# Sem II

## Practical no: 2

Topic :- Demostrate the use of different file accessing modes, different attributes read methods:-

Algorithms:-

step1 :- Create a file object using open method and use write mode to write a some contents onto the file & close the file.

step2 : open the file in read mode and than use read(), readlines and readline and store the output in variable.

step3 : Now use the fileobject for finding the name of the file, the file mode in which its opened whether the file is still open or close.

step4 :- Now open the fileobject in write mode write some another content close subsequently then again open the file obj in 'w+' mode.

step 5:- open fileobj in read mode display the updated written contents and close open file again in 'r+' mode with parameter passed and display the output subsequently.

step 6: Now open file obj in append mode open write method write contents close the fileobj again open the fileobj in read mode and display the appending output

step 7: open the fileobj in readmode. declare a variable and perform fileobject dot tell method and store the output consequently in variable.

b= fileobj.closed
print("(close) attribute:", b)
>>> ('(close) attribute:', 'True')

c) c= fileobj.mode
print("file mode:", c)
>>> ('file mode:', 'r')

d) d = fileobj.softspace
print("softspace", d)
>>> ('softspace:', 0)

os

seek

```
# file open ("abs.txt", "w")
fileobj = open("abc.txt", "w")
fileobj.write("capital of india \n my name is \n he is
                                    very bad")

fileobj.close()
fileobj = open("abc.txt", "r")
pos1 = fileobj.seek(0,0)
str1 = fileobj.read(20)
print(str1)
pos2 = fileobj.seek(0,1)
str2 = fileobj.read(10)
print(str2)
pos3 = fileobj.seek(2,0)
str3 = fileobj.read(5)
print(str3)
```

```
>>> capital of india
    my
    name is
    he
    pital
```

length

```
# fileobj = open("abc.txt", "r")
sta = fileobj.readlines()
print("output:", sta)
for line in sta:
        print(len(line))

fileobj.close()
```

```
output:
>>> 17
    11
    14
```

step 8: use the seek method with the arguments with opening the file obj in read mode & closing subquently

step 9: open fileobj with read mode also use the readlines method and store the output consequently in and point the same for counting the length use the for conditional statement and display the length

6/12/19

# Practical no: 2

Aim: Write a program using iterable object for displaying the numbers in the range 1 to 20

Algorithm

step1:: Define an iter() with argument and initialize the value and return that value.

step2: Define the next() with an argument and compare the upper limit by using a conditional statement.

step3: Now create an object of the given class and pass this object in the iter()

Code

```
Class myrange:
    def __iter__(self):
        self.a = 1
        return self
    def __next__(self):
        if self.a <= 30
            x = self.a
            self.a += 1
            return x
        else:
            raise StopIteration

myclass = myrange()
myiter = iter(myclass)
for x in myclass:
    print x
```

```
>>>
1
2
3
4
5
6
7
8
9
10
11
...
```

Code:

```
def __iter__(self):
        self.f = 1
        return self
def __next__(self):
        if self.f <= 10:
            num = self.f
            self.f += 1
            fact = 1
            for i in range(1, num+1):
                fact = fact * i
            print(self.f-1, "! =", fact)
        else:
            raise StopIteration
f = fact()
x = iter(f)
for j in x:
    print(j)
```

```
>>> 1 ! = 1
    2 ! = 2
    3 ! = 6
    4 ! = 24
    5 ! = 120
    6 ! = 720
    7 ! = 5040
    8 ! = 40320
    9 ! = 362880
    10 ! = 3628800
```

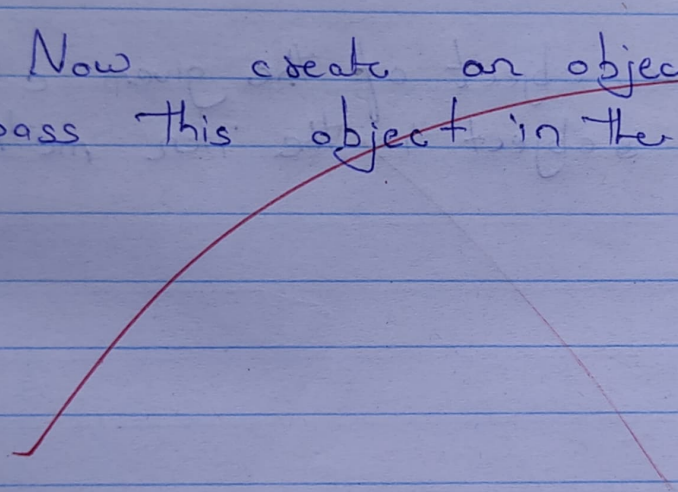
Write a program using iterable concept find the factorial of the number value and return that value in the range of 10

Algorithm:

step1: Define an iter() write argument and initialize the value and return that value.

step2: Define the next() with an argument and compare the upper limit by using a conditional statement

step3: Now create an object of the given class and pass this object in the iter method.



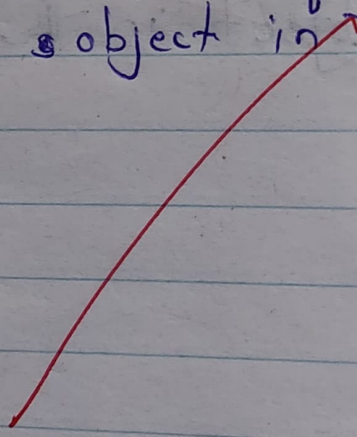
Write a program using the iteration for calculating powers of 2 which is $2^1, 2^2, 2^3$ ...

