Experiment No. 8
Creating GUI with python containing widgets such as labels, textbox, radio, checkboxes and custom dialog boxes
Date of Performance:
Date of Submission:

Experiment No. 8

Title: Creating GUI with python containing widgets such as labels, textbox, radio, checkboxes and custom dialog boxes

Aim: To study and create GUI with python containing widgets such as labels, textbox, radio, checkboxes and custom dialog boxes

Objective: To introduce GUI, TKinter in python

Theory:

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is the most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter is the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

To create a tkinter app:

Importing the module – tkinter

Create the main window (container)

Add any number of widgets to the main window

Apply the event Trigger on the widgets.

Importing tkinter is same as importing any other module in the Python code. Note that the name of the module in Python 2.x is 'Tkinter' and in Python 3.x it is 'tkinter'.

Code:

//tkinter page

```
import tkinter

top = tkinter.Tk()

top.mainloop()
```

//tkinter message box

```
import tkinter

from tkinter import messagebox

top = tkinter.Tk()

messagebox.showinfo("Hello Python","Hello World")

top.mainloop()
```

//tkinter canvas arc

```
import tkinter

from tkinter import messagebox

top = tkinter.Tk()

c=tkinter.Canvas(top,bg="lightblue",height=250,width=300)

coord=10,50,240,210

arc=c.create_arc(coord,start=0,extent=150,fill="red")
```

```
c.pack()

top.mainloop()
```

//tkinter canvas line

```
import tkinter

from tkinter import messagebox

top = tkinter.Tk()

c=tkinter.Canvas(top)

line=c.create_line(20,50,100,200,fill="red")

c.pack()

top.mainloop()
```

//**tkinter canvas oval**

```
import tkinter

from tkinter import messagebox

top = tkinter.Tk()

c=tkinter.Canvas(top)

oval=c.create_oval(20,50,100,200,fill="blue")

c.pack()

top.mainloop()
```

//**tkinter checkbutton**

```
from Tkinter import *
    import tkMessageBox
```

```
import Tkinter

top = Tkinter.Tk()
CheckVar1 = IntVar()
CheckVar2 = IntVar()
C1 = Checkbutton(top, text = "Music", variable = CheckVar1, \
    onvalue = 1, offvalue = 0, height=5, width = 20)
C2 = Checkbutton(top, text = "Video", variable = CheckVar2, \
    onvalue = 1, offvalue = 0, height=5, width = 20)
C1.pack()
C2.pack()
top.mainloop()
```

//**tkinter Entry**

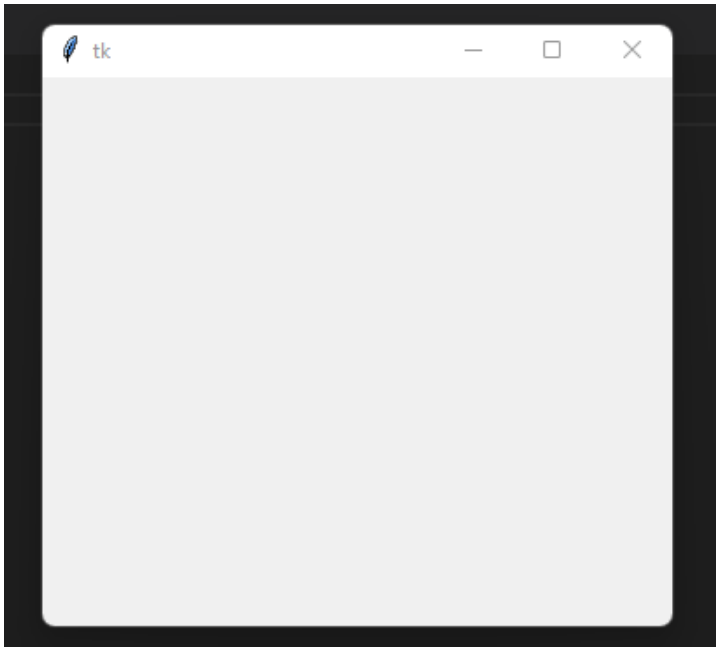
```
from Tkinter import *
top = Tk()
L1 = Label(top, text="User Name")
L1.pack( side = LEFT)
E1 = Entry(top, bd =5)
E1.pack(side = RIGHT)
top.mainloop()
```

