**Batch: A1 Roll No.: 1911004**

**Experiment / assignment / tutorial No.**

**Grade: AA / AB / BB / BC / CC / CD /DD**

**Signature of the Staff In-charge with date**

|  |
| --- |
| **TITLE:** Demonstrate the use of GUI Widgets to create Scientific Calculator (Min. 10 functionalities). |

**AIM:** Implementation of Python GUI Application with Tkinter.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Expected OUTCOME of Experiment:**

CO 4 - Explain how to design GUI Applications in Python and evaluate different database operations

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Books/ Journals/ Websites referred:**

1. Tkinter - GUI notes

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Pre Lab/ Prior Concepts:**

Tkinter is the standard GUI library for Python. Python when combined with Tkinter provides a fast and easy way to create GUI applications. Tkinter provides a powerful object-oriented interface to the Tk GUI toolkit.

Creating a GUI application using Tkinter is an easy task. All you need to do is perform the following steps −

* Import the Tkinter module.
* Create the GUI application main window.
* Add one or more of the above-mentioned widgets to the GUI application.
* Enter the main event loop to take action against each event triggered by the user

Example:

import Tkinter

top = Tkinter.Tk()

# Code to add widgets for gui will go here...

top.mainloop()

Following are 19 Widget

|  |  |
| --- | --- |
| **Widget** | **Explanation** |
| Button | The Button widget is used to display buttons in your application. |
| Canvas | The Canvas widget is used to draw shapes, such as lines, ovals, polygons and rectangles, in your application. |
| CheckButton | The Checkbutton widget is used to display a number of options as checkboxes. The user can select multiple options at a time. |
| Entry | The Entry widget is used to display a single-line text field for accepting values from a user. |
| Frame | The Frame widget is used as a container widget to organize other widgets. |
| Label | The Label widget is used to provide a single-line caption for other widgets. It can also contain images. |
| ListBox | The Listbox widget is used to provide a list of options to a user. |
| MenuButton | The Menubutton widget is used to display menus in your application. |
| Menu | The Menu widget is used to provide various commands to a user. These commands are contained inside Menubutton. |
| Message | The Message widget is used to display multiline text fields for accepting values from a user. |
| RadioButton | The Radiobutton widget is used to display a number of options as radio buttons. The user can select only one option at a time. |
| Scale | The Scale widget is used to provide a slider widget. |
| Scrollbar | The Scrollbar widget is used to add scrolling capability to various  widgets, such as list boxes. |
| Text | The Text widget is used to display text in multiple lines. |
| TopLevel | The Toplevel widget is used to provide a separate window container. |
| SpinBox | The Spinbox widget is a variant of the standard Tkinter Entry widget, which can be used to select from a fixed number of values. |
| PanelWindow | A PanedWindow is a container widget that may contain any number of panes, arranged horizontally or vertically. |
| LabelFrame | A labelframe is a simple container widget. Its primary purpose is to act as a spacer or container for complex window layouts. |
| TkMessageBox | This module is used to display message boxes in your applications. |

**Program Problem statement** :

Demonstrate the use of GUI Widgets to create Scientific Calculator

**Code:**\_

from tkinter import \*

import math as math

tk = Tk()

tk.title("Scientific Calculator")

tk.configure(background='white')

tk.resizable(width=False, height=False)

tk.geometry("530x487")

calc = Frame(tk)

calc.grid()

equation = StringVar()

txtDisplay = Entry(calc, font=('Arial',18, 'bold'), bg='black', fg='white', bd=6,width=30, justify=RIGHT,textvariable=equation)

txtDisplay.grid(row=0, column=0, columnspan=4, pady=1)

expression = ''

lastnum=''

sop=0

def scalc(num,lastnum):

num=float(num)

if lastnum=='sin':

return (math.sin(num))

elif lastnum=='cos':

return math.cos(num)

elif lastnum=='tan':

return math.tan(num)

elif lastnum=='log':

return math.log(num)

elif lastnum=='abs':

