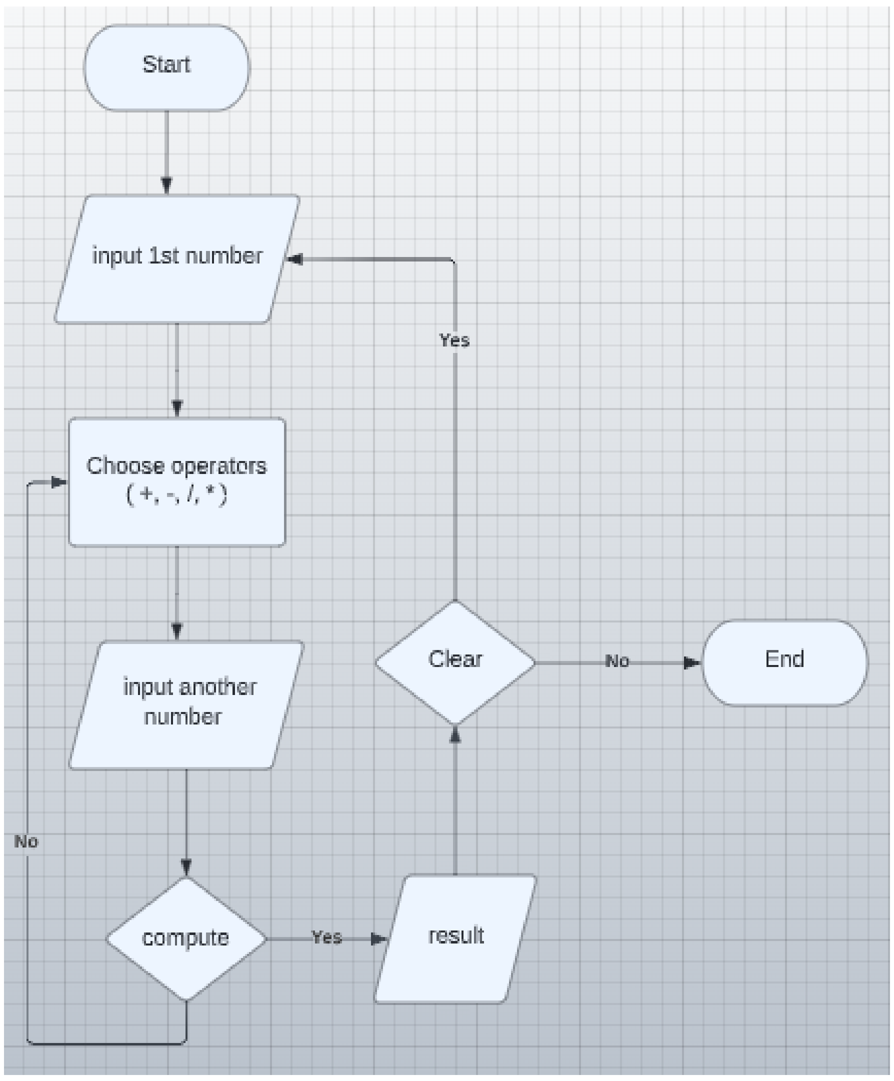


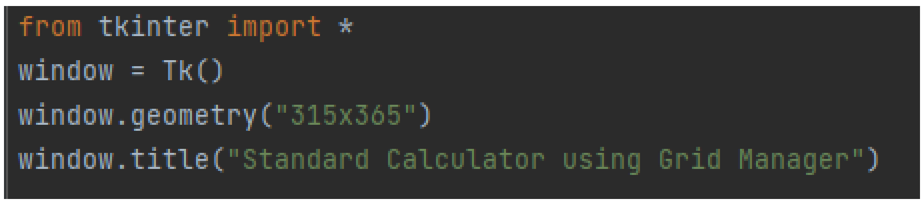
Figure 1. Standard Calculator Flowchart

We used a GUI toolkit that runs in python, the “TK Interface” , which is the most basic interface that can be used and available in different systems such as in UNIX platforms, MacOS, and especially on Windows.

Creation of New Window

First, we will start to initialize a code that creates a new window, this code will serve as the base for our program.

To create a basic window, we used the code:



```
from tkinter import *
window = Tk()
window.geometry("315x365")
window.title("Standard Calculator using Grid Manager")
```

Figure 2. Codes for the Creation of Window.

Defining the variables

Second, we started to define the variables that we will be needing to run the program smoothly throughout the process.

In this data, We define three (3) variables for different purposes, it is for the input button, clear button, and equal button. We also used “global” this will serve as a caller of the variable so that the whole program will run, “.set” for retrieving the information of the variable, and “eval” this function is for input typed in the input field that string convert to sentence and sentence convert to string.

These are the codes that are used:

1. def input_btn (item)

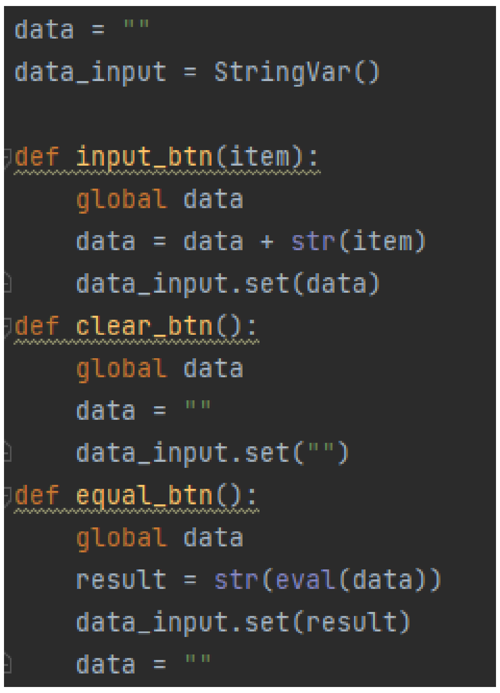
   This function continuously updates the input field whenever you enter a number.