Algorithm.

step1: Define an iter() with argument and initialize the value and return that value

step2: Define the next() with an argument and compare the upper limit by using a conditional statement

step3: Now create an object of the given class and pass this object in the iter method.



Code:

```
class PowTwo:
    def __iter__(self):
        self.n = 0
        return self
    def __next__(self):
        if self.n <= 10:
            result = 2**self.n
            self.n += 1
            return result
        else:
            raise StopIteration
myclass = PowTwo()
mydata = iter(myclass)
for x in mydata:
    print(x)
```

```
>>>
1
2
4
8
16
32
```

AS

Codes

```
def acceptage():
    age = int(input("Enter Age"))
    if age >30 or age <10
        raise ValueError
return age
```

Codes

```
arithematic error
while True:
    try:
        x = int(input("Enter class:"))
        break
    except ValueError
        print("Enter Numeric Value")
```

>>> Enter class types
Enter Numeric value
Enter class: 13

Codes

```
try:
    fo.open("abc.txt", "w")
    fo.write("computer Source")
accept IOError.
        print "Error writing onto the file")
else:
    print("operation carried out successfully")
    fo.close.
```

>>> operation carried out successfully

# Practical : 3



Topic: Exceptions

Algorithm

step 1: Except the Arithmetic Error. Use the try block and accept the input using input method and convert it into integer datatype and subsquetly terminate the block.

step 2: Use the accept block with the reception name as value error and display the appropriate message if the suspesious code is a part of the try block.

2) Write a program for accepting file in a given mode and use environment error as exception for the given input.

step 1: Within the try block open the file using the write mode and write some content onto the file.

step 2: Use the accept block with IO Error and display the message regarding missing of the file or incompatibility

step 3: Use the else block to display the message that the operation is carried out succesfully

55

3.) Write a program to check the range of the age of the students in a given class and if the age do not fall in the given range use the value error exception

Algorithm:

step 1: Define a function which will accept the age of a student from the standard input.

step 2: Use the if condition to check wheather the input ags falls in the given age and if so turn the age else use the value error exception

step 3: Define while loop to check wheather the expression holds true use the try block to accept the age of student and terminate the looping condition

step 4: Use the except with Valueerror and print the message not a valid age.

Code:

```
def acceptage():
    age = int(input("Enter Age"))
    if age>30 or age<10:
        raise ValueError
    return age
valid = False
while not valid
    try:
        age = acceptage()
        valid = True
    except ValueError:
        print("Not a Valid age")
```

>>>

```
Enter age: 10
Not a valid age
Enter age: 18
```

Code

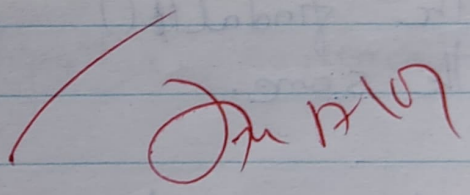
```
def assert (n):
    assert (len(n) == 0)
    print ("list is empty")

var 1 = []
print (assert_(var))
```

>>> list is empty
None

4) Write program using the assert method to check if the list elements are empty.

step 1: Define a function which accepts an argoments and then check using the assert statement wheather the given list is an empty list and accordingly return the message.

step 2: Close the function and in the body of program define certain elements in the list and take some appropriate action.



Topic: Regular Expression

Algorithm:-

step1: Import re module declare pattern and declare sequence use match method with declare arguments. If arguments matched then print the same otherwise print pattern NOT FOUND!

step2: Import re module declare pattern with literal and meta character Declare string value use the findall() with arguments and print the same.

step 3: Import re module declare pattern with meta character use the split() and print the output

step 4: import re module declare string and accordingly declare pattern replace the letter space with no-space use sub() with argument and print the string without

Code

```
# no-space
import re
string = 'abc def ghi'
pattern = r'\s+'
replace = ''
vi = re.sub(pattern, replace, string)
print(vi)
```

```
>>> abcdefghi
```

```
# group()
import re
sequence = 'python is an interesting language'
v = re.search('\Apython', sequence)
print(v)
vi = v.group()
print(vi)
```

```
>>> <_sre.SRE_Match object; span=(0, 6), match='python
python
```

# verifying the given set of phone-numbers.

```
import re
list = ['8854692135', '7859612549', '9270716145']

for value in list:
    if re.match(r'[8-9]{1}[0-9]{9}',
                value or len(value) == 10):
        print("criteria match for cell number:")
    else:
        print("criteria failed:")
```

```
>>> criteria match for cell number:
criteria failed:
criteria match for cell number:
criteria failed:
```

# vowels

```
import re
str: 'plant is life overall'
output: re.findall(r'\b[aeiouAEIOU]\w+', str)
print(output)
```

```
>>> ['is', 'overall']
```

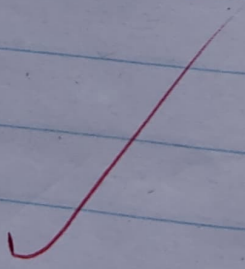
step5:- import re module declare a sequence use search method for finding subsequently use the group() with dot operator as search() gives memory coration using group() it will show up the match string

step 6: import re module declare list with numbe use the conditional statement here we have use up the for condition for checking first number is either 8 or 9 and next and next numbe is one in range 0 to 9 and check.

step 7: import re module declare a string use module with findall() for finding the vowels in the string and declare the same

step 8: import re module declare the host domain name declare pattern for separate the host & domain name. use the findall and print the output respectively.

step 9: import re module enter a string use pattern ~~for to separating~~ to display only two elements of the particular string use findall() declare text variables with initial value as zero use for condition and subsequently use the if condition check wheather condition satisfy add up the or else increment value. and display the value subsequently.