return math.fabs(num)

else:

return " Error "

def getval(num):

global expression

global sop

if num in ['sin', 'cos', 'tan', 'log', 'abs']:

sop=num

expression = ""

equation.set(num)

else:

expression = expression + str(num)

equation.set(expression)

def equals():

global expression

global sop

if sop:

try:

total=str(scalc(expression,sop))

if total==" Error ":

equation.set(" Error ")

expression = ""

else:

equation.set(total)

expression = ""

except:

equation.set(" Error ")

expression = ""

sop=0

else:

try:

total = str(eval(expression))

print(total)

equation.set(total)

expression = ""

except:

equation.set(" Error ")

expression = ""

def clear():

global expression

expression = ""

equation.set("")

btn7= Button(calc, width=6, height=2, bg='white', fg='black',font=('Arial', 18, 'bold'),command=lambda:getval('7'),text='7').grid(row=2, column=0, pady=1)

btn8= Button(calc, width=6, height=2, bg='white', fg='black',font=('Arial', 18, 'bold'),command=lambda:getval('8'),text='8').grid(row=2, column=1, pady=1)

btn9= Button(calc, width=6, height=2, bg='white', fg='black',font=('Arial', 18, 'bold'),command=lambda:getval('9'),text='9').grid(row=2, column=2, pady=1)

btn4= Button(calc, width=6, height=2, bg='white', fg='black',font=('Arial', 18, 'bold'),command=lambda:getval('4'),text='4').grid(row=3, column=0, pady=1)

btn5= Button(calc, width=6, height=2, bg='white', fg='black',font=('Arial', 18, 'bold'),command=lambda:getval('5'),text='5').grid(row=3, column=1, pady=1)

btn6= Button(calc, width=6, height=2, bg='white', fg='black',font=('Arial', 18, 'bold'),command=lambda:getval('6'),text='6').grid(row=3, column=2, pady=1)

btn1= Button(calc, width=6, height=2, bg='white', fg='black',font=('Arial', 18, 'bold'),command=lambda:getval('1'),text='1').grid(row=4, column=0, pady=1)

btn2= Button(calc, width=6, height=2, bg='white', fg='black',font=('Arial', 18, 'bold'),command=lambda:getval('2'),text='2').grid(row=4, column=1, pady=1)

btn3= Button(calc, width=6, height=2, bg='white', fg='black',font=('Arial', 18, 'bold'),command=lambda:getval('3'),text='3').grid(row=4, column=2, pady=1)

btnsin = Button(calc, text="sin", width=6, height=2, bg='white',fg='black', font=('Arial', 18, 'bold'),command=lambda:getval('sin')).grid(row=1, column=0)

btncos = Button(calc, text="cos", width=6, height=2, bg='white',fg='black', font=('Arial', 18, 'bold'),command=lambda:getval('cos')).grid(row=1, column=1)

btntan = Button(calc, text="tan", width=6, height=2, bg='white',fg='black', font=('Arial', 18, 'bold'),command=lambda:getval('tan')).grid(row=1, column=2)

btnce = Button(calc, text="CE", width=6, height=2, bg='white',fg='black', font=('Arial', 18, 'bold'),command=clear).grid(row=1,column=3)

btndiv = Button(calc, text="/", width=6, height=2, bg='white',fg='black', font=('Arial', 18, 'bold'),command=lambda:getval('/')).grid(row=2, column=3)

btnmul = Button(calc, text="\*", width=6, height=2, bg='white',fg='black', font=('Arial', 18, 'bold'),command=lambda:getval('\*')).grid(row=3, column=3)

btnsub = Button(calc, text="-", width=6, height=2, bg='white',fg='black', font=('Arial', 18, 'bold'),command=lambda:getval('-')).grid(row=4, column=3)

btneq = Button(calc, text="=", width=6, height=2, bg='white',fg='black', font=('Arial', 18, 'bold'),command=equals).grid(row=5, column=3)

btnadd = Button(calc, text="+", width=6, height=2, bg='white',fg='black', font=('Arial', 18, 'bold'),command=lambda:getval('+')).grid(row=5, column=2)

btnzero = Button(calc, text="0", width=6, height=2, bg='white',fg='black', font=('Arial', 18, 'bold'),command=lambda: getval('0')).grid(row=5, column=1)