2. def clear_btn()

   This function is used to clear the input field

3. def equal_btn()

   This function is used to calculate the expression present in the input field.



```
data = ""
data_input = StringVar()

def input_btn(item):
    global data
    data = data + str(item)
    data_input.set(data)
def clear_btn():
    global data
    data = ""
    data_input.set("")
def equal_btn():
    global data
    result = str(eval(data))
    data_input.set(result)
    data = ""
```

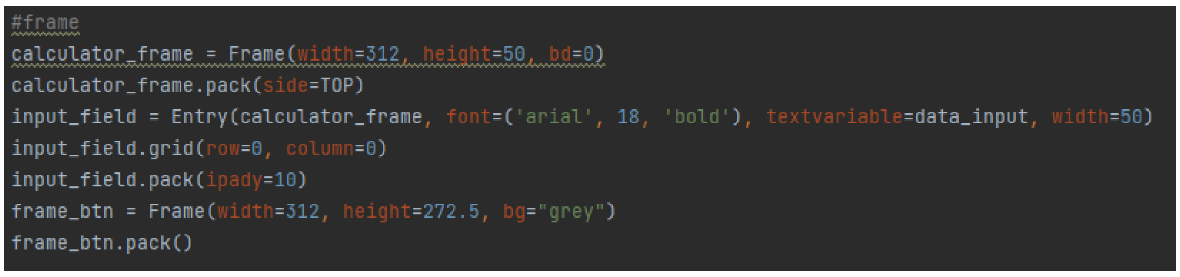
Figure 3. Codes for Defining Variables.

Creation of Frame

Third, we started to create a code for our frame, the frame will serve as the input field of our program. In this part, we indicated the elements of our programs such as font, font size, frame button, and colors to avoid confusions for the user.

The calcutor_frame is to set the frame of the input field so that it can have a space where the inputted values can be showed. While the .grid is used to locate on what position should be the grid placed. The frame_btn is use to size the grid of the symbols and numbers of the calculator and by using the .pack, a python GUI package, we can fix the widgets or the button itself.

To create the frame, we used these codes:



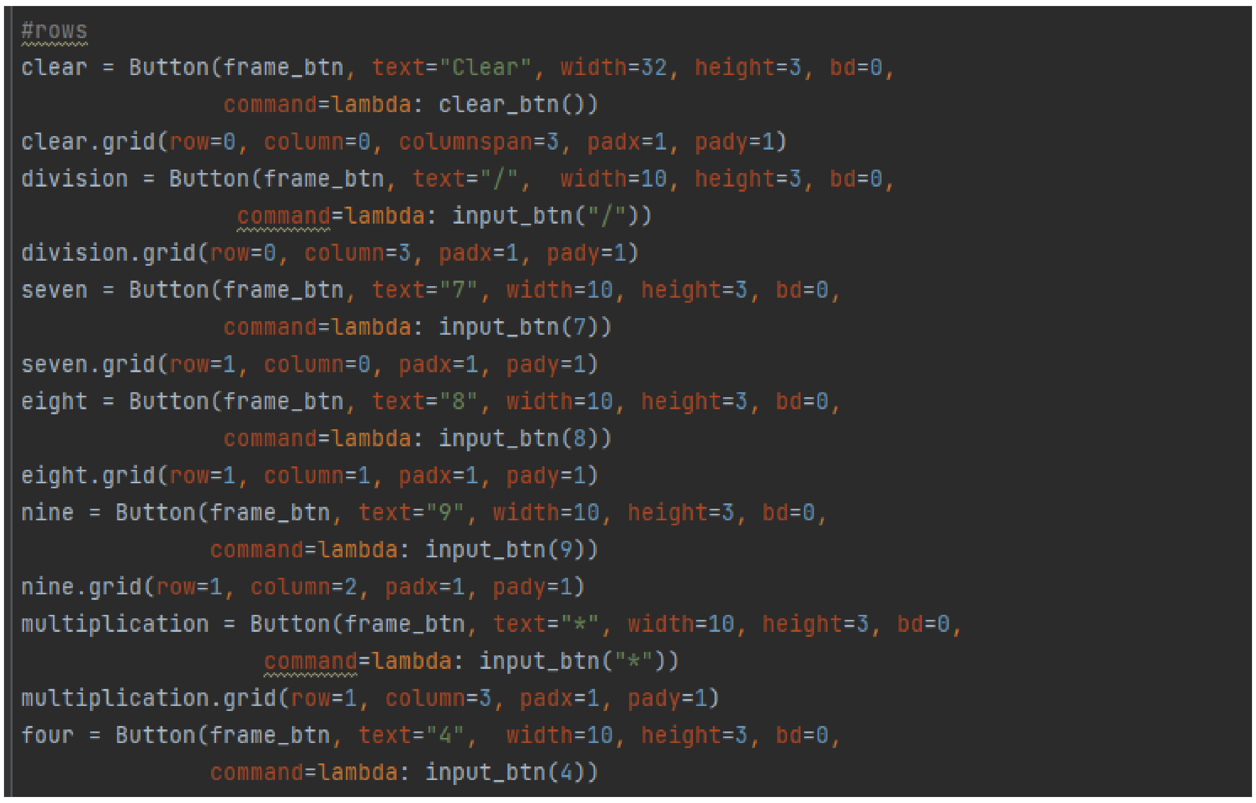
```
#frame
calculator_frame = Frame(width=312, height=50, bd=0)
calculator_frame.pack(side=TOP)
input_field = Entry(calculator_frame, font=('arial', 18, 'bold'), textvariable=data_input, width=50)
input_field.grid(row=0, column=0)
input_field.pack(ipady=10)
frame_btn = Frame(width=312, height=272.5, bg="grey")
frame_btn.pack()
```

Figure 4. Codes for the Creation of the Frame.

