

Practical-01

- * Objective: Demonstrate the use of different file accessing mode, different attributes read method.

Step:01.

- Create a file object using open method and use the write access mode followed by writing some contents onto the file and then closing the file.

Step:02.

- Now open the file in read mode and then use read(), readline() and readlines() and store the output in variable and finally display the contents of variable.

Step:03:

- Now use the file object for finding the name of the file, the file mode in which it is opened whether the file is still open or close and finally the output of the softspace attribute.

11S

```
fileobj = open("abc.txt", "w") # file open (write mode)
fileobj.write("computer science subjects" + "\n")
fileobj.write("DBMS in python in OS\n") # file write
fileobj.close() # file close.
```



```
fileobj = open("abc.txt", "r") # read mode
# read()
str1 = fileobj.read()
print("The output of read method:", str1)
fileobj.close()
>>> ('The output of read method:', 'computer Science
subjects in DBMS in python in OS\n')
```



```
# read line()
fileobj = open("abc.txt", "r")
str2 = fileobj.readline()
print("The output of readline method:", str2)
fileobj.close()
>>> ('The output of readline method:', 'computer Science
subject\n')
```



```
# readlines()
fileobj = open("abc.txt", "r")
str3 = fileobj.readlines()
print("The output of readlines method:", str3)
fileobj.close()

>>> ('The output of readlines method:', ['computer Science
subjects\n', 'DBMS\n', 'python\n', 'OS\n'])
```

Step.04:-

→ Now open the file object in write mode. Write some another content close subsequently then again open the file object in 'w+' mode that is the update mode and write contents.

Step.05:

→ Open file object in read mode, display the update written contents and close open Again '+' mode with parameter passed and display the output subsequently

Step.06:

→ Now open file object in append mode open write method write content close the file object again. Open the file object in read mode and display the append output.

#file attributes

```
a = fileobj.name
print("name of file (name attribute):", a)
>>> ('name of file (name attribute):' 'abc.txt')

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

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

d = fileobj.softspace
print('soft space', d)
>>> ('soft space:', 0)
```

w+ mode

```
fileobj = open("abc.+xt", "w+")
fileobj.write("Pusheraj")
fileobj.close()
```

write mode

```
fileobj = open("abc.+xt", "wt")
fileobj.write("Python")
fileobj.close()
```

r+ mode

```
fileobj = open("abc.+xt", "rt")
str1 = fileobj.read()
print("output of rt", str1)
fileobj.close()
>>> ('output of rt', 'Pusheraj')
```

#read mode

```
fileobj = open("abc.+xt", "rt")
str2 = fileobj.read()
print("output of read mode!")
>>> ('output of read mode!', 'Pusheraj')
```

Step 07:

- Open the file object in read mode, declare a variable and perform file object dot tell method and store the output consequently in variable.

Step 08:

- Use the seek method with the arguments with opening the file object in read mode and ~~and~~ closing subsequently.

Step 09:

- Open file object with read mode also use the readlines method and store the output consequently in and print the same for counting the length use the for condition statement and display the length.

~~not all~~
~~all~~

```
fileobj=open("abc.txt","w") # fileopen (write mode)
fileobj.write("CompSciencsubject"+'\n')
fileobj.write("Python in DBMS in PS\n")
fileobj.close()
```

```
fileobj=open("abc.txt",'r') # read mode
#readall()
str1 = fileobj.read()
print("The output of read method:", str1)
fileobj.close()
>>>("The output of readmethod:", 'Comp Science subject
    in python in DBMS in PS\n')
```

```
#readline()
fileobj=open("abc.txt",'r')
str2 = fileobj.readline()
print("The output of readline method:", str2)
fileobj.close()
>>>('The output of readline method:', 'Comp. Science
    Subject\n')
```

```
#readlines()
fileobj= open('abc.txt',"r")
str3 = fileobj.readlines()
print("The output of readlines method:", str3)
fileobj.close()
```

Practical: 2

(a) Aim:

To display elements of a tuple using iterator method.

