THAKUR COLLEGE OF SCIENCE & COMMERCE

NAAC Accredited with Grade "A" (3rd Cycle)

THAKUR TRUSTS

ISO 9001 : 2015 Certified

Degree College

Computer Journal

CERTIFICATE

SEMESTER II **UID No.** 2020858

Class FYBSC CS **Roll No.** 1146 **Year** 2020-2021

This is to certify that the work entered in this journal is the work of Mst. / Ms. SHAIKH KAYSAN RAZAUDDIN

who has worked for the year 2020-2021 in the Computer Laboratory.

Teacher In-Charge

Head of Department

Date :

Examiner

INDEX

PRACTICAL -1

GENERATING ITERABLES

#how to convert str into iterable

```
>>> str="I am kaysan"
>>> iter_str=iter(str)
>>> print(str)
I am kaysan
>>> print(iter_str)
<str_iterator object at 0x000001FFC9520C40>
>>> print(next(iter_str))
I
>>> print(next(iter_str))
a
>>> print(next(iter_str))
m
>>> #generating iterable from dictionary
>>> student_details={'Kaysan':1111,'Ajay':2222}
```

 # Convert dictionary(iterable) into iterator object using iter()method.

```
>>>iter_object=iter(student_details)>>> print(iter_object)
<dict_keyiterator object at 0x000002D69EF93060>
>>> print(next(iter_object))
Kaysan
>>> print(next(iter_object))
Ajay
```

```
>>> value_ele=iter(student_details.values())
>>> item_ele=iter(student_details.items())
>>> print(next(value_ele))
1111
>>> print(next(value_ele))
2222
```

#generating iterable from list

```
>>> list_1=["orange","apple","mango","banana","strawberry"]
>>> iter_list=iter(list_1)
>>> print(iter_list)
<list_iterator object at 0x000001FFC3520C40>
>>> print(next(iter_list))
orange
>>> print(next(iter_list))
apple
>>> print(next(iter_list))
mango
>>> print(next(iter_list))
banana
>>> print(next(iter_list))
strawberry
```

#generating iterables from tuple

```
>>> tuple_1=("red","green","blue","yellow","pink")
>>> iter_tuple=iter(tuple_1)
```

```
>>> print(iter_tuple)
<tuple_iterator object at 0x000001FFC9520C40>
>>> print(next(iter_tuple))
red
>>> print(next(iter_tuple))
green
>>> print(next(iter_tuple))
blue
>>> print(next(iter_tuple))
yellow
>>> print(next(iter_tuple))
pink
```

#generating iterables from set

```
>>> set_1={"dog","cat","rabbit","eagle","snake"}
>>> iter_set=iter(set_1)
>>> print(iter_set)
<set_iterator object at 0x0000021F941B1E80>
>>> print(next(iter_set))
dog
>>> print(next(iter_set))
cat
>>> print(next(iter_set))
rabbit
>>> print(next(iter_set))
eagle
```

```
>>> print(next(iter_set))
snake
```

 # generating iterables from range

```
>>> x=range(8)
>>> for i in x:
print(i)


0
1
2
3
4
5
6
7
>>> iter_range=iter(x)
>>> print(iter_range)
<range_iterator object at 0x0000021F98271C70>
>>> print(next(iter_range))
0
>>> print(next(iter_range))
1
>>> print(next(iter_range))
2
>>> print(next(iter_range))
```

```
3
>>> print(next(iter_range))
4
>>> print(next(iter_range))
5
>>> print(next(iter_range))
6
>>> print(next(iter_range))
7
```

PRACTICAL -2

FILE METHODS

```
>>> fileobj=open("kaysan.txt","w")
>>> fileobj.write(We are learning GUI concepts in python programming")


SyntaxError: invalid syntax
>>>
>>> fileobj=open("kaysan.txt","w")
>>> fileobj.write("We are learning GUI concepts in python programming")
50
>>> fileobj.close()
>>>
>>> fileobj=open("kaysan.txt","r")
>>> readstr=fileobj.read(20)
>>> print("The contents of the file are \n",readstr)
The contents of the file are
We are learning GUI
>>> fileobj.close()
>>>
>>> fileobj=open("kaysan.txt","a")
```

```
>>> fileobj.write("Next we shall be learning use of widgets")
40
>>> fileobj.write("\nThis is append at last")
23
>>> fileobj.close()
>>>
>>> fileobj=open("kaysan.txt","r")
>>> #readstr=fileobj.read(1000)
>>> #readstr=fileobj.readline()
>>> readstr=fileobj.readlines()
>>> print("The contents of the file are \n",readstr)
The contents of the file are
['We are learning GUI concepts in python programmingNext we shall be learning use of widgets\n', 'This is append at last']
>>> fileobj.close()
>>>
>>> f=open("kaysan.txt","w")
>>> f.write("Hello How are you")
17
>>> f.close()
>>>
>>> #opening "kaysan.txt" text file
>>> f=open("kaysan.txt","r")
>>> # Second parameter is by default 0
>>> # sets Reference point to 24th
>>> #index position from the beginning
```

```
>>> f.seek(24,0)
24
>>> #print current position
>>> print(f.tell())
24
>>> print(f.readline())

>>> #print(f.readlines())
>>> #print(f.read(30))
>>> f.close()
>>>
>>> #Opening "kaysan.txt" text file
>>> # in binary mode
>>> f=open("kaysan.txt", "rb")
>>> f.seek(-11,2)
6
>>> print("Current position")
Current position
>>> print(f.tell())
6
>>> print(f.readline().decode("utf-8"))
How are you
>>> print("Reading Through variable")
Reading Through variable
>>> f.seek(0,0)
```

```
0
>>> line=f.readlines()
>>> print(line)
[b'Hello How are you']
>>>
>>> #File attributes
>>> print(f.name)
kaysan.txt
>>> print(f.closed)
False
>>> print(f.mode)
rb
>>> print(f.closed)
False
>>> f.close()
>>> print(f.closed)
True
```

PRACTICAL -3

ITERATORS AND ITERABALES

```
>>> it_rable=["Python","Linux","OS",".NET",]
>>> it_rator=iter(it_table)
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    it_rator=iter(it_table)
NameError: name 'it_table' is not defined
>>> it_rator=iter(it_rable)
>>> print (it_rator)
<list_iterator object at 0x0000023F3A5D5550>
>>>
```

#Iterating through iterable using for loop

```
>>> for i in it_rable:
        print(i)
```

```
Python
Linux
OS
.NET
```

```
>>> print("\n\n")
```

#Iterating through iterator using for loop.

```
>>> for i in it_rator:
        print (i,end=" ")

Python Linux OS .NET
```

#calling iter() on iterator itself

```
>>> #i.e.we can get iterator from iterable
>>> #and iterator object from itself as well
>>> it_rator1=iter(it_rator)
>>> if it rator==it_rator1:

SyntaxError: invalid syntax
>>> it_rator1=iter(it_rator)
>>> if it_rator==it_rator1:
        print("\nBoth are iterator objects")
        else:

SyntaxError: invalid syntax
>>> it_rator1=iter(it_rator)
>>> if it_rator==it_rator1:
        print("\nBoth are iterator objects")
```

Both are iterator objects

PRACTICAL -4

ERRORS AND EXCEPTIONS

```
#errors and exceptions
print("\n\n\n\n ERRORS AND EXCEPTIONS \n")

try:
    file=open("new.txt","r")
#using exception with except method
except Exception:
    print("cant read the file")
else:
    print("file is ready to be read")

#using as keyword
print("\n\n\n using as keyword\n")
try:
    file=open("name.txt","r")
except Exception as e:
    print("cant read the file")
    print(e)
else:
    print("file is ready to read")
```

```
try:
   file=open("new.txt","r")
except Exception as e:
   print("cant read the file")
   print(e)
else:
   print("file is ready to read")

print("\n ++++++++++++++++++++++++++++++\n")
```

#with exception name

```
try:
   file=open("new.txt","r")

except IOError:
   print("file not found")
else:
   print("file opened sucessfully")
   file.close()

print("\n+++++++++++++++\n")

try:
   file=open("name.txt","r")
```

```
except IOError:
    print("file not found")
else:
    print("file opened sucessfully")
    file.close()

print("\n+++++++++++++++\n")
```

#multiple exceptions

```
print("\n\n\n\n multiple exceptions \n")
try:
    a=100/0
    file=open("name.txt","r")
except ArithmeticError:
    print("Arithmetic Error")
except IOError:
    print("IO Error")
else:
    print("print file successdully")
    file.close()
#use of multiple exception
try:
    a=100/10
    file=open("name.txt","r")
```

```
except ArithmeticError:
    print("Arithmetic Error")
except IOError:
    print("IO Error")
else:
    print(" file opened successdully")
    file.close()

print("\n+++++++++++++++++++++++++++++++\n")
```