Creation of Rows and Symbols

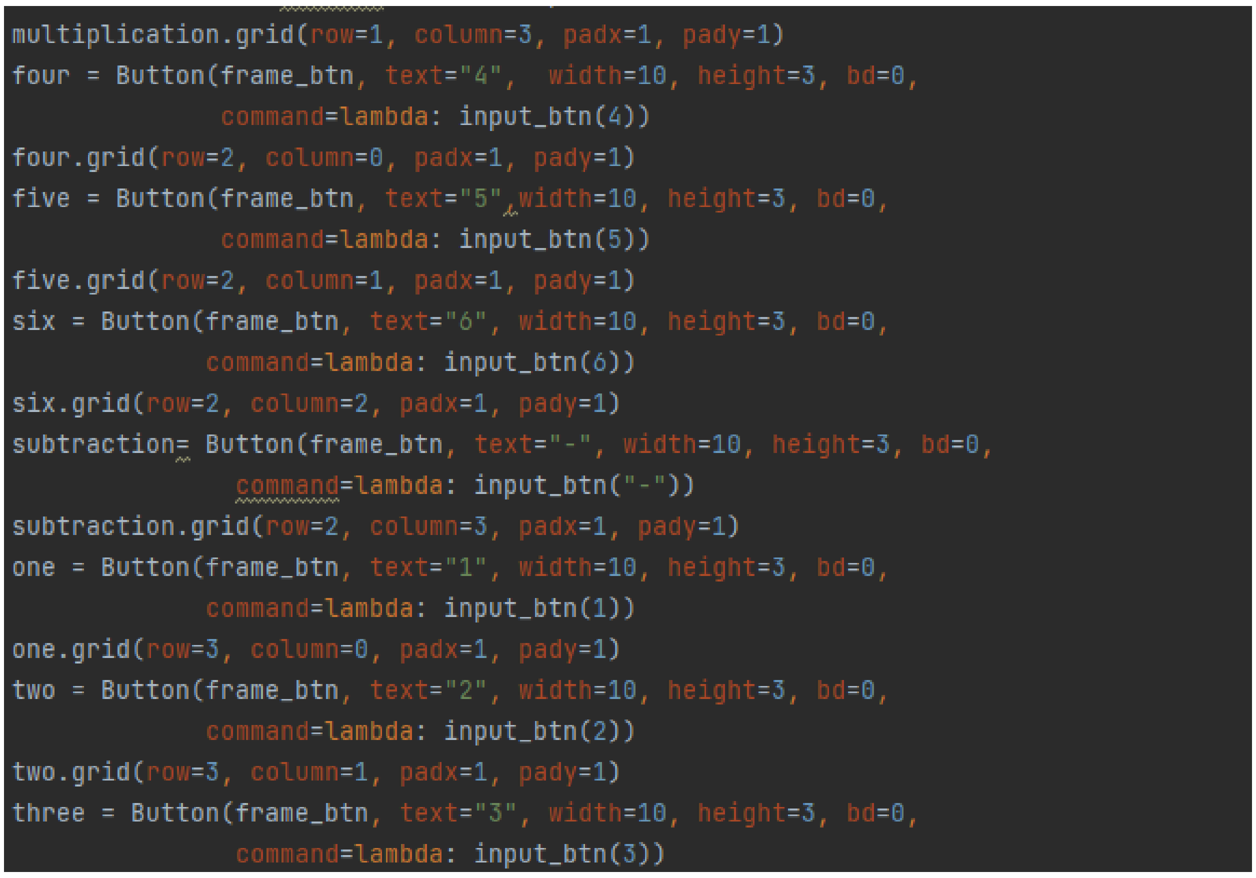
Last, this part of the code is for the numbers and operators that will be used for computing. We indicated specific numbers and symbols for the operators in order to run the code smoothly.

There are some unfamiliar codes that we used for our program like "lambda" for the unnamed function so that the values will be displayed in the input field, ".grid" is for the section of rows and symbol to place the button in the correct spot of the grid in the window. The codes that are used:



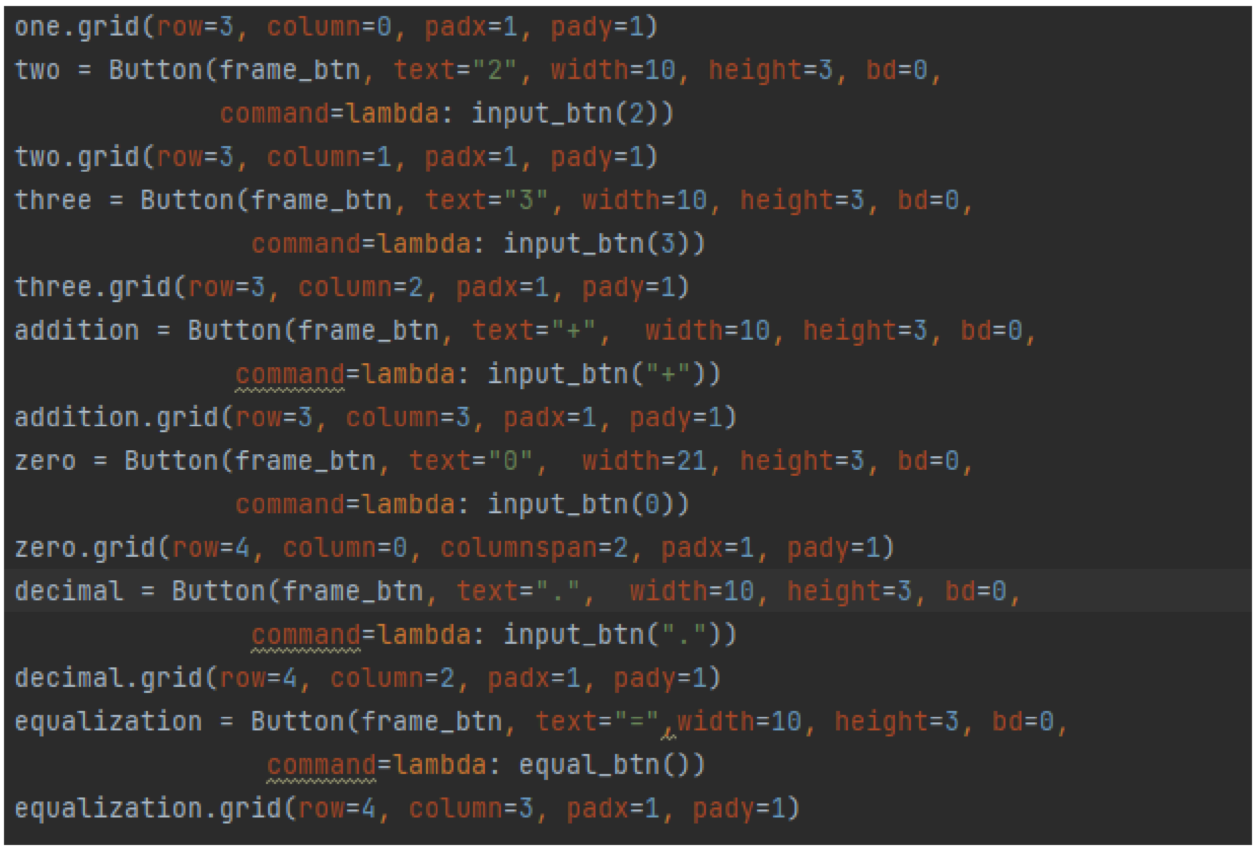
```
#rows
clear = Button(frame_btn, text="Clear", width=32, height=3, bd=0,
               command=lambda: clear_btn())
clear.grid(row=0, column=0, columnspan=3, padx=1, pady=1)
division = Button(frame_btn, text="/",  width=10, height=3, bd=0,
                command=lambda: input_btn("/"))
division.grid(row=0, column=3, padx=1, pady=1)
seven = Button(frame_btn, text="7", width=10, height=3, bd=0,
               command=lambda: input_btn(7))
seven.grid(row=1, column=0, padx=1, pady=1)
eight = Button(frame_btn, text="8", width=10, height=3, bd=0,
               command=lambda: input_btn(8))
eight.grid(row=1, column=1, padx=1, pady=1)
nine = Button(frame_btn, text="9", width=10, height=3, bd=0,
              command=lambda: input_btn(9))
nine.grid(row=1, column=2, padx=1, pady=1)
multiplication = Button(frame_btn, text="*", width=10, height=3, bd=0,
                        command=lambda: input_btn("*"))
multiplication.grid(row=1, column=3, padx=1, pady=1)
four = Button(frame_btn, text="4",  width=10, height=3, bd=0,
              command=lambda: input_btn(4))
```

Figure 5.1. Codes for the Creation of Rows and Symbols.



```
multiplication.grid(row=1, column=3, padx=1, pady=1)
four = Button(frame_btn, text="4",  width=10, height=3, bd=0,
              command=lambda: input_btn(4))
four.grid(row=2, column=0, padx=1, pady=1)
five = Button(frame_btn, text="5",width=10, height=3, bd=0,
              command=lambda: input_btn(5))
five.grid(row=2, column=1, padx=1, pady=1)
six = Button(frame_btn, text="6", width=10, height=3, bd=0,
             command=lambda: input_btn(6))
six.grid(row=2, column=2, padx=1, pady=1)
subtraction= Button(frame_btn, text="-", width=10, height=3, bd=0,
               command=lambda: input_btn("-"))
subtraction.grid(row=2, column=3, padx=1, pady=1)
one = Button(frame_btn, text="1", width=10, height=3, bd=0,
             command=lambda: input_btn(1))
one.grid(row=3, column=0, padx=1, pady=1)
two = Button(frame_btn, text="2", width=10, height=3, bd=0,
             command=lambda: input_btn(2))
two.grid(row=3, column=1, padx=1, pady=1)
three = Button(frame_btn, text="3", width=10, height=3, bd=0,
               command=lambda: input_btn(3))
```