Algorithm:

Step: 1

Form a tuple with certain elements inserted in it.

Step: 2 Use iter method with tuple and assign it to a variable

Step: 3 Use the next method with variable and print elements.

(b) To use iter method with for loop:

Algorithm:

Step: 1: Form a tuple with certain elements inserted in it.

Step: 2: Use the for conditional statement to access each element of tuple.

Step: 3: Print the elements of tuple.

Code:-

```

mytuple = ("Adarsh", "Ankit", "Yash", "Dinesh")
myiter = iter(mytuple)
print(next(myiter))
print(next(myiter))
print(next(myiter))
print(next(myiter))

```

Output:

Adarsh
Ankit
Yash
Dinesh

Code:-

```

mytuple = ("Adarsh", "Ankit", "Yash", "Dinesh")
for a in mytuple:
    print(a)

```

Output:

Adarsh
Ankit
Yash
Dinesh

Drill

Practical-3

Aim:-

Iterators.

Algorithm:

Step-1:

→ Create a tuple with elements that we need to iterate using the `iter` and `next` method the number of time we use the `iter` and `next` method we will get the next iterating ~~ele~~ element in the tuple. Display the same.

Step-2:

→ The similar output can be obtained by using ~~for~~ conditional statement. An iterable variable is to be calculated in for loop which will iterate.

Step-3:

→ Define a function name `square` with a parameter which will obtain output of square value of the given number. In similar way declare var. to get value raised 3 and return the same.

Step-4:

→ Call the declare function using function call.

iter() and next():-

```
mytuple1= ("apple", "orange", "banana")
```

```
myiter1 = iter(mytuple1)
```

```
print(next(myiter1))
```

```
myiter2 = iter(mytuple1)
```

```
print(next(myiter2))
```

```
myiter3 = iter(mytuple1)
```

```
print(next(myiter3))
```

Output

⇒ apple

orange

banana

for loop:-

```
mytuple1= ("Tom", "Staart", "Jerry")
```

```
for x in mytuple1:
```

```
    print(x)
```

Output

⇒ Tom

Staart

Jerry.

Square and Cube:-

```
def square(x):
```

```
    y = x * x
```

```
    return y
```

```
def cube(x):
```

```
    z = x * x * x
```

```
    return z
```

```
count1=[square, cube].
```

```
× [for n in range(5)]:
```

Value set map lambda:

P.S.

Step 5:

- Using for conditional statement specifying the range use the list typecasting with map method, declare a lambda function i.e anonymous function and print the same.

Step 6:

- Declare a list num variable and declare some elements and use the map method with help of lambda function give two arguments display the output.

Step 7:

- Define a function even with a parameter then using conditional statement do check whether the number is even and odd and return the value respectively.

Step 8:

- Define a class and within that define the iter() method which will initialize the first element within the container object.

Step 9:

- Now use the next() and define the logic for displaying odd while.

for i in range(x):

 value = list(map(lambda x: x*i), func))

26

 print(value)

```
↳ [0,0]
    [1,1]
    [4,8]
    [9,27]
    [16,24]
```

#map():-

list num = [0, 2, 3, 4, 5, 6, 7, 8]

listnum = list(map(lambda x: x**2, listnum))

print(listnum)

def even(x):

 if (x%2 == 0):

 return "EVEN"

 else:

 return "ODD"

list(map(even, listnum))

```
↳ [0, 2, 4, 6, 8]
```

#Odd Number:-

class odd

 def __init__(self):

 self.num = 1

 return self

 def __next__(self):

 num = self.num

 self.num += 2

 def __next__(self):

 num = self.num

 self.num += 2

 return num

Step 10:

→ Define an object of a class.

Step 11:

→ Accept a number from the user till which we want to display the odd number.

Step 12:

→ Define a function with if condition if exactly equal to one return else on parameter.