#using final block

```
try:
    file=open("next.txt","r")
    try:
        file.write("writing in read mode")
    finally:
        file.close()
        print("now thefile is closed ")
except:
    print("error writing in the read mode in file")

print("\n+++++++++++++++++++++++++++++++\n")
```

Output:

ERRORS AND EXCEPTIONS

can’t read the file

using as keyword

file is ready to read

can’t read the file

[Errno 2] No such file or directory: 'new.txt'

++++++++++++++++++++++++++++++

file not found

+++++++++++++++

file opened sucessfully

+++++++++++++++

multiple exceptions

Arithmetic Error

file opened successdully

++++++++++++++++++++++++++++++

error writing in the read mode in file

++++++++++++++++++++++++++++++

++

ASSIGNMENT-1

Using Assert, Finding roll numbers and Names

```
f=open("roll.txt","w")
f.write=("111,222,333,444,555,666,777,888,999,1100")
f.close()
file=open("name.txt","r")


class roll(Exception):
    pass
def check_roll(roll):
    if int(roll) <= 1100:
        raise roll
    else:
        print('roll: '+str(roll))


def assert_roll(roll):
    try:
        assert int(roll) > 1100
    except ValueError:
        return"value cant convert into int"
    else:
        return"roll within range"
```

```
print("\n\n\n++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
+++++=\n\n\n")
```

Output:

```
=================== RESTART: C:\Users\Admin\Desktop\12333.py ===================


++++++++++++++++++++++++++++++++++++++++++++++++++++++++++=
```

PRACTICAL -5

REGULAR EXPRESSIONS

#Use of regular expression

```
print("\n\n\n\n use of ^ character \n")
import re
string="hola! this is the word hello we say in espanol or spanish"
#use of ^ character
foundstart=re.findall("^holu", string)
if foundstart:
    print("YES! the given string starts with the word hola")
else:
    print("NO the string dont start with the word hola")
```

#Use of $ character

```
print("\n\n\n\n use of dollar sign \n")
string="hola! this is the word hello we say in espanol or spanish"
foundend=re.findall("spanish$", string)
if foundend:
    print("yes the given string end  with the word spanish")
else:
    print("no the given string does not end with the word spanish")
```

#Use of dot character

```
print("\n\n\n\n use of dot character \n")
string="hola! this is the word hello we say in espanol or spanish"
x=re.findall("st...s", string)
print(x)
x=re.findall("e.....l", string)
print(x)
```

#Use of star character

```
print("\n\n\n\n use of star character \n")
string="hola! this is the word hello we say in espanol or spanish"
usestar=re.findall("inx*", string)
print(usestar)
usestar=re.findall("in*", string)
print(usestar)
```

#Use of plus character

```
print("\n\n\n\n use of plus character \n")
string="hola! this is the word hello we say in espanol or spanish"
useplus=re.findall("in+", string)
print(useplus)
useplus=re.findall("o+", string)
```

```
print(useplus)
```

#Use of Compile Function

```
import re
print("\n\n\n\n use of compile function \n")
#complie creates regular expression character class(a-e)
#which is eqivalent to [abcde]
pattern=re.compile('[a-e]')
print(pattern.findall("learning regular expressions"))
pattern=re.compile('[a-g]')
print(pattern.findall("learning regular expressions"))
pattern=re.compile('[A-Z]')
print(pattern.findall("Learning Regular Expressions"))
```

```
import re
print("\n\n\n\n use of capital d \n")
#\d is equivalent to [0-9].
usebackslashd=re.compile('\d')
print(usebackslashd.findall("Today is the day i go to the campsite with my family the date is 14th February 2021”))
print("\n\n\n\n use of capital d+ \n")
```

#Use of \d+

```
usebackslashd=re.compile('\d+')

print(usebackslashd.findall("Today is the day i  go to the campsite with my family the date is 14th February 2021”))

print("\n\n\n\n use of capital w \n")
```

#Use of \w

```
usebackslashw=re.compile('\w')

print(usebackslashw.findall("Today is the day i go to the campsite with my family the date is 14th February 2021”))

print("\n\n\n\n use of capital(W) \n")
```

#Use of \w+

```
usebackslashw=re.compile('\W')

print(usebackslashw.findall("Today is the day i go to the campsite with my family the date is 14th February 2021"))

print("\n\n\n\n use of split funtion\n")
```

```
import re

#useof split function
```

```
gstring="Today is the day i go to the campsite with my family the date is 14th February 2021"

x=re.split("\s", gstring)

print(x)



print("\n\n\n\n use of sub method\n")
```

#Use of sub method

```
gstring="Today is the day i go to the campsite with my family the date is 14th February 2021"

x=re.sub("\s",",", gstring)

print(x)
```

#Use of group method

```
print("\n\n\n\n use of group \n")

string=re.match(r"(\w+) (\w+) (\w+)","Computer Science Python")

print(string.group(0))

print(string.group(1))

print(string.group(2))

print(string.group(3))


print(string.group(1,2,3))
```

```
def substitutor():
    #a string variable
    string_var="it's all about practice and learning"
    print(re.sub(r"practice", "experience", string_var))


print("\n\n\n\n\n")
```

#Using mobile number

```
mobile_no_list=["9867260530","854796355","12344455566","7208521460"]
for val in mobile_no_list:
  if re.match("[8-9] [1] [0-9] {9}",val):
    print ("The mobile number",val,"is Correct")
  else:
     print("The mobile number",val," is not Correct")
```

#Using Mail Id

```
import re
#any character a-z, sny digit 0-9 and symbol '_' followed by a '@' symbol
#insert your text here
text= """This is an interesting scenario wherein you come across mail ids
engrossed in the textual information
        mail ids engrossed in the textual imformation
        and you have to segregate these ids. Hello i Am Kaysan and my email id is
sha.i.kh-kaysan@gmail.com this is
        to extract mail id from text and this is learnt by kazim720@gmail.com
        and also by kaheesha@yahoo.in"""
```

```
text1=text.lower()
print(re.findall(r"[\w.-]+@[\w.-]+", text1))
```

OUTPUT:

Use of ^ character

NO the string dont start with the word hola

Use of dollar sign

yes the given string end with the word spanish

Use of dot character

[]

['espanol']

Use of star character

['in']

['i', 'i', 'in', 'i']

Use of plus character

['in']

['o', 'o', 'o', 'o', 'o']

Use of compile function

['e', 'a', 'e', 'a', 'e', 'e']

['e', 'a', 'g', 'e', 'g', 'a', 'e', 'e']

['L', 'R', 'E']

Use of capital d

['1', '5', '2', '0', '2', '1']

Use of capital d+

['15', '2021']

Use of capital w

['T','o','d','a','y','i','s','t','h','e','d','a','y','I','g','o','t','o','t','h','e','c','a','m','p','s','I','t','e','w','i','t','h','m','y','f','a','m','i','l','y','t','h','e','d','a','t','e','i','s','1','4','t','h','F','e','b','r','u','a','r','y','2','0','2','1']

Use of capital(W)

[' ', ' ']

Use of split function

['Today' 'is' 'the' 'day' 'i' 'go' 'to' 'the' 'campsite' 'with' 'my' 'family' 'the' 'date' 'is' '14th' 'February' '2021']

Use of sub method

Today, is, the, day, i, go, to, the, campsite, with, my, family, the, date, is, 14th, February, 2021

Use of group

Computer Science Python

Computer

Science

Python

('Computer', 'Science', 'Python')

Using mobile number

The mobile number 9867260530 is not Correct

The mobile number 854796355 is not Correct

The mobile number 12344455566 is not Correct

The mobile number 7208521460 is not Correct

Using Mail Id

['sha.i.kh-kaysan@gmail.com', 'kazim720@gmail.com', 'kaheesha@yahoo.in']

date="2021-02-23 06:08:18"

PRACTICAL -6

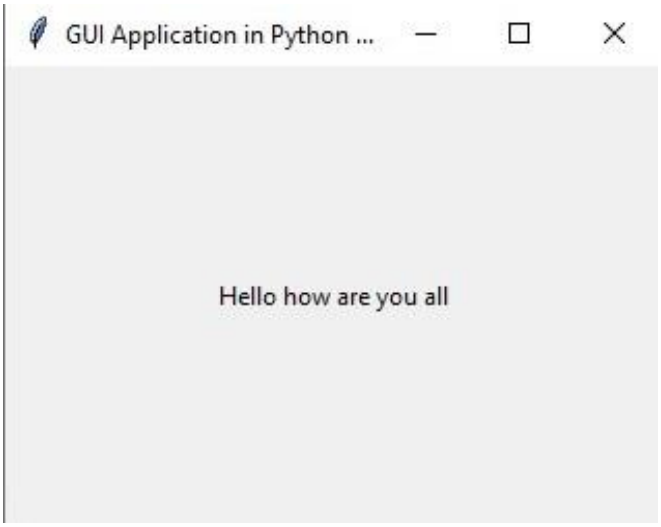
GUI (Using Tkinter Library)