Figure 5.2. Codes for the Creation of Rows and Symbols.



```
one.grid(row=3, column=0, padx=1, pady=1)
two = Button(frame_btn, text="2", width=10, height=3, bd=0,
             command=lambda: input_btn(2))
two.grid(row=3, column=1, padx=1, pady=1)
three = Button(frame_btn, text="3", width=10, height=3, bd=0,
               command=lambda: input_btn(3))
three.grid(row=3, column=2, padx=1, pady=1)
addition = Button(frame_btn, text="+",  width=10, height=3, bd=0,
              command=lambda: input_btn("+"))
addition.grid(row=3, column=3, padx=1, pady=1)
zero = Button(frame_btn, text="0",  width=21, height=3, bd=0,
              command=lambda: input_btn(0))
zero.grid(row=4, column=0, columnspan=2, padx=1, pady=1)
decimal = Button(frame_btn, text=".",  width=10, height=3, bd=0,
               command=lambda: input_btn("."))
decimal.grid(row=4, column=2, padx=1, pady=1)
equalization = Button(frame_btn, text="=",width=10, height=3, bd=0,
                command=lambda: equal_btn())
equalization.grid(row=4, column=3, padx=1, pady=1)
```

Figure 5.3. Codes for the Creation of Rows and Symbol.

III. Results

In this chapter the results of the study are presented and discussed with reference to the aim of the project, which was to create a Standard Calculator that applies the Grid method. The two sub-aims - first, to show the visualization and effectiveness of the programmed code through the widget, standard calculator, performing the four basic operations and second is to compare the programmed standard calculator to Microsoft's built-in calculator.

The visualization of the widget

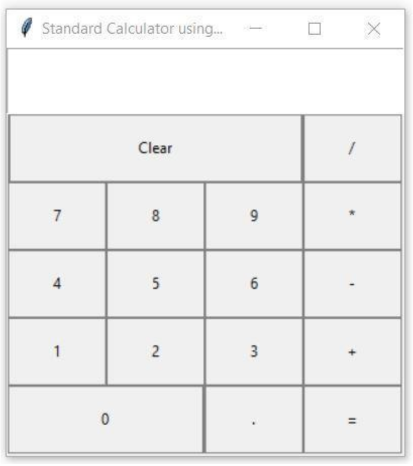


Figure 6. Standard Calculator using the Grid Geometry Manager

Figure 6 shows the widget standard calculator using the grid geometry manager, this calculator is made in relevance with other standard calculator's built and appearance as well as its basic functionalities.

Parts of the calculator

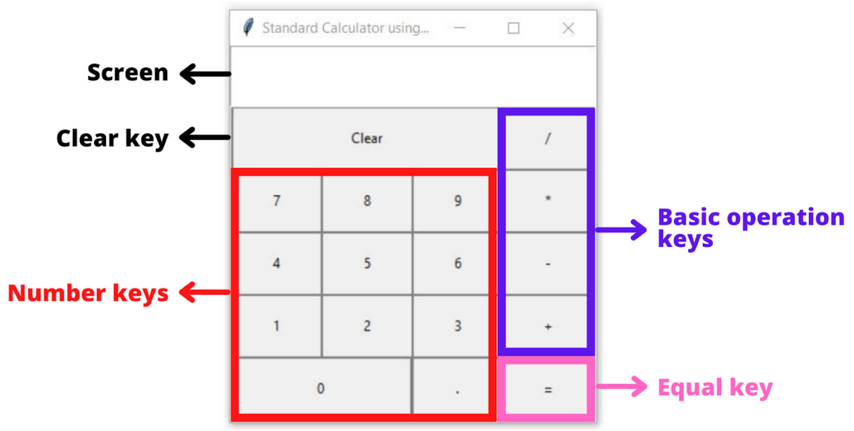


Figure 7. Parts of the standard calculator

Parts of the programmed standard calculator:

- Screen - show the numbers being input and operators used, as well as the result.
- Clear key - Clear all the entries made.
- Number keys - includes the numbers, decimal points and operators.
- Basic operation keys - the four main operators; add, subtract, multiply, divide
- Equal key - button that leads to the result of the operations used.

Figure 7 illustrates the parts and function of the standard calculator which includes the four basic operations in each button, addition, subtraction, multiplication and division operators respectively, also included the functions such as clear function and equal sign.

Test the widget

Result shows that the standard calculator has no limitations when it comes to the number of digits it can hold. Moreover, the calculator functions by using the computer keyboard or by pressing the buttons on the actual widget. All widget buttons function properly and exhibit accurate results. The final computation shown on the calculator screen isn't rounded off, instead it presents all the decimals for an exact outcome. However, the test reveals that the enter key unfortunately does not work as an equal key as it results in an “ERROR" indication. Therefore, the ‘equal key’ on the widget must be clicked to get an answer on the calculator screen.