Step 13:

→ Enter a input from the variable and in an empty list use append method for appending the input value.
Print the output variable.

Jyoti 11/11/19

```
my obj = odd()
my iter = iter(my obj)
x = list(input("Enter a number"))
for i in my iter:
    if (i < x):
        print(i)
```

111 Enter a number: 15

```
3  
5  
7  
9  
11
```

Factorial..

```
def f(n):
    if (n == 1):
        return 1
    else:
        return (n * f(n-1))
```

```
n = int(input("Enter a number"))
```

```
list = [ ]
```

```
list.append(n)
```

```
a = map(f, list)
```

```
print("The factorial is:", a)
```

111 Enter a number: 5

The factorial is: 120.

PS

Practical: 3

WAP using the exception block related to the environment error.

Algorithm:

Step 1: Use the try block to define normal character of action for eg. define the file object and open the file in the write mode and write some content onto the file.

Step 2: Use the except block with a i/o error as an environment sector and convey the appropriate error to the user else display the msg that the operation is carried out successfully.

try:

```
fo = open("Pushraj", "w")
```

```
fo.write("India is my country" + "all Indians are  
my brothers and sisters")
```

except IOError:

```
print("CS")
```

else:

```
print("Operation Successfull")
```

Output:

Operation Successfull.

WAP for demonstrating the use of value error in the given program statement.

Step-1:

- Except the value from the user and if it is a valid value display the entered value and terminate the condition by using break statement.

Step-2:

- Define the except block with the value error as a keyword and display the appropriate message.

Step-3:

- We can define the multiple exception using the except statement for finding the different categories of error.

Jyoti ✓

try:

```
fo=open("Pushpraj","w")
```

```
fo.write("all indians are my Indians brother  
and sister")
```

except ValueError:

```
print("Sorry! Invalid number")
```

Except IOError:

```
print("India")
```

else:

```
print("default action")
```

Output:

»» Default Action

while True:

try:

```
fileobj=open("Pushpraj","w")
```

```
fileobj.write("All Indians are my brothers")
```

```
a=int(input("Enter a number :"))
```

```
print(a)
```

```
break
```

except IOError:

print("There is an environmental error")

Except ValueError:

print("The value is invalid")

Output:

»» Enter a number : Pushpraj
The value is invalid

Enter a number : 123456

The value is invalid

Enter a number : 56
56

PS

Practical: 4

Aim: Regular Expression

- Step.1: Import re module declare pattern and sequence use match() method with declared argument and if argument match then print same.
- Step.2: Import re module declare pattern and literal and meta character. Declare string value use.findall() with arguments and print same.
- Step.3: Import re module and declare pattern meta character use split() and print.
- Step.4: Import re module declare string and accordingly and replace blank space with no-space use sub() with 5 arguments.
- Step.5: Import re module declare sequence use search method for finding subsequently use group and search() using memory using group().

1.) match()

```
import re
```

```
pattern = re.compile("CS")
```

```
sequence = "CS is computer Science."
```

```
if re.match(pattern, sequence):
```

```
    print("Match found")
```

```
else:
```

```
    print("Sorry")
```

```
>>> match found.
```

2.) Numerical values.

```
import re
```

```
pattern = re.compile("\d+")
```

```
string = ('Hello 123', 'there', 'yo0')
```

```
output = re.findall(pattern, string)
```

```
print(output)
```

```
>>> ['123', '']
```

3.) split()

```
import re
```

```
pattern = re.compile("\d+")
```

```
string = 'Hello456', 'there23'
```

```
output = re.split(pattern, string)
```

```
print(output)
```

Step 5:-

→ import re module declare list with num. Use the conditional statement here use up for conditional statement. Use if condition for checking number is a and other number are in range of 0 to 9 and check whether entered no is equal to 10.

Step 6:-

→ Import re module and declare a string using module.findall() for finding vowel vowels declare same.