#GUI in python (pack method and button widgets)

Input:

```
import tkinter
from tkinter import *

window = Tk()
window.title("GUI Application in Python Environment")

Kaysan = Label(window, text = "Hello how are you all")

#Kaysan.pack()

Kaysan.pack(side = LEFT, expand = True, padx = 100, pady = 100)
window.mainloop()
```

Output:



Input:

```
import tkinter
from tkinter import *


window=Tk()
def display():
   print("Hi this is Button Widget")
button1=Button(window,text="Click to see the
mesaage",command=display,bg='red',fg='green',font='arielblack',
            height=10,width=20,bd=10)
button1.pack()
window.mainloop()
```

Output:

Input:

```
import tkinter.messagebox


def callback():
    tkinter.messagebox.showinfo("Message box would appear as title"," Hi How are you all?")
callback()
def flatc():
    print("Hello!Enjoying learning GUI feature")
flatc()
```

Output:

Hello!Enjoying learning GUI feature

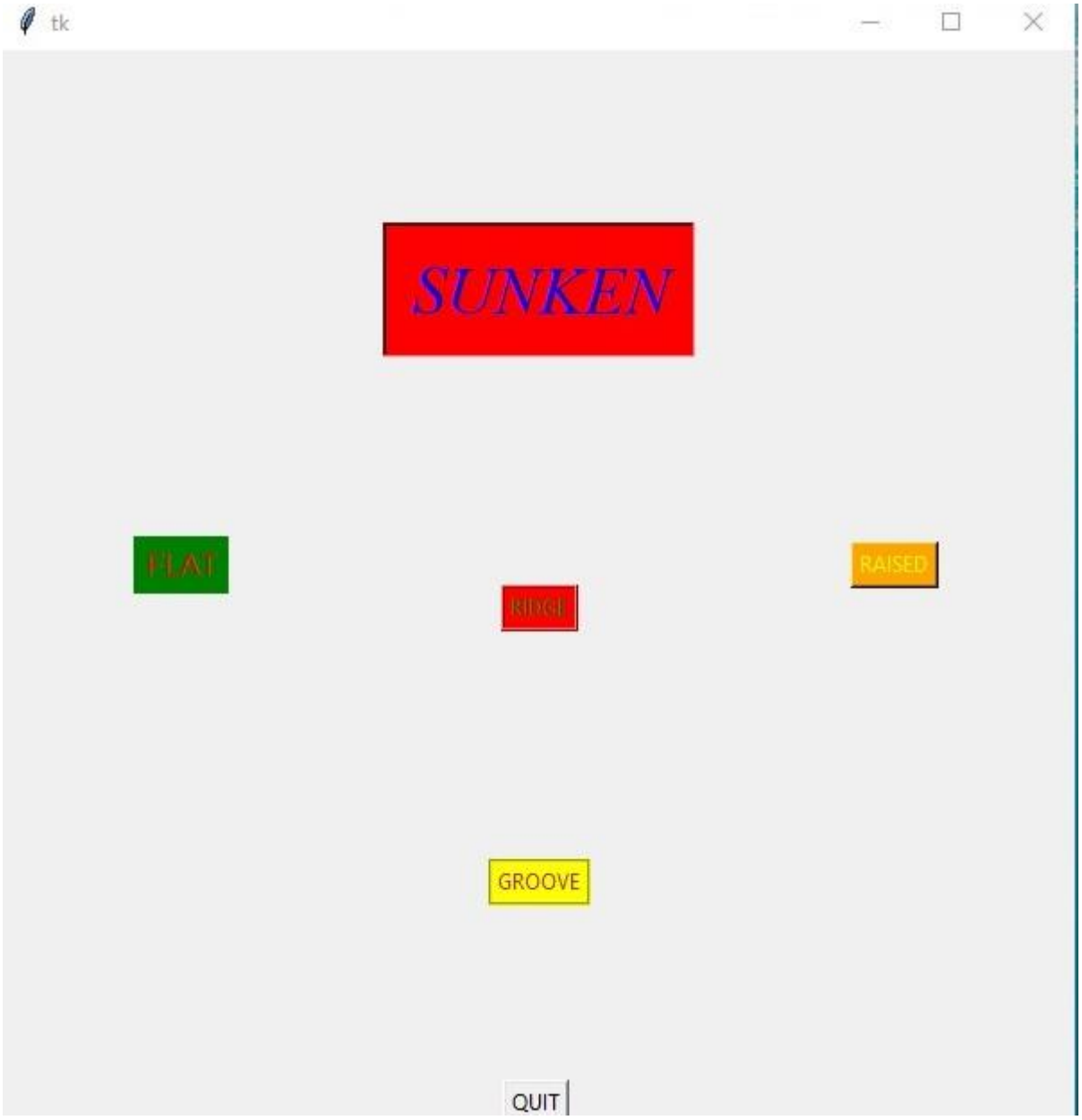
Input:

```
from tkinter import *
import tkinter

relief_attrib = tkinter.Tk()

relief_flat=tkinter.Button(relief_attrib, text = "FLAT",
                    relief = FLAT, bg = 'green', fg = 'red', font='arielblack',
                    command = 'flatc')

relief_raised =tkinter.Button(relief_attrib, text="RAISED", state = NORMAL,
                      relief = RAISED, bg = 'orange', fg = 'yellow',
                      cursor = 'pencil')
```

```
relief_sunken = tkinter.Button(relief_attrib, text= "SUNKEN", relief = SUNKEN,
                    command = 'callback', cursor = 'tcross',
                    font = ('times','28','italic'), bg = 'red', fg= 'blue')

relief_groove = tkinter.Button(relief_attrib, text = "GROOVE", relief = GROOVE,
                    cursor = 'hand2', bg = 'yellow', fg = 'brown')

relief_ridge = tkinter.Button(relief_attrib, text = "RIDGE", relief = RIDGE,
                    cursor = 'heart', bg = 'red', fg = 'green')

quitb = tkinter.Button(relief_attrib, text = "QUIT", command =
'relief_attrib.destory')



quitb.pack(side = BOTTOM)

relief_flat.pack(side = LEFT, expand = True, padx = 10, pady = 30)

relief_raised.pack(side = RIGHT, expand = True, padx = 10, pady = 30)

relief_sunken.pack(side = TOP, expand = True, padx = 10, pady = 30)

relief_groove.pack(side = BOTTOM, expand = True, padx = 10, pady = 30)

relief_ridge.pack(side = LEFT, expand = True, padx = 10, pady = 30)
```

relief_attrib.mainloop()

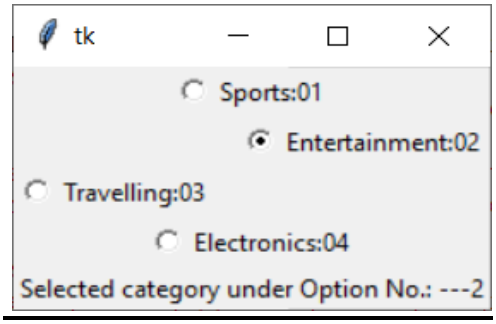
Output:



Input:

```
from tkinter import *
def opted():
   selection="Selected category under Option No.: ---" + str(var.get())
   label.config(text=selection)
mainwindow=Tk()
var=IntVar()

option1=Radiobutton(mainwindow,text="Sports:01",variable=var,value=1,comma
nd=opted)
option1.pack(anchor=N)

option2=Radiobutton(mainwindow,text="Entertainment:02",variable=var,value=2,
command=opted)
option2.pack(anchor=E)

option3=Radiobutton(mainwindow,text="Travelling:03",variable=var,value=3,co
mmand=opted)
option3.pack(anchor=W)

option4=Radiobutton(mainwindow,text="Electronics:04",variable=var,value=4,co
mmand=opted)
option4.pack(anchor=S)

label=Label(mainwindow)
label.pack()
mainwindow.mainloop()
```