# host and domain

```
import re
seq = 'abc.tcsc@edu.com, xyz@gmail.com'
pattern = r'[\w\.-]+[\w\.-]'
output = re.findall(pattern, seq)
print(output)
```

```
>>> ['abc.tcsc', 'edu.com', 'xyz', 'gmail.com'].
```

# counting of first two letter

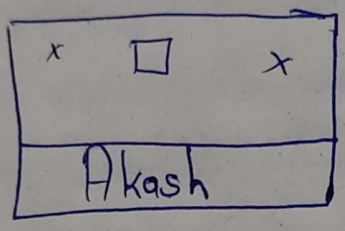
```
import re
s = 'mr.a, ms.b, ms.c, mr.t'
p = r'[/ms/mr/]+'
o = re.findall(p, s)
print(o)
m = 0
f = 0
for v in o:
    if(v == 'ms'):
        f = f+1
    else:
        m = m+1
print("no of males is:", m)
print("no of females is:", f)
```

```
>>> ['mr', 'ms', 'ms', 'mr']
    no. of males is : 2
    no of females is : 2
```

1) Code:-

```
from tkinter import *
root = Tk()
l = Label(root, text = "Akash")
l.pack()
root.mainloop()
```

Output:-




2) Code:-

```
from tkinter import *
root = Tk()
l = Label(root, text = "Python is best")
l.pack(side = LEFT, pady = 30)
L1 = Label(root, text "Akash", bg = "red", fg = "blue")
L1.pack(side = TOP, ipady = 40)
m = Label(root, text = "cs")
m.pack(side = LEFT, padx = 20)
m2 = Label(text, "shell", bg = "green", fg = "black"
m2.pack(side = LEFT, ipadx = 50)
root.mainloop()
```

# Practical: 5

Aim: GUI components.

step 1: Use the tkinter Library for importing the feature of text widget.

step 2: Create a variable from text method and position it on parent window.

step 3: Use the pack method along with the object created from text method.

step 4: Use the main loop method for triggering of corresponding events.

step 5 Use the tkinter library for importing the feature of text widget.

step 6: Create a variable from text method and position it onto patent window.

step 7: Use the pack method along with the object created from text method and use the parameter

step 8: side = "LEFT" padx = 20
side = "LEFT" padx = 30
side = "TOP" padx = 40
side = TOP padx = 50

step 9: Use the main loop method for triggering of corresponding events

step 10: Now repeat the above steps with the label which takes the following

1.) Text attribute 2.) Bg (background) colour
3.) Fg (colour etc.) Name of the parent window

for sele

step1: Use the

step2: Define a function

given selection need of the multiple

step3: Use the configuration method along with the

object and call the variable as an argument with

the method.

step4: Now define the parent window and define the option

using control variable

step5: Now create an object from the radiobutton meth

which will take the foll arguments

1) positioning on parent window

2) Defining the text variable [1, 2, 3, 4]

step6: Pack method for the corresponding radio objects so

created and specify the attribute as an anchor attrib

step7: Define the label object for corresponding meth

and place it on parent window

step8: Import the relevant method from tkinter

library.

step9: Define the object corresponding to the

window & define the size of parent window

terms of no of pixels

step10: Now define the frame object from the

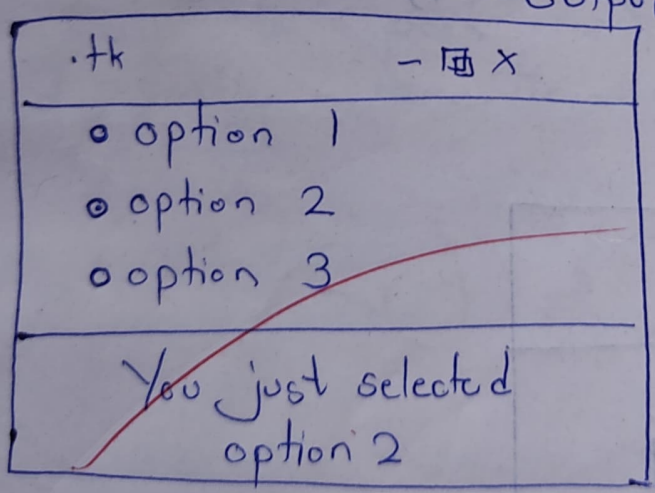
and place it onto the parent window.