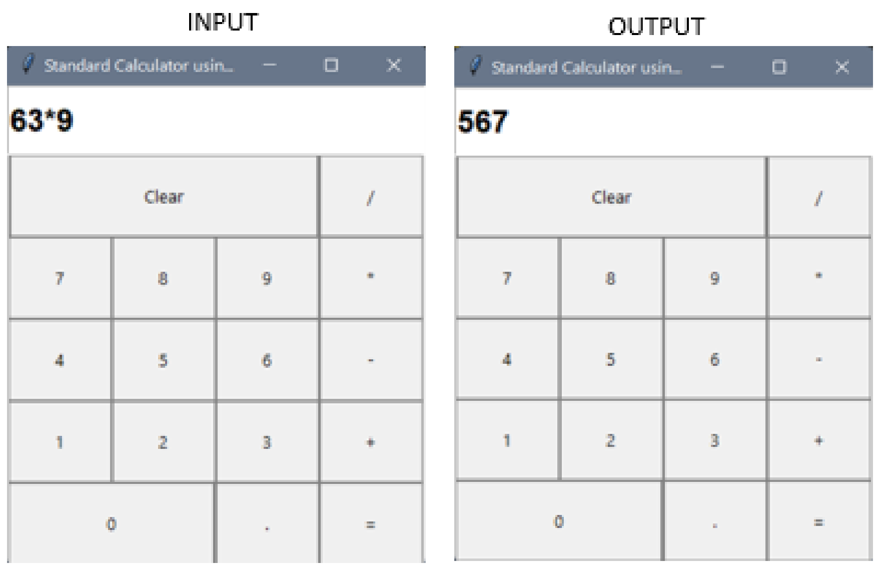


Figure 8. Testing the calculator using the multiplication operator

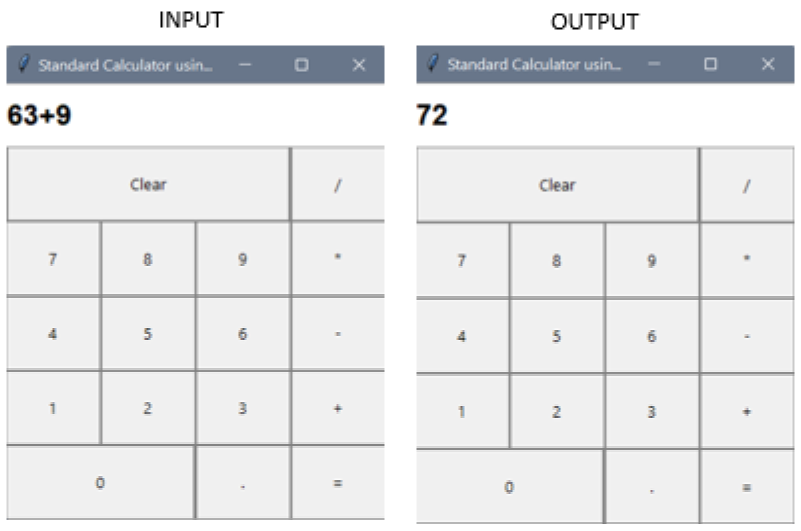
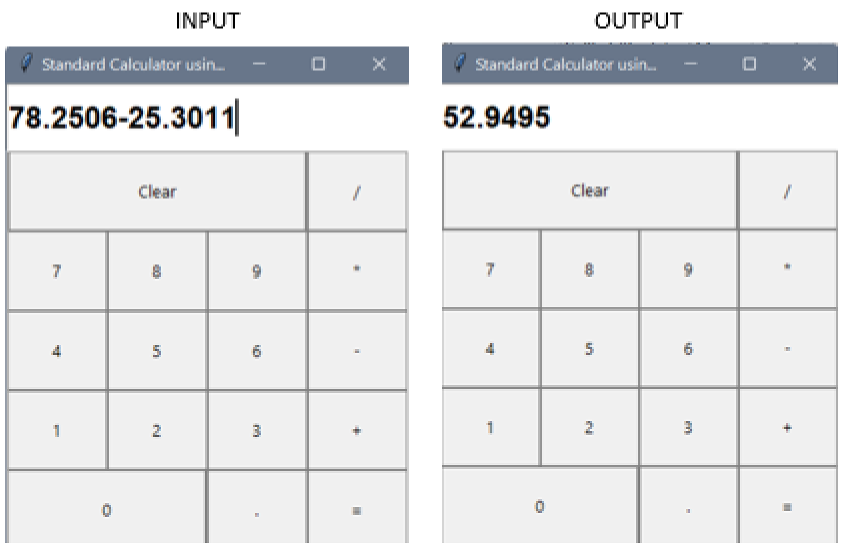


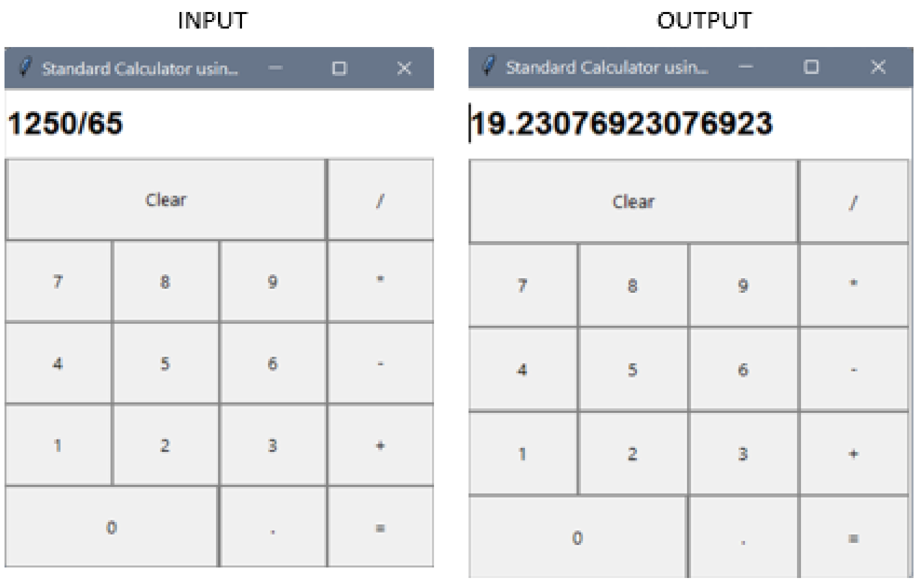
Figure 9. Testing the calculator using the addition operator



```
78.2506-25.3011
```

```
52.9495
```