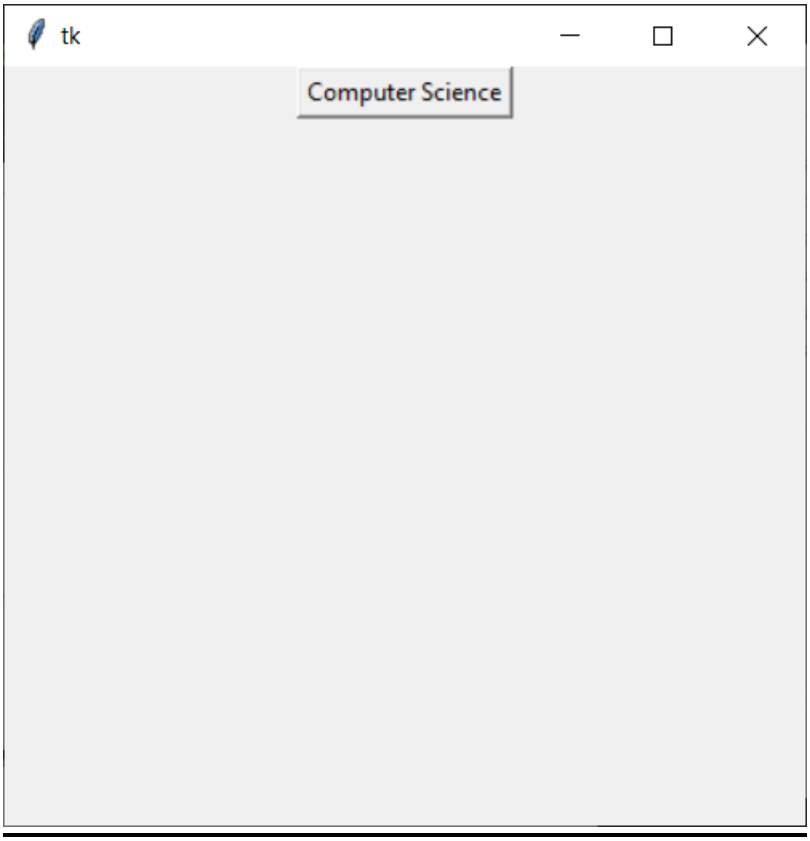
Output:



Input:

```
from tkinter import *

root_window = Tk()

root_window.geometry("400x400")

button_obj = Button(root_window, text = "Computer Science")

button_obj.pack()
```

Output:

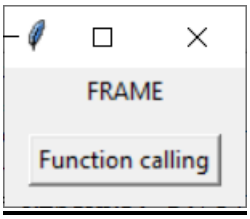


Input:

```
from tkinter import *
def frame_demo():
   print("Learning importance of frame creation")
parent=Tk()

frame_obj1=Frame(parent)
frame_obj2=Frame(parent)
parent.title("Frame creation in today's class")

label_obj=Label(frame_obj1,text="FRAME",justify=LEFT)
```

```
label_obj.pack(side=LEFT)

button_obj=Button(frame_obj2,text="Function calling",command=frame_demo)
button_obj.pack()

frame_obj1.pack(padx=1,pady=1)
frame_obj2.pack(padx=10,pady=10)

parent.mainloop()
```

Output:



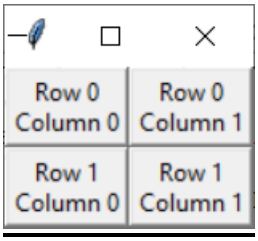
Learning importance of frame creation

Input:

```
from tkinter import *
root=Tk()
for i in range(2):
    for j in range(2):
        frame=Frame(root,relief=RAISED,borderwidth=2)
        frame.grid(row=i,column=j)
        label=Label(frame,text=f"Row {i}\nColumn {j}")
        label.pack()
```

```
root.mainloop()
```

Output:

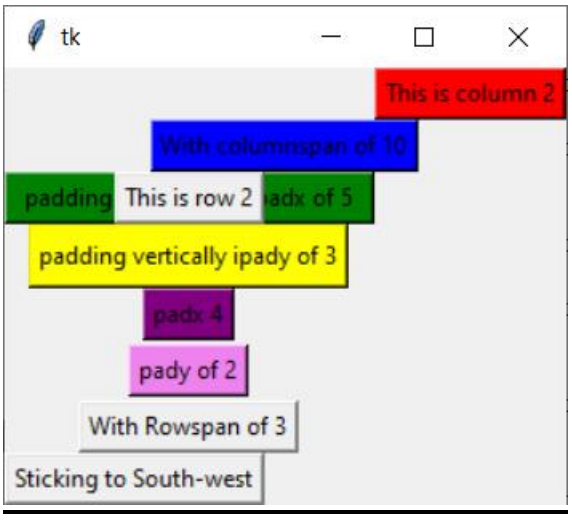


Input:

```
from tkinter import *
root=Tk()

btn_column=Button(root,text="This is column 2",bg="red")
btn_column.grid(column=2)

btn_columnspan=Button(root,text="With columnspan of 10",bg="blue")
btn_columnspan.grid(columnspan=10)

btn_ipadx=Button(root,text="padding horizontally ipadx of 5",bg="green")
btn_ipadx.grid(ipadx=5)

btn_ipady=Button(root,text="padding vertically ipady of 3",bg="yellow")
btn_ipady.grid(ipady=3)

btn_padx=Button(root,text="padx 4",bg="purple")
btn_padx.grid(padx=4)
```

```
btn_pady=Button(root,text="pady of 2",bg="violet")
btn_pady.grid(pady=2)

btn_row=Button(root,text="This is row 2")
btn_row.grid(row=2)

btn_rowspan=Button(root,text="With Rowspan of 3")
btn_rowspan.grid(rowspan=3)

btn_sticky=Button(root,text="Sticking to South-west")
btn_sticky.grid(sticky=SW)

root.mainloop
```

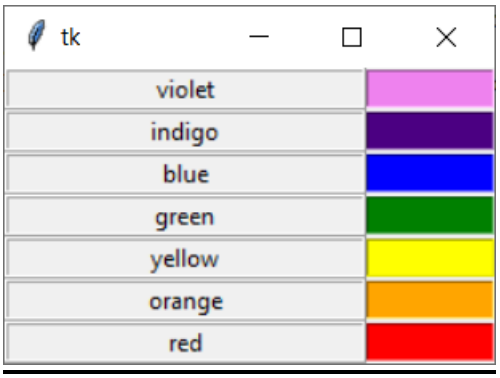
Output:



Input:

```
import tkinter as tk
color=["violet","indigo","blue","green","yellow","orange","red"]
rowno=0
for entities in color:
    tk.Label(text=entities,relief=tk.RIDGE,width=25).grid(row=rowno,column=0)
    tk.Entry(bg=entities,relief=tk.SUNKEN,width=10).grid(row=rowno,column=1)
    rowno=rowno+1
tk.mainloop()
```

Output:



Input:

```
import tkinter as tk
activities=['Reading','Learning','Playing','Drawing','Writing']
r=['ACTIVITY']
for c in activities:
    tk.Label(text=c,width=15).grid(column=0)
    tk.Label(text=r,relief=tk.RIDGE,width=15).grid(column=1)
tk.mainloop()
```

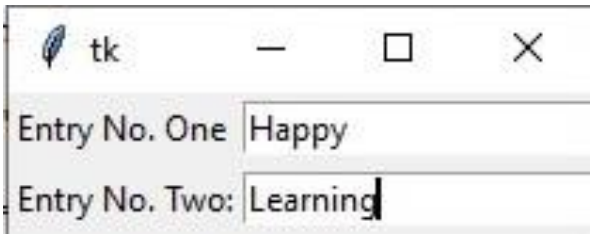
Output:

Input:

```
from tkinter import *
master=Tk()

#create a label widget
label_obj1=Label(master,text="Entry No. One")
label_obj2=Label(master,text="Entry No. Two:")

#Use of grid method
label_obj1.grid(row=0,column=0,sticky=W,pady=2)
label_obj2.grid(row=1,column=0,sticky=W,pady=2)

#Entry widgets, used to take entry from user
entry_obj1=Entry(master)
```

```
entry_obj2=Entry(master)


#This will arrange entry widgets
entry_obj1.grid(row=0,column=1,pady=2)
entry_obj2.grid(row=1,column=1,pady=2)


mainloop()
```

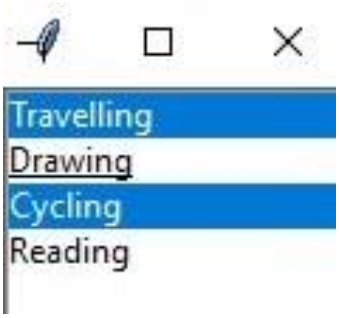
Output:



Input:

```
from tkinter import *
master=Tk()
listbox1=Listbox(master,selectmode=MULTIPLE)
listbox1.insert(1,"Travelling")
listbox1.insert(1,"Drawing")
listbox1.insert(2,"Cycling")
listbox1.insert(3,"Reading")
listbox1.pack()
master.mainloop()
```