# Radio button:

## Code

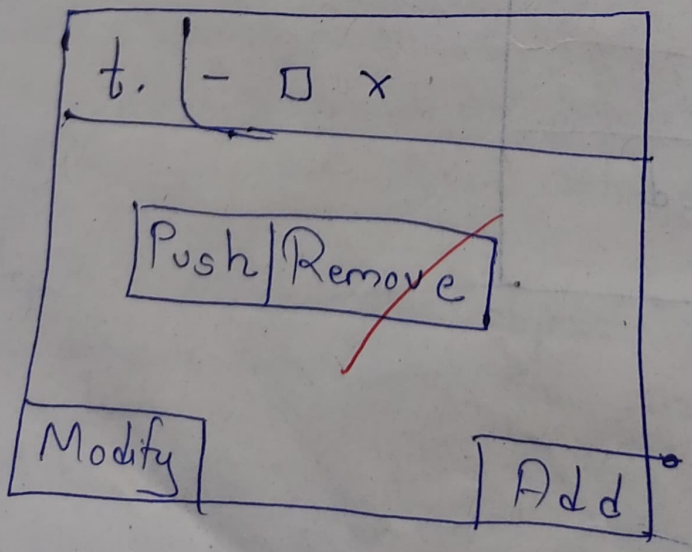
```
from tkinter import *
root = Tk()
def sel():
    selection = "you selected the option " + str(var.get())
    l = label.config(root, text = "selection", justify = LEFT)
    l = pack (anchor = S)
    V = IntVar()
r1 = Radiobutton (text = "option 1", variable = "v", value = "1", command = sel)
r1 = pack()
r2 = Radiobutton (text = "option 2", variable = "v", value = "2", command = sel)
r2 . pack()
r3 = Radiobutton ( text = "option 3", variable = "v", value = 3, command = sel)
r3 = pack()
label = Label (root)
label . pack ()
root . mainloop()
```

Output.



```
# from object
from tkinter import*
root = Tk()
frame = Frame(root)
frame.pack(padx=20, pady=50, side=TOP)
leftframe = Frame(root)
leftframe.pack(side = LEFT)
rightframe = Frame(root)
rightframe.pack(side = RIGHT)
buttonpush = Button(frame, text="Push", fg="blue", bg="red")
buttonpush.pack(side = LEFT)
buttonremove = Button(frame, text="Remove", fg="green", bg="ye
buttonremove.pack(side= BOTTOM)
buttonadd = Button(rightframe, text = "Add")
buttonadd.pack(side = LEFT)
buttonmodify = Button(leftframe, text = "Modify")
buttonmodify.pack(side = RIGHT)
root.mainloop()
```

output



step 11: Create another frame object left frame and put it onto parent window on its left side

step 12: Similarly define the right frame and subsequently define the button object placed onto the given frame with the attribute as text active bg & fg

step 13: Now use the pack method along with the side attribute

step 14: Create the button object corresponding to modify option and put it into the frame object with side equal to right attribute se

step 15: Add another button and put it on right frame object and term it as

step 16: use the pack method for all the object and finally use the main loop method

24/07

* Message box Method:

Step1: Import the relevant method from tkinter library

step2: Define a function and use the message box along with the methods available which contains one or more arguments

step3: Thus different options which are available are showwarning(), showerror(), askYesNo(), askquestion(), askokcancel

step4: Create object from button method and place it onto the specified window with the title of the button specified and the corresponding event called for triggering.

step5: Use the pack method to display the button widget and finally use mainloop method

Relief style:-

Step1: Use the button object with the following attributes
1. The parent window
2. Text attribute
3. Relief

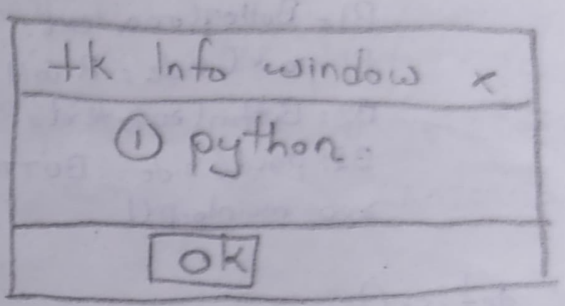
step2: Use the corresponding pack method for the respective button objects and trigger the corresponding event

step3: Finally use the mainloop method.

Code

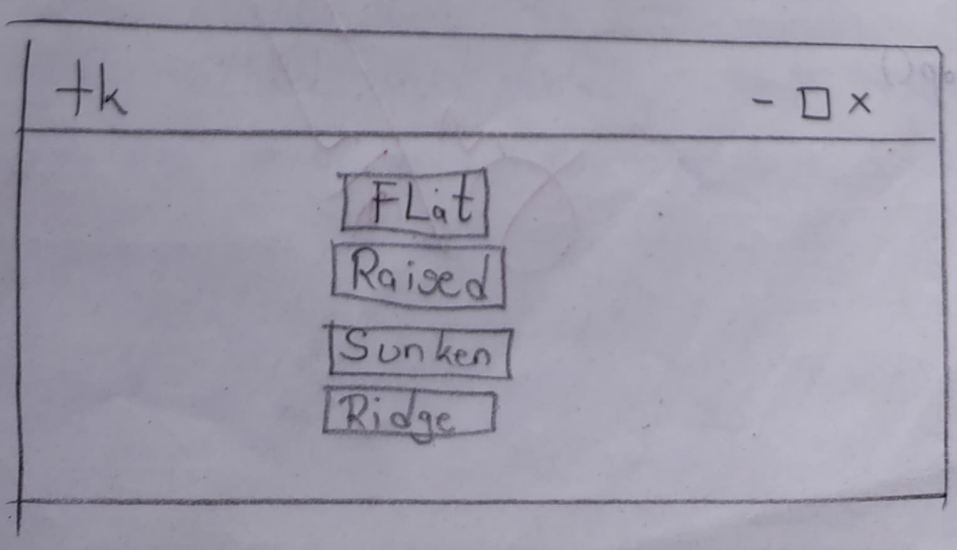
```
from Tkinter import *
import tkmessageBox
root = Tk()
def fun():
    tkmessageBox.showinfo("info window", python)
b1 = Button(root, text = "title", command = fun)
b1.pack
root.mainloop()
```