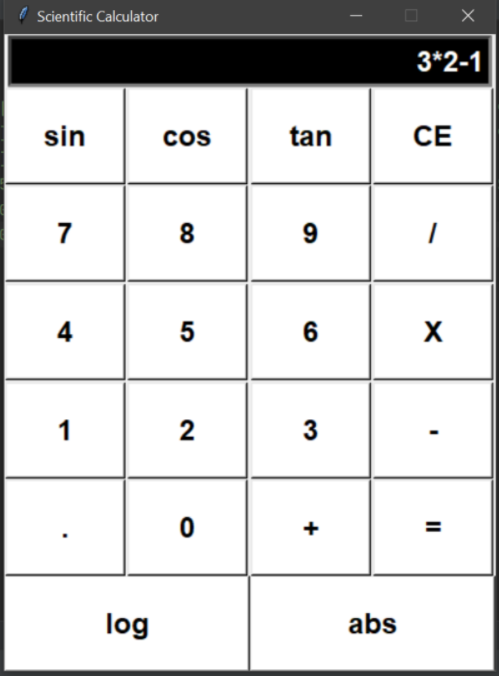
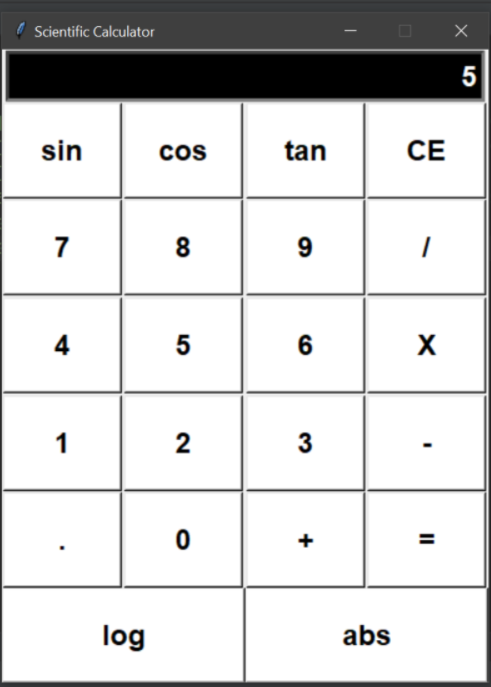
btndec = Button(calc, text=".", width=6, height=2, bg='white',fg='black', font=('Arial', 18, 'bold'),command=lambda: getval('.')).grid(row=5, column=0)

btnlog = Button(calc, text="log", width=13, height=2, bg='white',fg='black', font=('Arial', 18, 'bold'),command=lambda: getval('log')).grid(row=6, column=0,columnspan=2)

btnabs = Button(calc, text="abs", width=13, height=2, bg='white',fg='black', font=('Arial', 18, 'bold'),command=lambda: getval('abs')).grid(row=6, column=2,columnspan=2)

tk.mainloop()

**OUTPUT (snapshot) :**

**Conclusion:**

We successfully understood & implemented the concept of GUI & Tkinter in python ; thus used it to create a Scientific Calculator GUI.

**Post Lab questions:**

1.which widget is used as a container to house other widgets ?

**a) Frame ANS a**

b) Label

c) Message

d) Button

2. Which widget will be used to draw lines, circles, arcs, ovals and rectangles ?

a) Listbox

b) Frame

**c) Canvas ANS c**

d) Button

3) **mainloop()** is a method on the main window which is executed when we run application.

4) The **bind** method is used to bind an event with an event handler function.

5) The **grid**  layout manger arranges the widgets in a two-dimensional table containing rows and columns.

**Date: 18-04-2021 Signature of faculty in-charge**