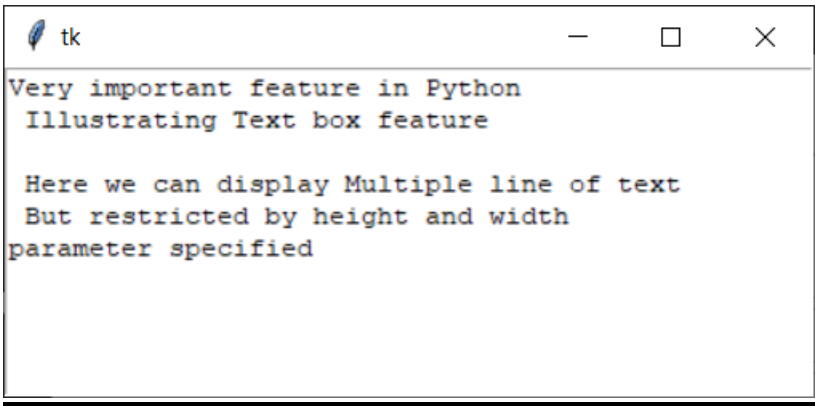
Output:



Input:

```
import tkinter as tk
root=tk.Tk()
T=tk.Text(root,height=10,width=50)
T.pack()
T.insert(tk.END,"""Very important feature in Python \n Illustrating Text box feature
\n Here we can display Multiple line of text \n But restricted by height and width
parameter specified""")
tk.mainloop()
```

Output:



PRACTICAL -7

HTML TAGS AND BROWSER

#HTML IN PYTHON

```
import webbrowser
file_open=open("py.html","w")

content="""<html>
<head><title>How To Start Your Own Blogging</title></head><br><br>
<body bgcolor="pink">
<center>
<h1>BLOGGING</h1>

<p>Types of Blogging</p>
<ol type="a">
<li>Food Blogs</li>
<li>Fitness Blogs</li>
<li>DIY Blogs</li>
<li>Lifestyle Blogs</li>
</ol>
</hr>
<hr>
```

```
<p> Steps To Follow </p>
<ol>
<li>Choose your blog name and get your blog hosting.</li>
<li>Start your blog by adding WordPress.</li>
<li>Pick a simple theme to make your blog your own.</li>
<li>Add two key blogging plugins to find your readers and track stats.</li>
<li>Write compelling content to create a blog that your readers love</li>
</ol>
</hr>
<hr>
<p>HOW TO MAKE IMPROVEMENTS</p>
<ol type="1">
<li>Write catchier headlines. ...</li>
<li>Use scannable formatting. ...</li>
<li>Ask experts to guest post.</li>
</ul>
</hr>

<hr>
<p> WANT TO MAKE YOUR OWN BLOG ...</p>
FILL THE REQUIRED DETAILS AND SUMBIT IT!!!<br><br>

<form>
<label for="fname">First name:</label><br>
```

```
<input type="text" id="fname" name="fname"><br>

<label for="lname">Last name:</label><br>
<input type="text" id="lname" name="lname"><br>
Enter your Address<br>
<textarea row="2" cols="20"></textarea><br><br>

<b>Which Blogging Page You Want To Create ?</b><br><br>
<input type="checkbox",name="Food Blog">Food Blog<br><br>
<input type="checkbox",name="Fitness Blog">Fitness Blog<br><br>
<input type="checkbox",name="DIY Blog">DIY Blog<br><br>
<input type="checkbox",name="Lifestyle Blog Blog">DIY Blog<br><br>
<input type="Submit"><br><br>
</form>
</hr>
</body>
</html>"""

file_open.write(content)
file_open.close()

webbrowser.open_new_tab('py.html')
```

Output:


BLOGGING
Types of Blogging
a. Food Blogs
b. Fitness Blogs
c. DIY Blogs
d. Lifestyle Blogs
Steps To Follow
1. Choose your blog name and get your blog hosting.
2. Start your blog by adding WordPress.
3. Pick a simple theme to make your blog your own.
4. Add two key blogging plugins to find your readers and track stats.
5. Write compelling content to create a blog that your readers love.
HOW TO MAKE IMPROVEMENTS
1. Write catchier headlines. ...
2. Use scannable formatting. ...
3. Ask experts to guest post.
WANT TO MAKE YOUR OWN BLOG ...
FILL THE REQUIRED DETAILS AND SUMBIT IT!!!
First name:
Kaysan
Last name:
Shaikh
Enter your Address
402,Queens Park
Which Blogging Page You Want To Create ?
Food Blog
Fitness Blog
DIY Blog
DIY Blog
Submit

ASSIGNMENT-2

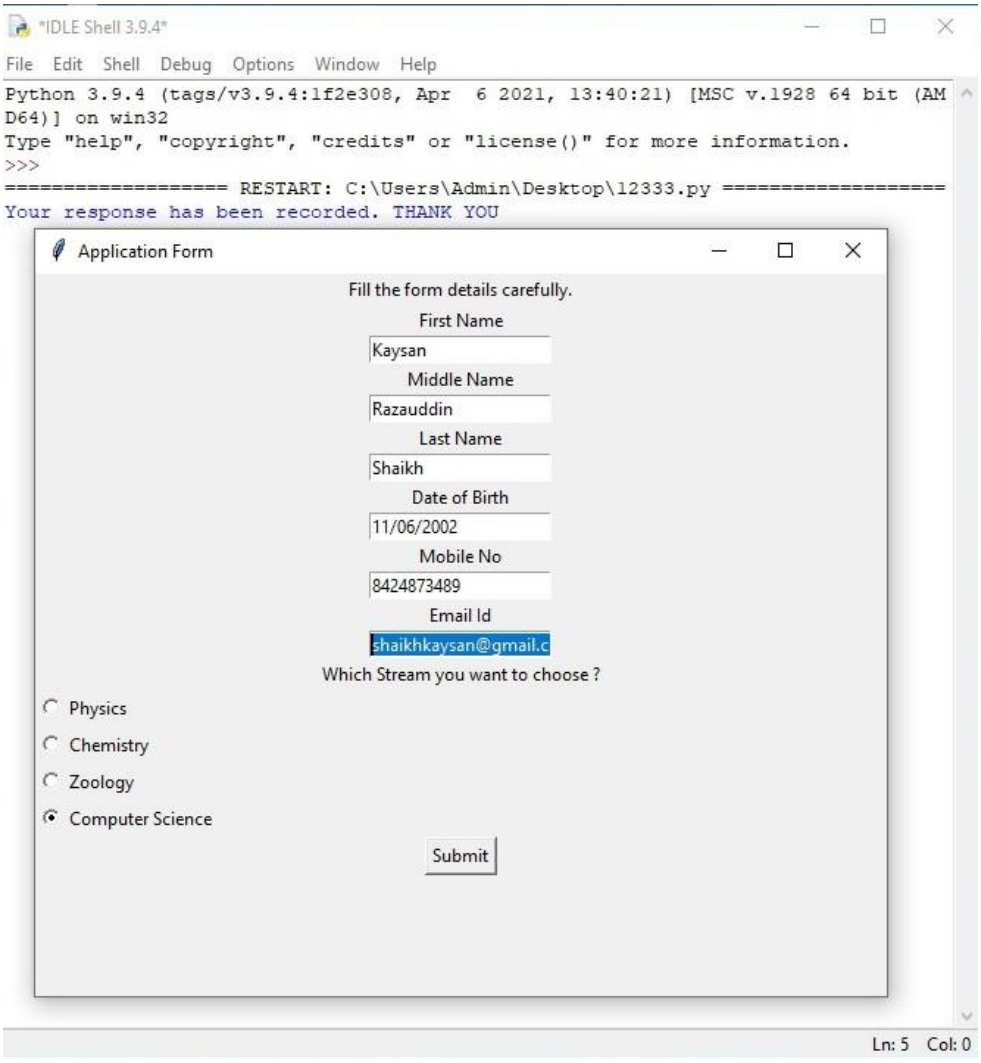
APPLICATION FORM (GUI) IN PYTHON

USING TKINTER LIBRARY

```
from tkinter import *

form = Tk()
form.geometry("600x600")

form.title("Application Form")
label=Label(form,text="Fill the form details carefully.")
label.pack()

Firstname=Label(form,text="First Name",).pack()
n1=Entry(form).pack()
Middlename=Label(form,text="Middle Name").pack()
n2=Entry(form).pack()
Lastname=Label(form,text="Last Name").pack()
n3=Entry(form).pack()
DOB=Label(form,text="Date of Birth").pack()
n4=Entry(form).pack()
MobileNo=Label(form,text="Mobile No").pack()
n5=Entry(form).pack()
```

```
EmailId=Label(form,text="Email Id").pack()
n6=Entry(form).pack()

var=IntVar()
stream=Label(form,text="Which Stream you want to choose ?").pack()
opt1=Radiobutton(form,text="Physics",variable=var)
opt1.pack(anchor=W)
var=IntVar()
opt2=Radiobutton(form,text="Chemistry",variable=var)
opt2.pack(anchor=W)
var=IntVar()
opt3=Radiobutton(form,text="Zoology",variable=var)
opt3.pack(anchor=W)
var=IntVar()
opt4=Radiobutton(form,text="Computer Science",variable=var)
opt4.pack(anchor=W)

def click():
  print("Your response has been recorded. THANK YOU")

button=Button(form,text="Submit",command=click)
button.pack()

form.mainloop()
```