Step 7:-

→ Import re module and declare host domain name declare pattern for separating host and domain name.

Dr. Jyoti

```
# no space  
import re  
String = 'abc efg'  
Pattern = re.compile('abc')  
replace = ''  
ui = re.sub(Pattern, replace, String)  
print(ui)  
abc efg
```

```
# group()  
import re  
Sequence = 'Python is programming language'  
U = re.search('Python', Sequence)  
print(U)  
Ui = U.group()  
print(Ui)
```

```
# Verifying set of phone number  
import re  
list1 = ['9833440010', '9985041609', '895456790']  
for value in list1:  
    if re.match('[9][1-9][0-9]{9}': len(value) == 10):  
        print("Criteria match")  
    else:  
        print("Sorry!!")
```

>>> criteria matched
Sorry !!.

32

vowels

import re

seq = 'abc.txt @ edu.com', xyz@gmail.com'

pattern = re '[\W-] + [\W]'

output = re.findall(pattern, seq)

print(output)

>>> ['abc.txt @ 'edu.com', 'xyz' '' gmail.com']

host domain

import re

str1 = 'plant is life'

output = re.findall(''. b '[aeiouAEIOU]{0,1}', str1)

print(output)

>>> ['s', '']

Practical: 5

Aim: To make use of GUI application along with the basic pack method.

Algorithm:

Step 1:

- use the tkinter library for importing the features of text widget.

Step 2:

- Create a variable from a text variable and position it onto the parent window.

Step 3:

- Use the pack() along with the object created from the text method of use the parameters.

(i) side = TOP , padx = 20 , ipadx = 40 , ipady = 50 .

Step 4:

- Use the mainloop method for triggering corresponding event.

Step 5:

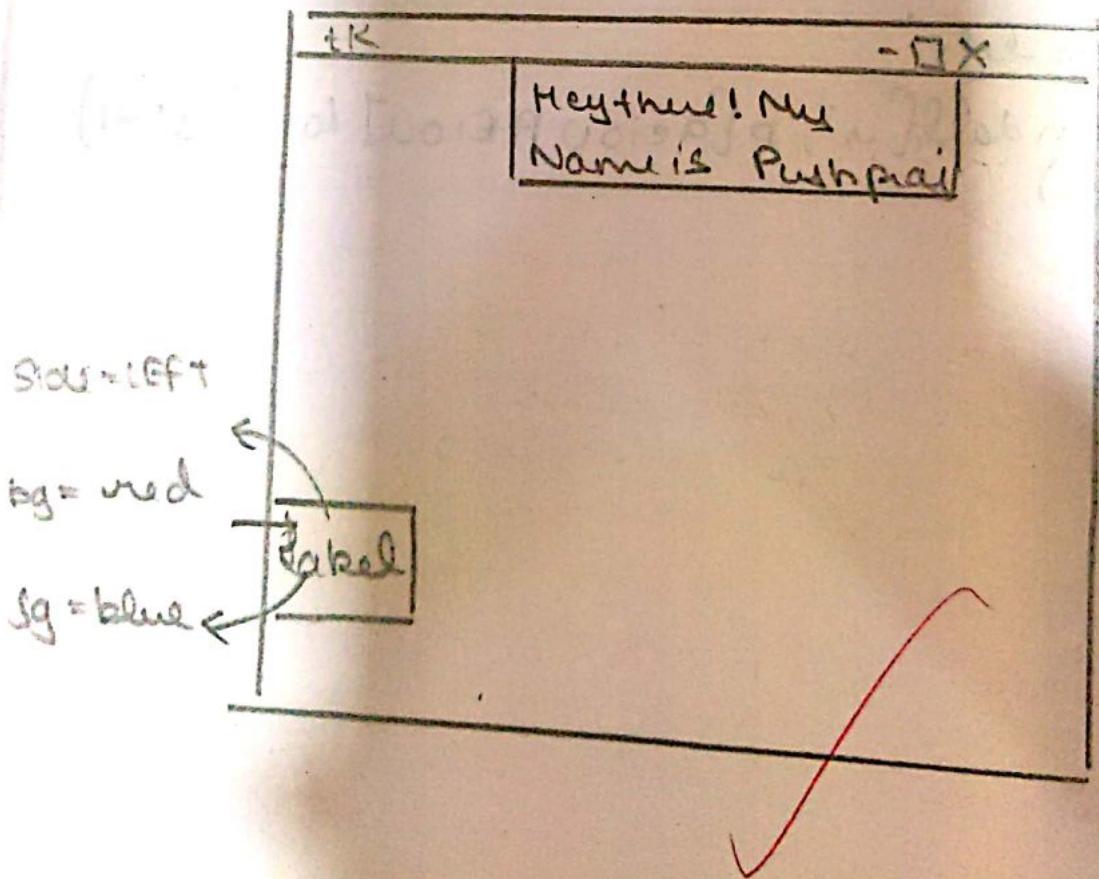
- Now repeat above step with a label method which takes the following argument.