output




Code

```
from Tkinter import *
root = Tk()
b1 = Button(root, text = ("Flat", relief = FLAT)
b1.pack()
b2 = Button(root, text = "Raised", relief = RAISED)
b2.pack()
b3 = Button(root, text = "Sunken", relief = SUNKEN)
b3.pack()
b4 = Button(root, text = "Ridge", relief = RIDGE)
b4.pack()
root.mainloop()
```



# Code:

```
from tkinter import*
root = Tk()
def main():
    roo = Tk()
    roo.config(bg = "black")
    roo.title("Main")
    roo.minsize(200,200)
    l = Label(roo, text = "Music")
    l.pack()
    l1 = Label(roo, text = "music give us relation")
    l1.pack()
    B1 = Button(roo, text = "Next", command = se)
    B1.pack(side = RIGHT)
    B2 = Button(roo, text = "Terminate", command = ter)
    B2.pack(side = BOTTOM)
    roo.mainloop()

def se():
    master = Tk()
    master.config(bg = "Green")
    master.title("Page 2")
    master.minisize(400,400)
    l2 = Label(master, text = "Workout")
    l2.pack()
    l3 = Label(master, text = "workout give u healthy life")
    l3.pack()
    B3 = Button(master, text = "Prev", command = main)
    B3.pack(side = LEFT)

def ter():
    quit()

B5 = Button(root, text = "ENTER", command = main)
B5.pack()
root.mainloop()
```

# Transversing

step1: Define the function and create an object of the given window by using the three methods named config, title & minsize.

step2: Create a button object and use the text and command attribute for trigering the given event and used grid method along with internal & external poding.

step3: Define second function corresponding to second window with attributes config title, minsize for the window object and define one button object which will shift the focus onto the third window

step4: Create third window object and in this create two button object for moving onto first window for restarting the process and second button for terminating.

step5: Define a function for termination & call the quit method and finally call the first form created and trigger mainloop method.

Displaying the image:-

Algorithm:-

step1: Create an object corresponding to the parent window and use the following 3 methods. Title minsi- config

step 2: Create a leftframe object from the frame method and place it onto the parent window with the height, width and the bg specified. subsequently

step 3: Now create a rightframe object from the frame method with the width, height specified with row & colum

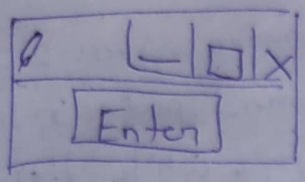
step 4: Create a label object from the label metho and place it onto the leftframe with text attribute denoting the original image with relief attribute use as RAISED value and subsequently use grid method with padding

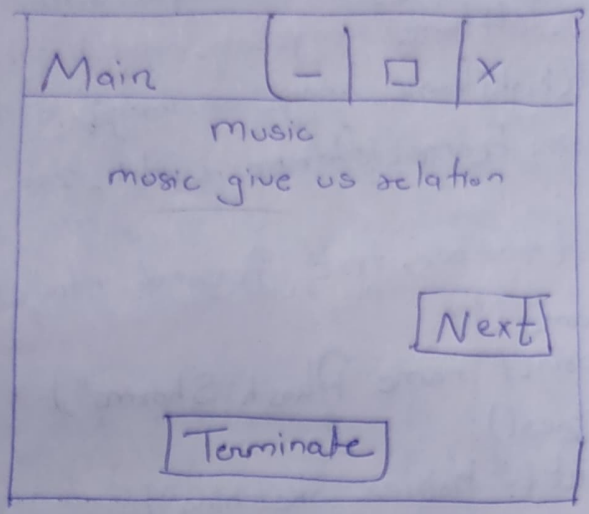
step 5: Now use the photo image method the file attribute specified

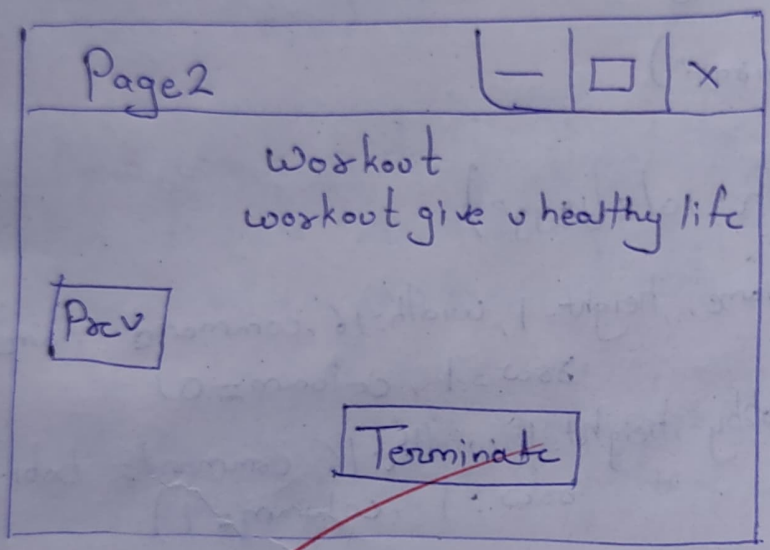
step 6: Use the sub sample method with the of the image and give the x, y co-ordinate