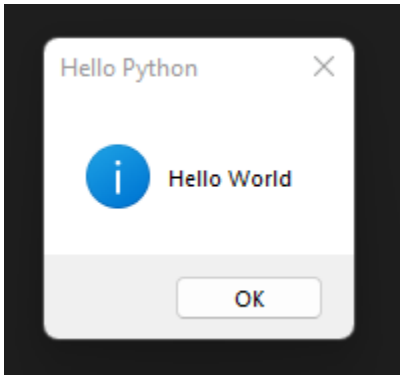
//**tkinter label**

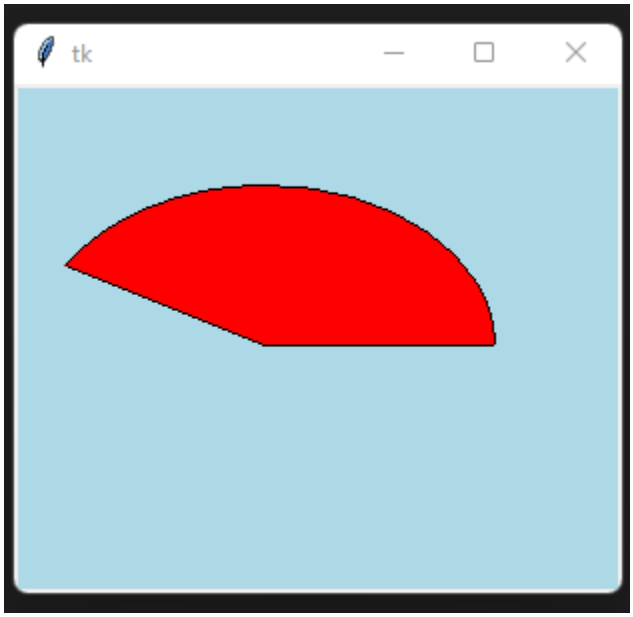
```
from Tkinter import *
root = Tk()
var = StringVar()
label = Label( root, textvariable=var, relief=RAISED )
var.set("Hey!? How are you doing?")
label.pack()
root.mainloop()
```

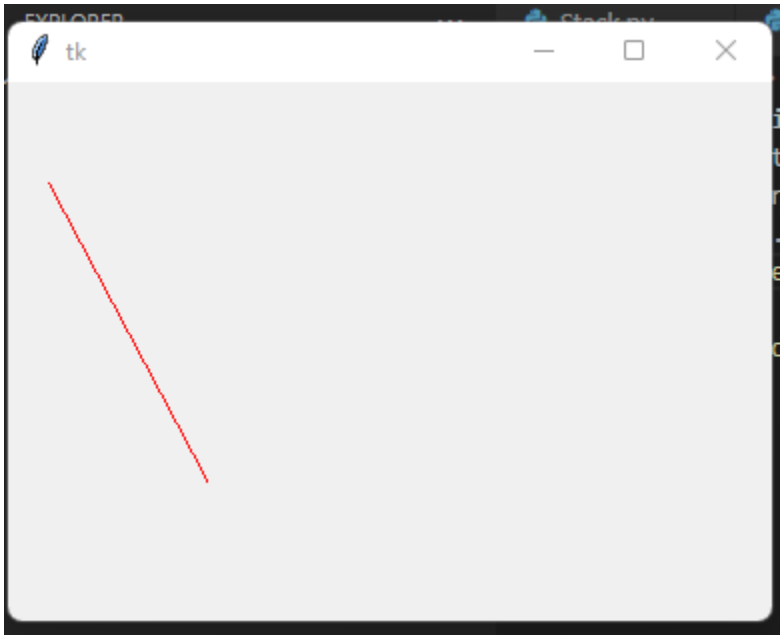
```
//tkinter Radiobutton
from Tkinter import *
def sel():
   selection = "You selected the option " + str(var.get())
   label.config(text = selection)
root = Tk()
var = IntVar()
R1 = Radiobutton(root, text="Option 1", variable=var,                    value=1,
command=sel)
R1.pack( anchor = W )
R2 = Radiobutton(root, text="Option 2", variable=var,
value=2,command=sel)
R2.pack( anchor = W )
R3 = Radiobutton(root, text="Option 3", variable=var,
value=3,command=sel)
R3.pack( anchor = W)
label = Label(root)
label.pack()
root.mainloop()
```

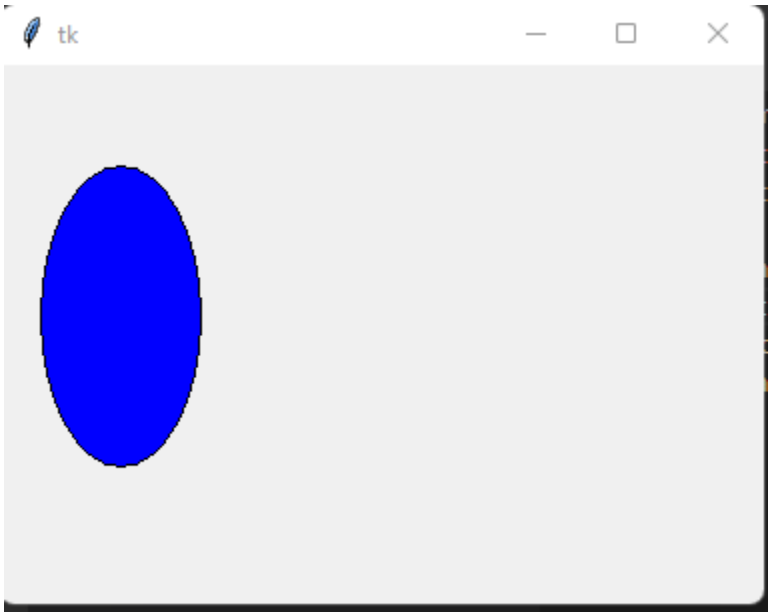
Output:




Hello Python
Hello World
OK


tk


tk


tk

Conclusion: Each code snippet demonstrates the creation and usage of different GUI components:

- Labels: Used for displaying text messages or information.
- Textbox (Entry): Allows users to input text.
- Radio buttons: Provides multiple options where only one option can be selected at a time.

- Checkboxes: Allows users to select multiple options simultaneously.
- Canvas: Used for drawing shapes such as arcs, lines, and ovals.
- Message box: Used to display custom messages or alerts.

The provided code snippets serve as a concise introduction to creating GUI applications in Python using Tkinter. They illustrate the basic syntax and usage of various Tkinter widgets and functionalities, making it a suitable starting point for beginners interested in GUI development with Python.