(i) Name of Parent window
 (ii) The text attribute which defines the string

PROGRAM:-

```
from tkinter import *
root = Tk()
T1 = Text(root)
T1.insert(END, "Hey There! My name is Pushpraj .")
T1.pack(side=TOP, padx=20, pady=30, ipadx=40, ipady=20)
L1 = Label(root, text="Label", bg="red", fg="blue")
L1.pack(side=LEFT, padx=10, ipadx=20, ipady=30)
root.mainloop()
```

OUTPUT:



(iii) The background colour (bg)

(iv) The foreground colour (fg)

Now use pack() with relevant attributes.

(ii)

Aim: To make use of Radio button widget for selection of one of the multipli's option.

Algorithm:

Step.1: Use the tkinter method to import the relevant method.

Step.2: Define a function which tells user about given selection mode from multiple option available

Step.3: Use the config method along with label method of and call the variable as an argument within method.

Step.4: Now define the parent window and define option using config variable.

Step.5: Now create object of Radiobutton which will take following arguments:

(i) Positioning on Parent window

(ii) Text Variable

(iii) Define variable argument

(iv) Corresponding value and trigger the given function

PROGRAM:

```
from tkinter import *
def sel 1():
    selection="Pushpraj"
    label.config(text=selection)
def sel 2():
    selection="Sushant"
    label.config(text=selection)
def sel 3():
    selection = "Mahesh"
    label.config(text=selection)
def sel 4():
    selection = "Bhavya"
    label.config(text=selection)
root=TK()
var=IntVar()
L1=Label(root, text="Select any roll number")
L1.pack(side=TOP)
R1=Radiobutton(root, text="1744", variable=var, value=0, command=sel1)
R1.pack(anchor=N)
R2=Radiobutton(root, text="1764", variable=var, value=1, command=sel2)
R2.pack(anchor=N)
R3=Radiobutton(root, text="1766", variable=var, value=2, command=sel3)
R3.pack(anchor=N)
R4=Radiobutton(root, text="1775", variable=var, value=3, command=sel4)
R4.pack(anchor=N)
label=Label(root)
label.pack(side=BOTTOM)
root.mainloop()
```

Step 6:

- Now call the pack method for corresponding Radio object so created and specify argument as an anchor attribute.

Step 7:

- Now define a label object and place it onto parent window using pack method and finally use mainloop method.

OUTPUT:- 68

LR	- □ X
Select any roll Nos	
• 1474	
• 1475	
• 1464	
• 1466	

Pushpraj

(iii) Aim: To make use of scroll Bar widget of the GUI application.