## ⇒ Output

Transversing



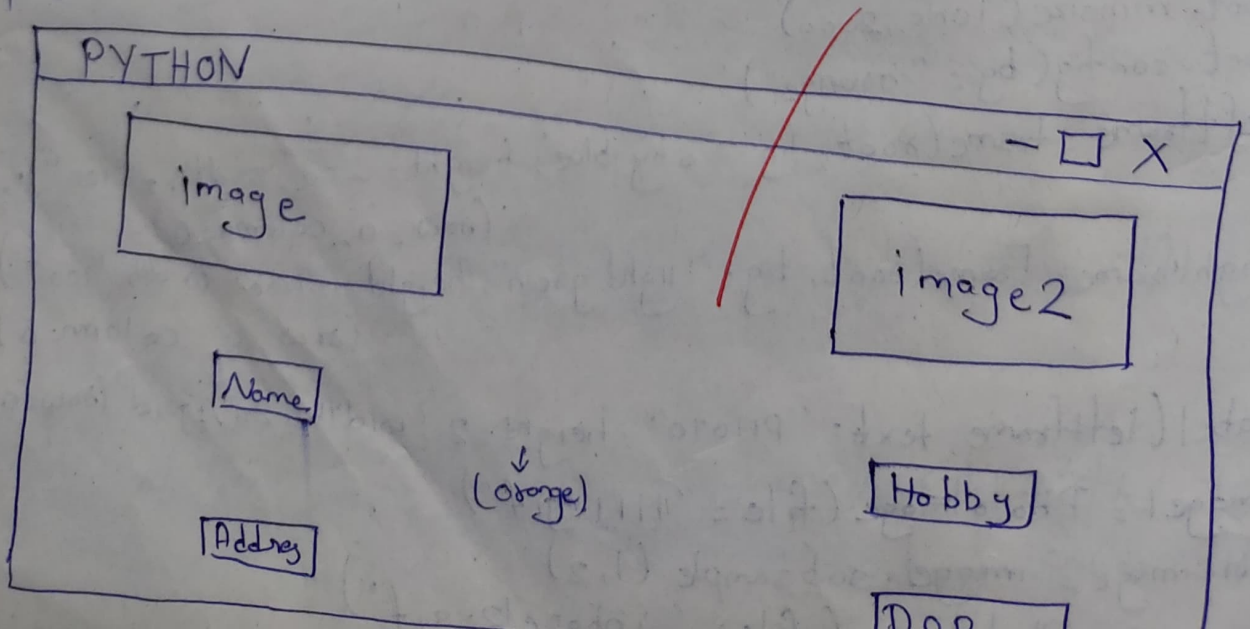




## #Code:-

```
from tkinter import *
root= Tk()
root: title("PY THON")
root. minsize(1000, 2000)
root. config(bg = "orange")
leftframe : Frame(root, bg: "sky blue", height: "200", width = "100").gri
                                                    (row : 0, column= 0)
rightframe: Frame(root, bg: "light green", height: "400", width: "10".g
                                                    (row :0, colom
Label(leftframe, text: "PHOTO", height: 2, width :20).grid (ro
image1 : PhotoImage (file = "1111. gif")
```

```
ori_image = image2.subsample(3,4)
Label(leftframe, image=ori_image).grid(row=0, column=0, padx=20, pady=
Label(rightframe, image=ori_image).grid(row=0, column=1, padx=30, pady=2
toolbox = Frame(leftframe, width=200, height=400, bg="white").grid(
                                        row=2, column=0)
Label(toolbox, text="Personal Info", height=2, width=20, relief=RAISED
def name():
    print("name: Akash Sharma")
def hobies():
    print("hobies: Sketching")
def add():
    print("Address: Virar")
def dob():
    print("date of birth: 10/8/1999")
Button(toolbox, text="Name", height=1, width=16, command=name).grid(
                                        row=1, column=0)
Button(toolbox, text="Hobby", height=1, width=16, command=hobies).grid(
                                        row=1, column=1)
Button(toolbox, text="Address", height=1, width=16, command=add).grid
                                        (row=2, column=0)
Button(toolbox, text="DOB", heigth=1, width=16, command=dob).grid
                                        (row=2, column=1)
```

Output



step 7:- Use the label method and and position it onto the left frame and display the image after the sampling and use the grid method for the positioning in the first row.

step 8: Create another label object positioning it onto the rightframe specifying the image and background attribute with row and column (0,0)

step 9: Now create a toolbox object from the frame method and position it onto the leftframe with the height and width specified and position it onto the second row.

step 10: Now define the various function for different toolbox options provided in the leftframe

step 11: Create the label method position it onto the toolbox with the next title as personal information and position it

step 12: Now make use of mainloop method.