Figure 10. Testing the calculator using subtraction operator and application of decimal

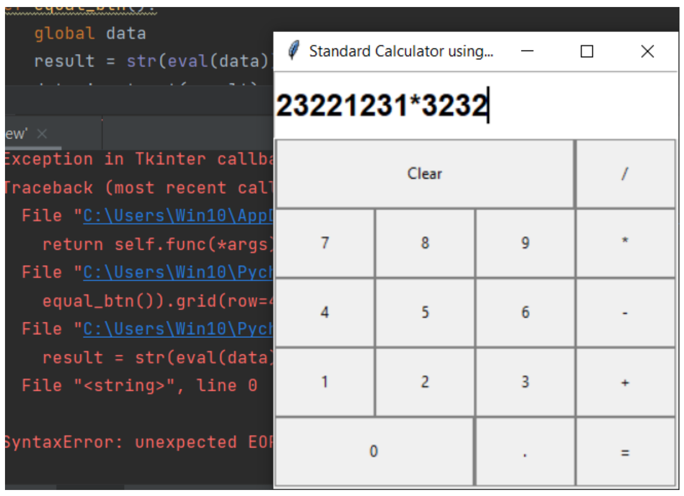


```
1250/65
```

```
19.23076923076923
```

Figure 11. Testing the calculator using division operator

Figures 8,9,10,11, shows the visualization and effectiveness of the programmed code through the widget, standard calculator, performing the four basic operations. All these results were shown by pressing the buttons programmed and not the computer keyboard.



```
  global data
  result = str(eval(data)
```

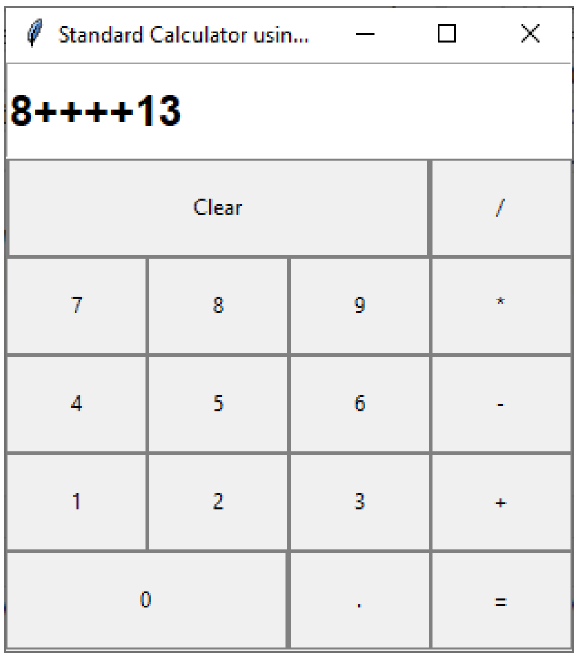
```
xception in Tkinter callb
raceback (most recent cal
 File "C:\Users\Win10\App
   return self.func(*args
 File "C:\Users\Win10\Pyc
   equal_btn()).grid(row=
 File "C:\Users\Win10\Pyc
   result = str(eval(data
 File "<string>", line 0

yntaxError: unexpected EO
```

```
23221231*3232
```

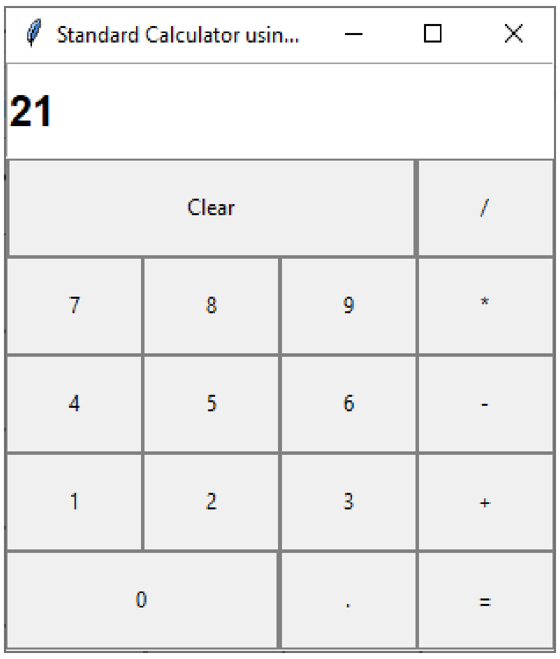
Figure 12. Testing using computer keyboard and pressing enter key for results

Figure 12, demonstrated the limited specification in which the standard calculator can function even if the buttons are not pressed, instead by inputting numbers using the computer keyboard.

However, it shows that the enter key does not work as the equal key. In order to get the result, the equal key must be pressed or else, the program will show an error.


Standard Calculator usin...

```
8++++13
```


Standard Calculator usin...