Algorithm:

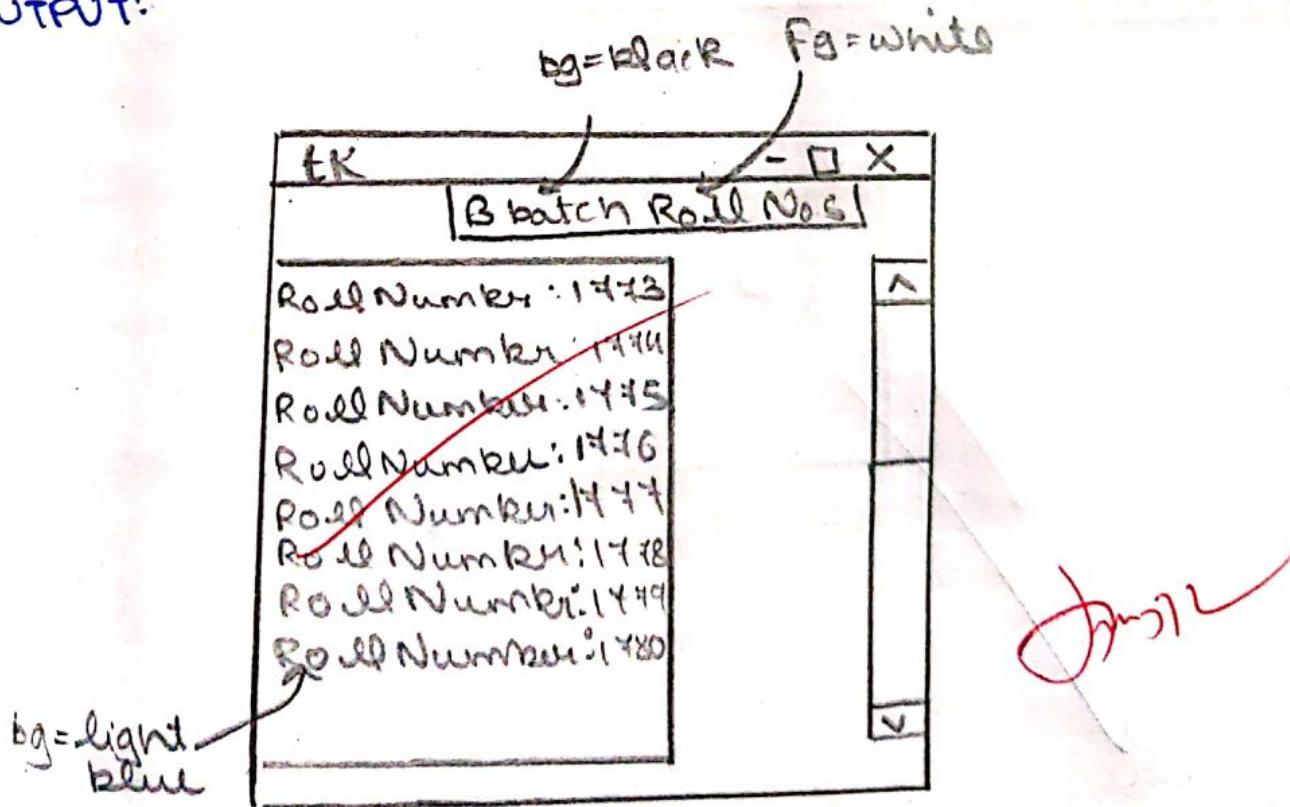
- Step 1: Import tkinter library to use Scroll Bar widget.
- Step 2: Create an object corresponding to scroll parent window and create an object from scroll bar and place it on to the parent window so created.
- Step 3: Create an object of label method to provide a heading and place it on parent window.
- Step 4: Use pack method along with object of scroll bar method and use argument side and fill.
- Step 5: Create an object of listbox method and place it onto parent window with attribute command.
- Step 6: Use for loop to insert values in the object of list box by using insert method.
- Step 7: Use config method along with scroll bar object and use command attribute.
- Step 8: Finally call the mainloop method.