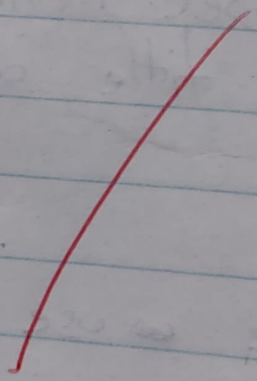
Spin box:-

step 1: Use the tkinter library to import the relevant method.

step 2: Create the parent window object

step 3: Create an object from a spin box method and place it on to the parent window with the options specified.

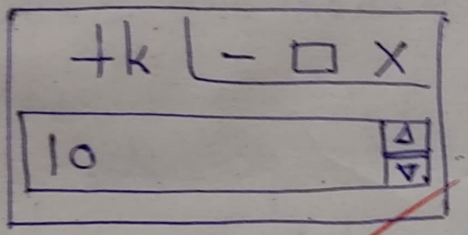
step 4: Now use the pack method to make the object visible onto the parent window and called the mainloop method.



# code

```
from tkinter import *
top = Tk()
s1 = Spinbox(top, from_=0, to=10)
s1 = pack()
top.mainloop()
```

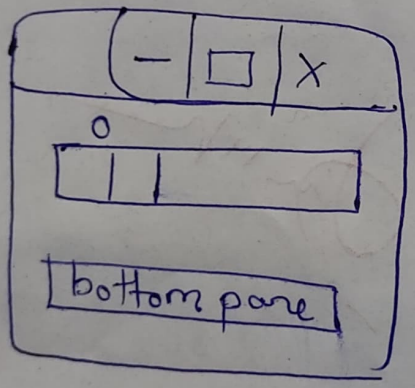
=> Output :-



Code:

```
from tkinter import *
m1 = PanedWindow()
m1.pack(fill=BOTH, expand=10)
e = Entry(m1, text="left pane", bd=10)
m2 = PanedWindow(m1, orient=VERTICAL)
m1.add(m2)
top = Scale(m2, orient=HORIZONTAL)
m2.add(top)
bottom = Button(m2, text="bottom pane", command=top)
m2.add(bottom)
m2.pack()
m2.mainloop()
```

Output



Paned Window:-

step1: Create an object from the paned window method and use the pack method to make this object visible.

step2: Now create an object from the entry widget and placed it onto the paned window and use the add method. Similarly create an object of a panned window and

step3: Create an object from the ~~entry~~ scale widget & place it onto the preceding paned window and use the add method accordingly.

step4: Create a button widget and placed it onto the paned window define a functionality along with the button widget.

step5: Use the pack method & mainloop method for the corresponding event to trigger.

14/2

Canvas widget..

step1: Create an object from the canvas widget by using the attribute height, width, bg colour & the parent window object.

step2: Use the corresponding method for drawing the simple geometrical shape like arc, oval & line and specify the co-ordinate values.

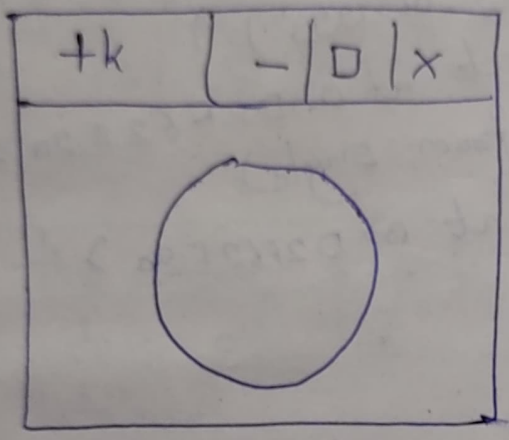
Similarly use the create line & create oval method along with the co-ordinates values & the fill attribute for specifying the colour.

step3: finally use the pack & mainloop method.

#for oval

```
from Tkinter import *
master= Tk()
c= Canvas(master= bg: "white", height =1000, width= 1500)
c.pack()
c.create_oval(50, 50, 400, 400)
c.mainloop()
```

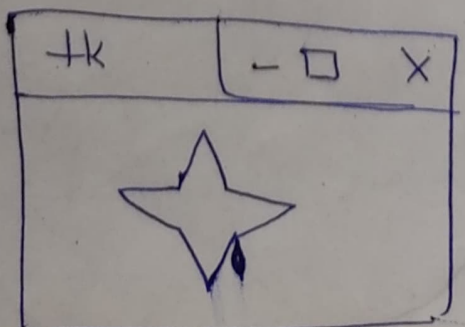
Output




#for polygon.

```
from tkinter import *
master = Tk()
a= Canvas(master, width =200, height= 200)
a.pack()
points= [100,140, 110,110,140, 100,110, 90,100, 60,90, 90,60,
90,110]
a.create_polygon(points, outline= "green", fill= "yellow", width
a.mainloop()
```

output



## Programs:

9h

```
>>> import os, sqlite3
>>> conn = sqlite3.connect("student.db")
>>> cur = conn.cursor()
>>> cursor.execute("Create table info(RNO int, Name text, Do
    <sqlite3.cursor object at 0x02F62Fa0>
>>> cursor.execute("insert into single value(01, "Akash", "10-08-
            (02, "Rishi", 05-06-1999")")
    <sqlite3.cursor object at 0xD2F62F20>
>>> cursor.execute("select DoB from single)
    <sqlite3.cursor object at 0xD2F62F20>
>>> cur.fetchall()
    cur.close()
conn.commit()
cursor.execute("Select Name from student")
print(cursor.fetchall())
cursor.close
```