Output:


IDLE Shell 3.9.4
File Edit Shell Debug Options Window Help
Python 3.9.4 (tags/v3.9.4:1f2e308, Apr 6 2021, 13:40:21) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\Admin\Desktop\12333.py ==========
Your response has been recorded. THANK YOU
Application Form
Fill the form details carefully.
First Name
Kaysan
Middle Name
Razauddin
Last Name
Shaikh
Date of Birth
11/06/2002
Mobile No
8424873489
Email Id
shaikhkaysan@gmail.c
Which Stream you want to choose ?
Physics
Chemistry
Zoology
Computer Science
Submit
Ln: 5 Col: 0

PRACTICAL -8

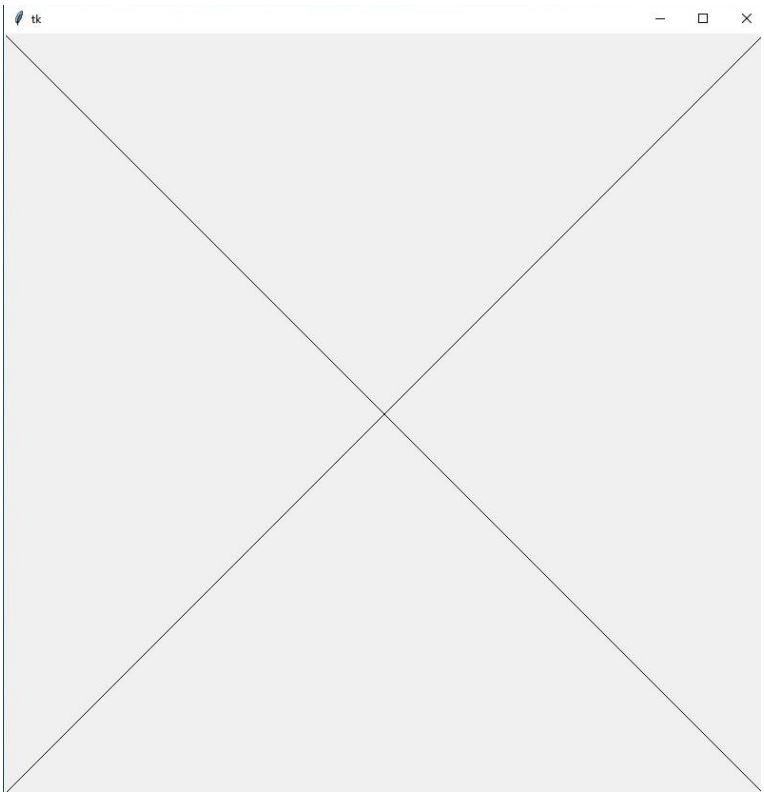
CANVAS

#canvas in python

#creating line in canvas

```
from tkinter import *
class CSline():
    def __init__(self):
        self.main_window=Tk()
        self.canvas=Canvas(self.main_window,width=800,height=800)
        self.canvas.create_line(0,0,799,799)
        self.canvas.create_line(799,0,0,799)
        self.canvas.pack()
        mainloop()
abline=CSline()
```

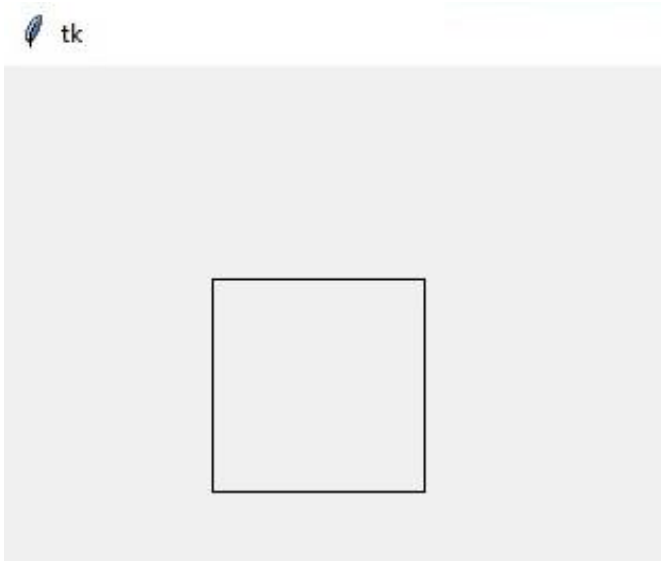
Output:



#creating rectangle

```
from tkinter import *
class rectangle():
    def __init__(self):
        self.main_window=Tk()
        self.canvas=Canvas(self.main_window,height=900,width=800)
        self.canvas.create_rectangle(100,100,200,200)
        self.canvas.pack()
        mainloop()
abrectangle=rectangle()
```

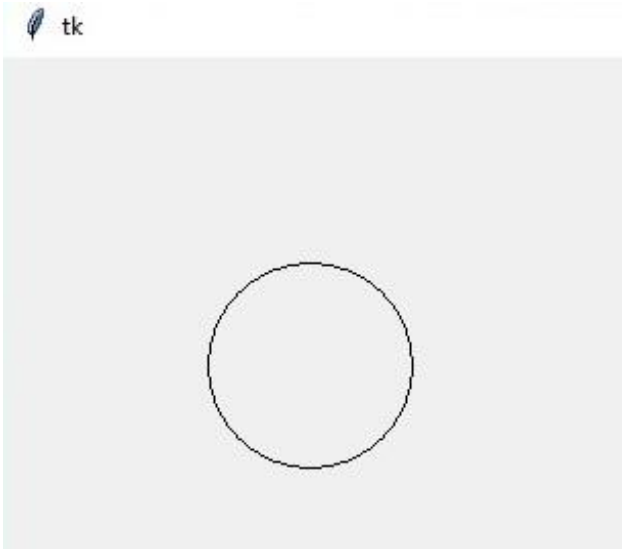
Output:



#creating oval

```
from tkinter import *
class oval():
    def __init__(self):
        self.main_window=Tk()
        self.canvas=Canvas(self.main_window,height=900,width=800)
        self.canvas.create_oval(100,200,100,200)
        self.canvas.pack()
        mainloop()
aboval=oval()
```

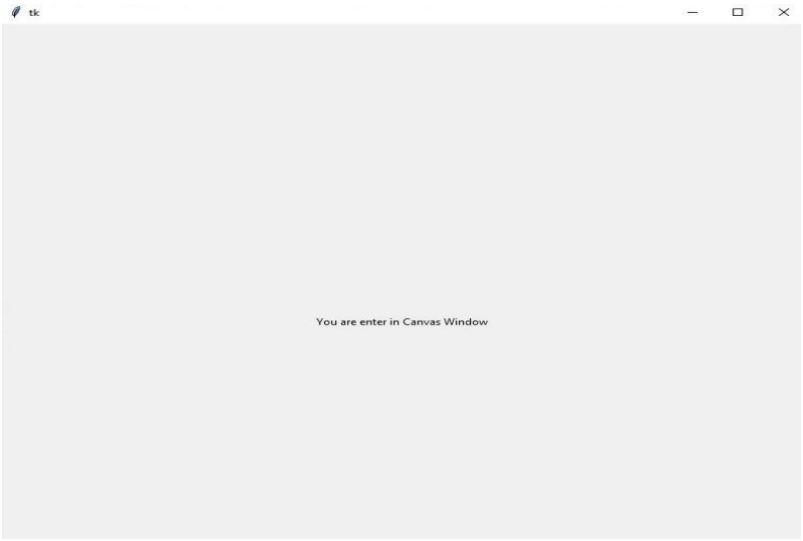
Output:



#creating text

```
from tkinter import *
class text():
    def __init__(self):
        self.main_window=Tk()
        self.canvas=Canvas(self.main_window,height=900,width=800)
        self.canvas.create_text(450,400,text="You are enter in Canvas Window")
        self.canvas.pack()
        mainloop()
abtext=text()
```

Output:

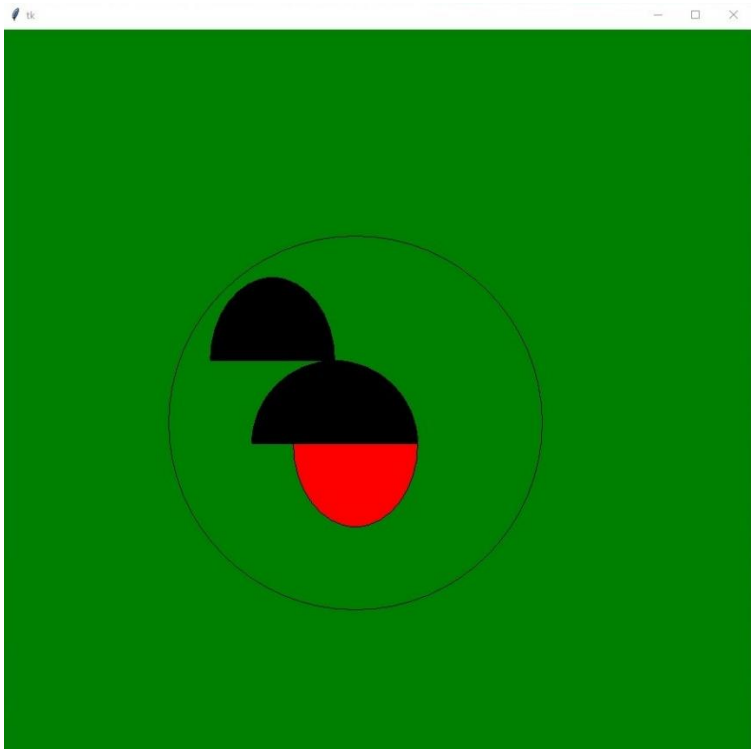


#creating canvas without using the init method

```
from tkinter import *
main=Tk()

can_obj = Canvas(main,bg="green",height=900,width=900)

#line = can_obj.create_line(0,899,899,0,fill='yellow')

arc = can_obj.create_arc(250,300,400,500,start=0,extent=180,fill="black")

arc = can_obj.create_arc(300,400,500,600,start=0,extent=180,fill="black")
```

```
arc = can_obj.create_arc(350,400,500,600,start=0,extent=-180,fill="red")


oval = can_obj.create_oval(200,250,650,700)


can_obj.pack()
mainloop()
```

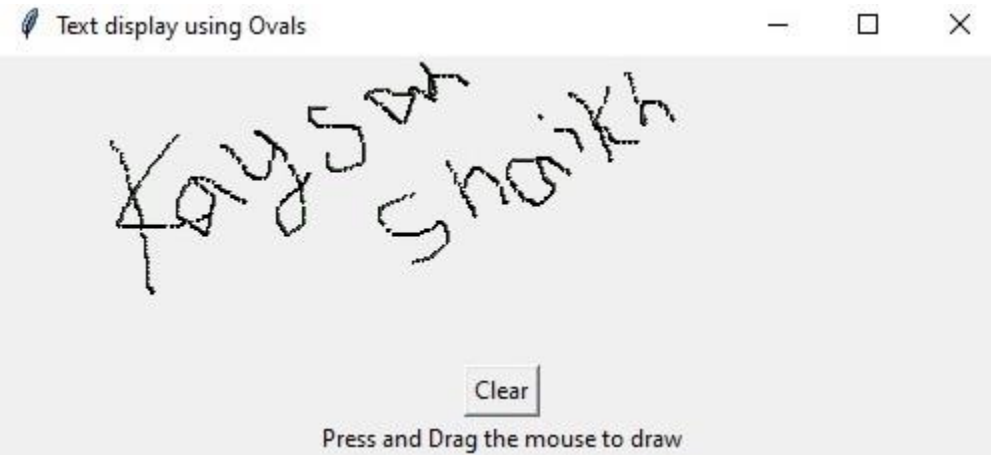
Output:



```
from tkinter import *
canvas_width=500
canvas_height=150
```

```
def paint(event):
    python_green="#476042"

    x1,y1=(event.x-1),(event.y-1)
    x2,y2=(event.x-1),(event.y-1)
    can_obj.create_oval(x1,y1,x2,y2,fill=python_green)

def clearall():
    can_obj.delete("all")

master=Tk()
master.title("Text display using Ovals")
can_obj=Canvas(master,width=canvas_width,height=canvas_height)
can_obj.pack(expand=YES,fill=BOTH)
can_obj.bind("<B1-Motion>",paint)

message=Label(master,text="Press and Drag the mouse to draw")
message.pack(side=BOTTOM)

button=Button(master,text="Clear",command=clearall)
button.pack(side=BOTTOM)

mainloop()
```

Output:

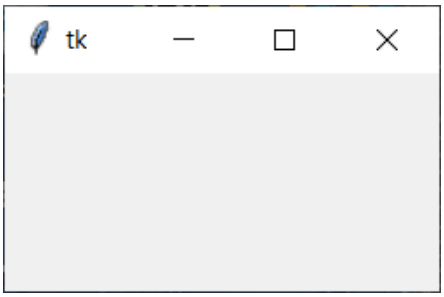


#Example of bind method

```
from tkinter import *
root=Tk()
root.geometry('200x100')

#function to be called when mouse enters in a frame
def enter(event):
    print("Coordinates [Entering frame] at x=%d,y=%d"%(event.x,event.y))

#function to be called when mouse exits the frame
def exit(event):
    print("Coordinates [Exiting frame] at x=%d,y=%d"%(event.x,event.y))

frameobject=Frame(root,height=100,width=200)
```

```
frameobject.bind('<Enter>',enter)
frameobject.bind('<Leave>',exit)

frameobject.pack()

quitb=tkinter.Button(frameobject,text="QUIT",command=relief_attrib.destroy)
quitb.pack(side=BUTTON)
mainloop()
```

Output:




```
Coordinates [Entering frame] at x=166,y=95
Coordinates [Exiting frame] at x=230,y=39
Coordinates [Entering frame] at x=33,y=94
Coordinates [Exiting frame] at x=28,y=144
```

#finding the key pressed in keyboard

```
from tkinter import *
#finding key pressed in keyboard
def presskey(label):
```

```
    value=label.char
    print(value,'button is pressed')


mainwindow=Tk()
mainwindow.geometry('300x150')
mainwindow.bind('<Key>',lambda i:presskey(i))
mainloop()
```

Output:

```
========== RESTART: C:\Users\Admin\Desktop\12333.py ==========
g button is pressed
h button is pressed
. button is pressed
k button is pressed
a button is pressed
y button is pressed
s button is pressed
a button is pressed
n button is pressed
1 button is pressed
2 button is pressed
 button is pressed
# button is pressed
 button is pressed
 button is pressed
 button is pressed
 button is pressed
 button is pressed
```

ASSIGNMENT-3

Create a Simple Calculator using Tkinter

```
# Python program to create a simple GUI
# calculator using Tkinter

from tkinter import *

# globally declare the expression variable
expression = ""

def press(num):
    global expression
    expression = expression + str(num)
    equation.set(expression)

# Function to evaluate the final expression
def equalpress():

    try:
        global expression
```

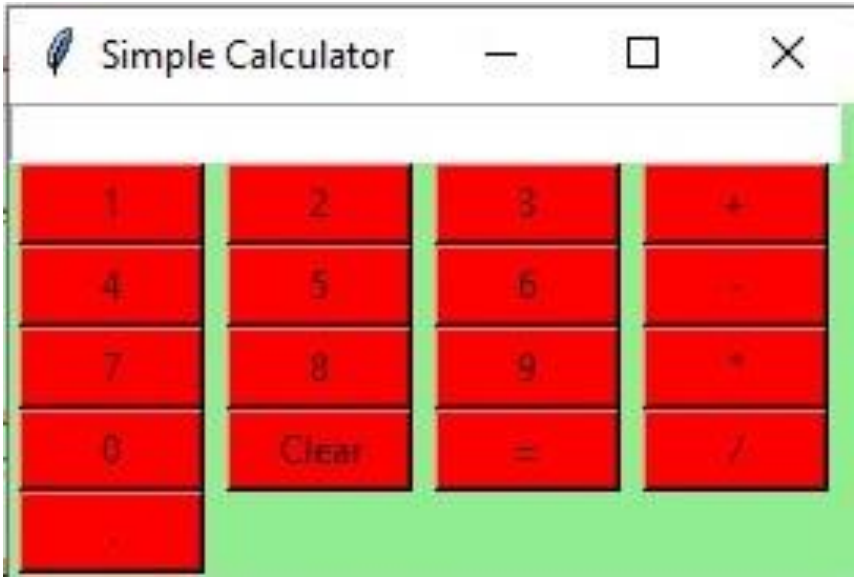
```
            total = str(eval(expression))
            equation.set(total)
            expression = ""

        except:

            equation.set(" error ")
            expression = ""


# Function to clear the contents
# of text entry box
def clear():
        global expression
        expression = ""
        equation.set("")


# Driver code
if __name__ == "__main__":
        # create a GUI window
        gui = Tk()

        gui.configure(background="light green")
        gui.title("Simple Calculator")
```

```
    gui.geometry("270x150")
    equation = StringVar()
    expression_field = Entry(gui, textvariable=equation)
    expression_field.grid(columnspan=4, ipadx=70)


button1 = Button(gui, text=' 1 ', fg='black', bg='red',
                                command=lambda: press(1), height=1, width=7)
    button1.grid(row=2, column=0)

    button2 = Button(gui, text=' 2 ', fg='black', bg='red',
                                command=lambda: press(2), height=1, width=7)
    button2.grid(row=2, column=1)

    button3 = Button(gui, text=' 3 ', fg='black', bg='red',
                                command=lambda: press(3), height=1, width=7)
    button3.grid(row=2, column=2)

    button4 = Button(gui, text=' 4 ', fg='black', bg='red',
                                command=lambda: press(4), height=1, width=7)
    button4.grid(row=3, column=0)

    button5 = Button(gui, text=' 5 ', fg='black', bg='red',
                                command=lambda: press(5), height=1, width=7)
    button5.grid(row=3, column=1)
```

```
button6 = Button(gui, text=' 6 ', fg='black', bg='red',
                         command=lambda: press(6), height=1, width=7)
button6.grid(row=3, column=2)

button7 = Button(gui, text=' 7 ', fg='black', bg='red',
                         command=lambda: press(7), height=1, width=7)
button7.grid(row=4, column=0)

button8 = Button(gui, text=' 8 ', fg='black', bg='red',
                         command=lambda: press(8), height=1, width=7)
button8.grid(row=4, column=1)

button9 = Button(gui, text=' 9 ', fg='black', bg='red',
                         command=lambda: press(9), height=1, width=7)
button9.grid(row=4, column=2)

button0 = Button(gui, text=' 0 ', fg='black', bg='red',
                         command=lambda: press(0), height=1, width=7)
button0.grid(row=5, column=0)

plus = Button(gui, text=' + ', fg='black', bg='red',
                        command=lambda: press("+"), height=1, width=7)
plus.grid(row=2, column=3)
```

```
minus = Button(gui, text=' - ', fg='black', bg='red',
                        command=lambda: press("-"), height=1, width=7)
minus.grid(row=3, column=3)

multiply = Button(gui, text=' * ', fg='black', bg='red',
                                command=lambda: press("*"), height=1, width=7)
multiply.grid(row=4, column=3)

divide = Button(gui, text=' / ', fg='black', bg='red',
                                command=lambda: press("/"), height=1, width=7)
divide.grid(row=5, column=3)

equal = Button(gui, text=' = ', fg='black', bg='red',
                        command=equalpress, height=1, width=7)
equal.grid(row=5, column=2)

clear = Button(gui, text='Clear', fg='black', bg='red',
                        command=clear, height=1, width=7)
clear.grid(row=5, column='1')

Decimal= Button(gui, text='.', fg='black', bg='red',
                                command=lambda: press('.'), height=1, width=7)
Decimal.grid(row=6, column=0)
gui.mainloop()
```