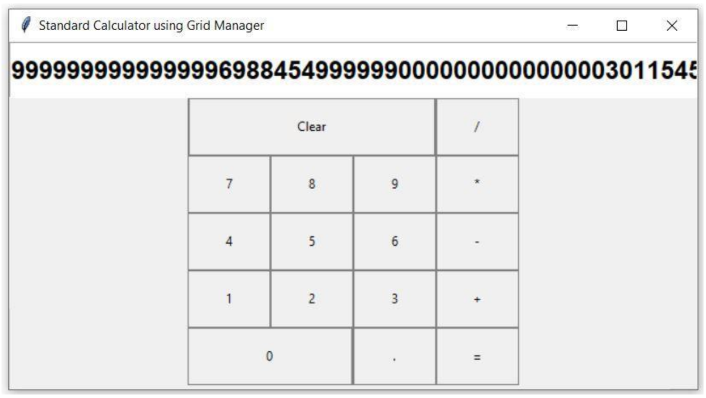
```
21
```

Figure 13. Display of answer with multiple presses of the same operation. (input & output)

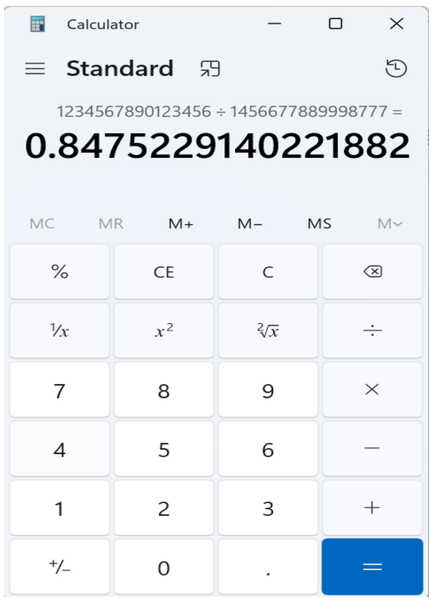
Figure 13, demonstrated that even though the operation is repeatedly pressed, the answer will still be the same since one operation is only included..

Compare the accuracy with other calculator

Comparison between the programmed widget (standard calculator) vs built-in Microsoft calculator


Standard Calculator using Grid Manager

```
99999999999999698845499999900000000000003011545
```


Calculator

Standard

```
1234567890123456 ÷ 1456677889998777 =
0.8475229140221882
```

Figure 14. Compare the accuracy with other calculator.

Compared to the built-in Microsoft Calculator, the results are the same with the programmed standard calculator. However, the Microsoft Calculator is only limited to 16 digits while the programmed standard calculator has no limits, which decreases its accuracy. This is due to the fact that the Standard calculator is resizeable, so the screen has no limit in showing the complete decimals unlike the screen of the built- in microsoft calculator that has limited space. It also lacks the functionality of accessibility in terms of ease for the user experience, a given situation is the lack of function to the keyboard input, wherein when the user input using a keyboard and press enter to show the results, it errors and needs to reset the operation by manually clicking the clear button. Unlike the built-in calculator that saves the preview's answer and can continue computing the additional numbers inputted, the standard calculator can only provide a one way solution. Once the 'equal key' is pressed, no more additional numbers can be added and computed along with the most recent outcome on screen as the calculator automatically resets.

Moreover, user friendly wise, the built-in Microsoft Calculator has a clear function for each digit individually, unlike the standard calculator where the numbers must be cleared all together if a mistake is made. And lastly, the built-in Microsoft Calculator can function purely by using the computer keyboard in which in the standard calculator, the equal key must be pressed as the enter key causes error in the programmed calculator. Overall, the built-in Microsoft Calculator has more functions than the standard calculator created that can do only addition, subtraction, multiplication, and division.

IV. Conclusion

Analyzing the results, the goal was achieved by making a simple calculator using a grid method. The calculator made was comparable to the standard calculator we have in markets, the built-in calculator in Microsoft that we have, and sari-sari stores due to its limited functions. This calculator lacks the features of the calculators we use right now since the only thing that was added in this task is the simple operations. Also, while inspecting the widget, it is evident that the parts that makes a standard calculator functionable are present. During the testing of the widget, the calculator has done its purpose by computing numbers inputted by the user itself and it displays the correct answer. The only minor issue is that the “Enter” key cannot be substituted as an equal to the mathematical problem written by the user.

Ergo, we can conclude that the study attained the following results:

a. By using the Python GUI, it is possible to make a standard calculator by using a Grid Manager.
b. The created calculator program using grid manager displayed a textbox and it is clearly visible that the values and operations can be inserted on the window on the upper part of the calculator
c. The created calculator program using grid manager contained the four operations and the result was exact even though the specific operation was pressed repeatedly.

Reference

Website

[1]"Calculator Application using Tkinter (Python Project) - Studytonight," *www.studytonight.com*. https://www.studytonight.com/tkinter/calculator-application-using-tkinter (accessed May 15, 2022).

[2]"How to Create a Calculator in Python Tkinter ActiveState How to Create a Calculator in Python Tkinter -," *ActiveState*. https://www.activestate.com/resources/quick-reads/how-to-create-a-calculator-in-python-tkinter/ (accessed May 15, 2022).

[3]"Simple GUI Calculator in Python," *www.youtube.com*. https://www.youtube.com/watch?fbclid=IwAR0o9ceWHZvUlZBcfU6ZhhMasZ672_g7fdpqK0xZeglYGEnbsSVBAnJGu0w&v=NzSCNjn4_RI&feature=youtu.be (accessed May 15, 2022).

[4]"tkinter — Python interface to Tcl/Tk — Python 3.10.4 documentation," *docs.python.org*. https://docs.python.org/3/library/tkinter.html?fbclid=IwAR0oyScD1QnjB0qCa7X8Cz_u_UB7H7F_msVsiQnXA2Yvl5VAAuR2YEhBIlo (accessed May 15, 2022).