Output:-

('akash sharma', 'Male', '1847')

# Practical no: 6

Topic: Database connectivity.

step1: Import the relevant libraries for the database connection and the operating system functionalit

step2: Now create an object for making the connectio to the given database.

step3: Further create an object corresponding to the cursor area execution of the different query statement.

step4: Use the cursor object so created for implemen the structure of the database and the values w the database.

step5: Use the execute method for implementation o select clause for filtering the information.

step6: Now use the fetch all method along with cursor object for display the values on to scree

# * Project *

```
import tkinter
from tkinter import*
import random
from tkinter import messagebox

answers=[
    "python",
    "java",
    "swift",
    "canada",
    "india",
    "america",
    "london",
    "delhi",
    "germany",
    "bhutan",
    "russia",
    "australia",
    "brazil",
    "beijing",
    "mexico",
    "mumbai",
    "tokyo",
    "cairo",
    "japan",
```

```
]

words=[
  "nptoyh",
  "avja",
  "wfsit",
  "cdanaa",
  "aidin",
  "aiearcm",
  "odnlon",
  "dlhei",
  "graeynm",
  "buahtn",
  "irsaus",
  "asuaairlt",
  "brairl",
  "binegij",
  "xcomie",
  "mmauib",
  "tokoy",
  "cioar",
  "ajnpa",
  ]
```

```
num=random.randrange(0,19,1)


def reset():
    global words,answers,num
    num=random.randrange(0,19,1)
    lbl.config(text=words[num])
    e1.delete(0,END)




def default():
    global words,answers,num
    lbl.config(text =words[num])



def checkans():
    global words,answers,num
    var=e1.get()
    if var==answers[num]:
        messagebox.showinfo("Success","This is a correct answer")
        reset()
    else:
        messagebox.showerror("Error","This is not correct")
        e1.delete(0,END)
```

```
root=Tk()
root.geometry("350x400+400+300")
root.title("Jumbbled")
root.configure(bg="#000000")

lbl=Label(root,text="Your here",
      font=("verdana",18),
      bg="#000000",
      fg="#ffffff")

lbl.pack(pady=30,ipady=10,ipadx=10)

ans=StringVar()
e1=Entry(root,font=("verdana",16),textvariable=ans)
e1.pack(ipady=5,ipadx=5)

buttoncheck=Button(root,text="Check",font=("comic sans ms",16),
            width=16,
            bg="#4C4b4b",
```

```
            fg="#6ab04c",
            relief=GROOVE,
            command=checkans)
buttoncheck.pack(pady=40)



buttonreset=Button(root,text="Reset",font=("comic sans ms",16),
            width=16,
            bg="#4C4b4b",
            fg="#EA425C",
            relief=GROOVE,
            command=reset)
 buttonreset.pack()

 default()


  root.mainloop()
```

```
from tkinter import*
import sqlite3


root=Tk()
root.geometry('500x500')
root.title("Registration Form")


Fullname=StringVar()
Email=StringVar()
var=IntVar()
c=StringVar()
var1=IntVar()


def database():
    name1=Fullname.get()
    email=Email.get()
    gender=var.get()
    prog=var1.get()
    country=c.get()
    conn=sqlite3.connect('Form.db')
    with conn:
        cursor=conn.cursor()
    cursor.execute("Create TABLE IF NOT EXISTS Student(Fullname TEXT,Email TEXT,Gender
TEXT,country TEXT,Programming TEXT)")
    cursor.execute("INSERT INTO Student (Fullname,Email,Gender,country,Programming)
VALUES(?,?,?,?,?)",(name1,email,gender,country,prog))
    conn.commit()
```

```
label_0=Label(root,text="Registration form",width=20,font=("bold",20))
label_0.place(x=90,y=53)

label_1=Label(root,text="Fullname",width=20,font=("bold",10))
label_1.place(x=80,y=130)

entry_1=Entry(root,textvar=Fullname)
entry_1.place(x=240,y=130)

label_2=Label(root,text="Email",width=20,font=("bold",10))
label_2.place(x=68,y=180)

entry_2=Entry(root,textvar=Email)
entry_2.place(x=240,y=180)

label_3=Label(root,text="Gender",width=20,font=("bold",10))
label_3.place(x=70,y=230)

Radiobutton(root,text="Male",padx=5,variable=var,value=1).place(x=235,y=230)
Radiobutton(root,text="Female",padx=20,variable=var,value=12).place(x=290,y=230)

label_4=Label(root,text="country",width=20,font=("bold",10))
label_4.place(x=70,y=280)

list1=["Australia","America","Afganistan","Berlin","Bangladesh","India","UK","USA","Nepal","So
Africa"]

droplist=OptionMenu(root,c,*list1)
droplist.config(width=15)
```

```
c.set("select your country")
droplist.place(x=240,y=280)

label_4=Label(root,text="Programming",width=20,font=("bold",10))
label_4.place(x=85,y=330)
var2=IntVar()
Checkbutton(root,text="java",variable=var1).place(x=235,y=330)

Checkbutton(root,text="python",variable=var2).place(x=290,y=330)

Button(root,text="Submit",width=20,bg="brown",fg="white",command=database).place(x=180,y=3
0)

root.mainloop()
```