Output:



PRACTICAL -9

DATABASE CONNECTION

```
import sqlite3
#connecting to the database
#If the databse does not exist, then it will be created
#and then a database object will be returned.
conn_db =sqlite3.connect('cs.db')

print ("Opened database successfully")

cursor_object = conn_db.cursor()

print("Cursor object created successfully")

#conn_db.execute('drop table computer')
conn_db.execute(''' CREATE TABLE computer
            (ROLLNO INT PRIMARY KEY   NOT NULL,
            NAME         TEXT       NOT NULL,
            SUBJECT1     INT   NOT NULL,
            SUBJECT2     INT   NOT NULL,
            GRADE    CHAR(5),
```

```
            SGPI    REAL);'")
print("Table created")


conn_db.execute("INSERT INTO computer
(ROLLNO,NAME,SUBJECT1,SUBJECT2,GRADE,SGPI ) \
        VALUES(1001,'LOKESH',85,99,'A',9.20)");
conn_db.execute("INSERT INTO computer
(ROLLNO,NAME,SUBJECT1,SUBJECT2,GRADE,SGPI ) \
        VALUES(1002,'RAMESH',75,69,'B',8.00)");
conn_db.execute("INSERT INTO computer
(ROLLNO,NAME,SUBJECT1,SUBJECT2,GRADE,SGPI ) \
        VALUES(1003,'SURESH',95,99,'0',10.0)");



print("Records inserted into the database")



cursor_obj=conn_db.execute("Select
ROLLNO,NAME,SUBJECT1,SUBJECT2,GRADE,SGPI from computer")


for row in cursor_obj:
    print("ROLL NO = " ,row[0])
    print("NAME = ",row[1])
    print("SUBJECT1 =" ,row[2])
    print("SUBJECT2 =  ",row[3])
    print("GRADE = ",row[4])
    print("SGPI=",row[5],"\n")
```

```
#print(cursor_obj.fetchall())


conn_db.commit()

conn_db.close()
```

OUTPUT:

```
Opened database successfully
Cursor object created successfully
Table created
Records inserted into the database
ROLL NO =  1001
NAME =  LOKESH
SUBJECT1 = 85
SUBJECT2 =   99
GRADE =  A
SGPI= 9.2

ROLL NO =  1002
NAME =  RAMESH
SUBJECT1 = 75
SUBJECT2 =   69
GRADE =  B
SGPI= 8.0

ROLL NO =  1003
NAME =  SURESH
SUBJECT1 = 95
SUBJECT2 =   99
GRADE =  0
SGPI= 10.0
```

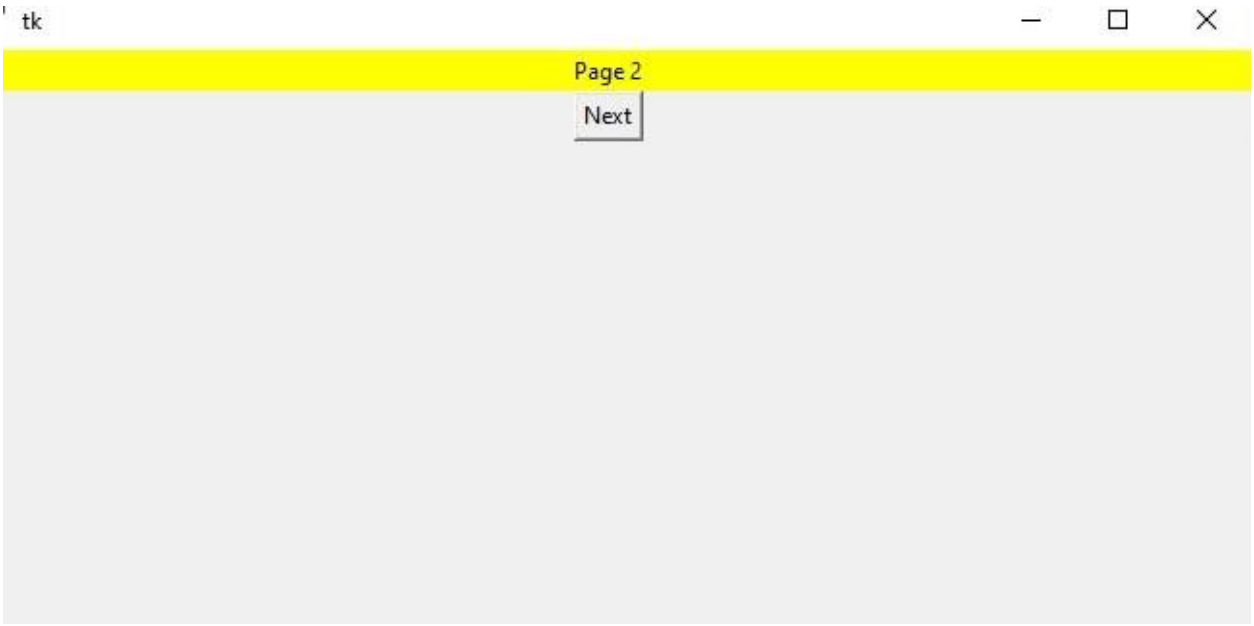
ASSIGNMENT-4

TRAVERSE IN PYTHON

```
import tkinter as tk
def show_frame(frame):
   frame.tkraise()
window=tk.Tk()
window.state('zoomed')
window.rowconfigure(0,weight=1)
window.columnconfigure(0,weight=1)
frame1=tk.Frame(window)
frame2=tk.Frame(window)
frame3=tk.Frame(window)
for frame in (frame1,frame2,frame3):
   frame.grid(row=0,column=0,sticky='nsew')
#=============frame1 code======================
frame1_title=tk.Label(frame1,text="Page 1",bg="red")
frame1_title.pack(fill='x')

frame1_btn=tk.Button(frame1,text="Next",command=lambda:show_frame(frame2
))
frame1_btn.pack()

frame2_title=tk.Label(frame2,text="Page 2",bg="yellow")
frame2_title.pack(fill='x')
```

```
frame2_btn=tk.Button(frame2,text="Next",command=lambda:show_frame(frame3
))
frame2_btn.pack()


frame3_title=tk.Label(frame3,text="Page 3",bg="green")
frame3_title.pack(fill='x')


frame3_btn=tk.Button(frame3,text="Next",command=lambda:show_frame(frame1
))
frame3_btn.pack()


window.mainloop()
```

Output:


tk
Page 2
Next


tk
Page 3
Next