PROGRAM:-

36

```
from tkinter import *
root = Tk()
root.geometry('450x400')
l=Label(root, text="B Batch roll numbers: "bg"=black, fg="white")
l.pack()
scroll=Scrollbar(root)
scroll.pack(side=RIGHT, fill=Y)
mylist=Listbox(root, yscrollcommand=scroll.set, bg="light blue")
for num in range(1, 21):
    mylist.insert(END, "Roll number : " + str(num))
mylist.pack(side=LEFT, fill=BOTH)
scroll.config(command=mylist.yview)
root.mainloop()
```

OUTPUT:



(17) Message Box Method:

Step-1:

- Import the relevant method from tkinter library.

Step-2:

- Define a function and use the message box along with different methods available which contains one or more arguments.

Step-3:

- Thus different options which are available are showinfo(), showwarning(), askyesno(), askquestion().

Step-4:

- Create object from button method and place it onto parent window with title of button specified.

Step-5:

- Use the pack method to display button widget and finally use mainloop method.

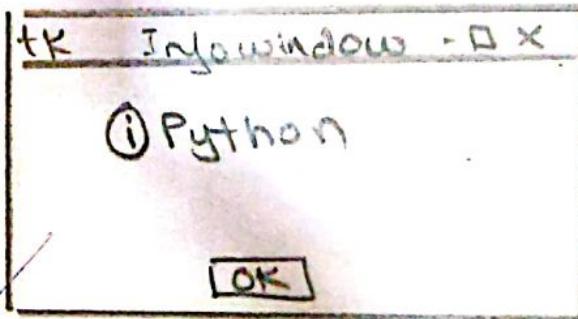
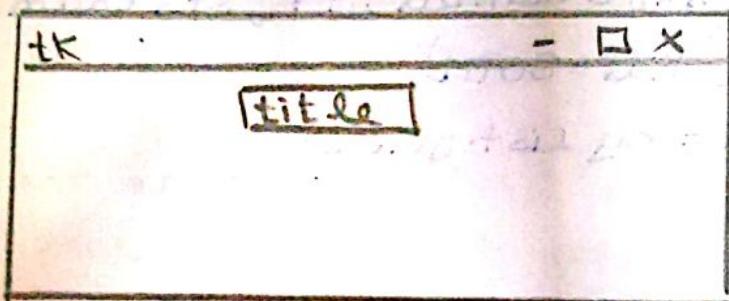
Step-6:

- If the user wants to hide the parent window and only the info window should be visible corresponding to the six options given below

Message Box:

```
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:



(V) Relief Style:

Step-1: Use the button with the following attribute.

- (i) The parent window.
- (ii) Text attribute
- (iii) Relief.

Step-2: Use the corresponding pack method for the respective button objects and trigger the corresponding event.

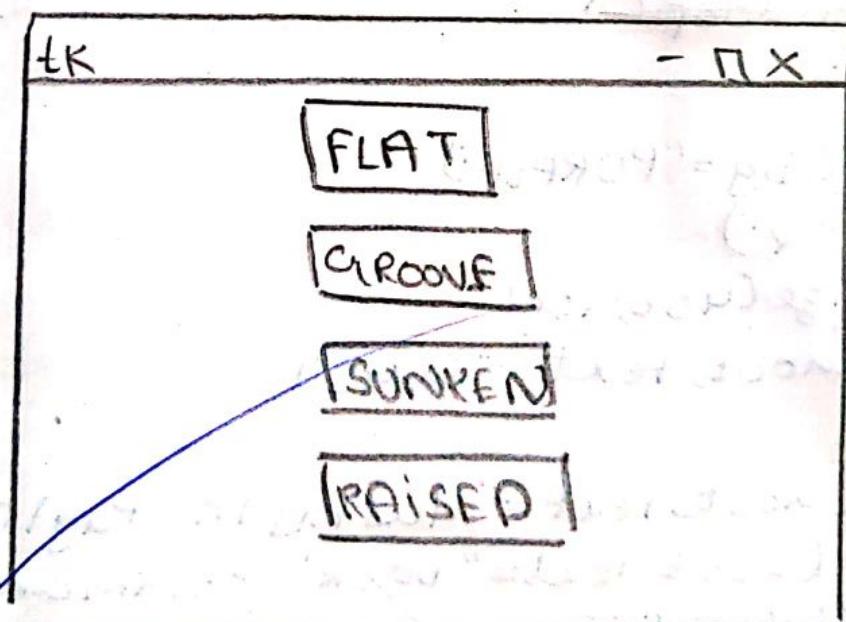
Step-3: finally use the mainloop event.

Source Code:

```

from Tkinter import *
root = Tk()
b1 = Button(root, text = "FLAT", relief = FLAT)
b1.pack()
b2 = Button(root, text = "GROOVE", relief = GROOVE)
b2.pack()
b3 = Button(root, text = "SUNKEN", relief = SUNKEN)
b3.pack()
b4 = Button(root, text = "RAISED", relief = RAISED)
b4.pack()
root.mainloop()

```

OUTPUT:

* Traversing and making use of geometry layout manager method.

Step.1:

→ Define a function and create a object of the given window by using the three methods namely config, title, minsize.

Step.2:

→ Create a button object and use the text and command attribute for triggering the given event and use grid method along with internal and external padding. Hence, button object will allow to terminate.

Step.3:

→ Define second function corresponding to second window with attributes config, title, minsize for the window object and shift the focus into third window.

Step.4:

→ 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.

Source Code: [Traversing]

```
from tkinter import *
root = TK()
def main():
    root = TK()
    root.config(bg="PINK")
    root.title("main")
    root.minsize(200, 200)
    L = Label(root, text="AIS")
    L.pack()
    L1 = Label(root, text="Persian cat\n-British  
short hair\n-Munchkin cat\n")
    L1.pack()
    b1 = Button(root, text="Second", command=sel)
    b1.pack(side=RIGHT)
    b2 = Button(root, text="terminate", command=te)
    b2.pack(side=BOTTOM)
    root.mainloop()
def sel():
    root = TK()
    root.config(bg="PURPLE")
    root.title("2")
    root.minsize(400, 200)
    L2 = Label(root, text="DOGS")
    L2.pack()
    L3 = Label(root, text="Husky\n-Pug\n-Labrador")
    b2 = Button(root, text="back", command=back)
    b2.pack(side=LEFT)
    b3 = Button(root, text="terminate", command=te)
    b3.pack(side=BOTTOM)
    root.mainloop()
```

P.E

Step 5:

→ Define a function for termination and call the quit method and finally call the first function created and trigger mainloop method.

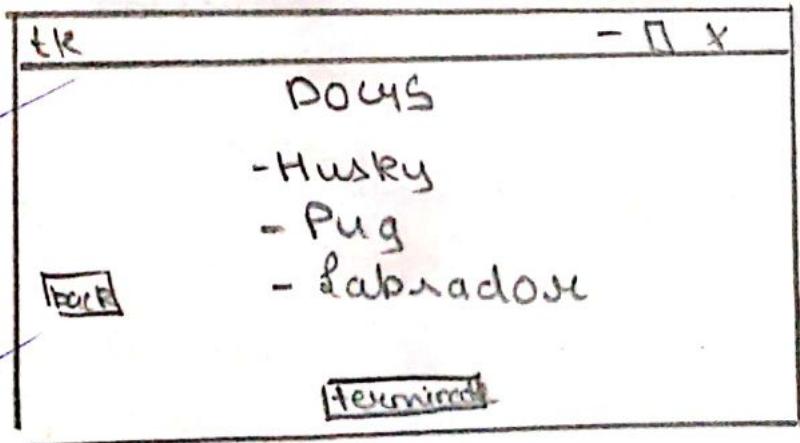
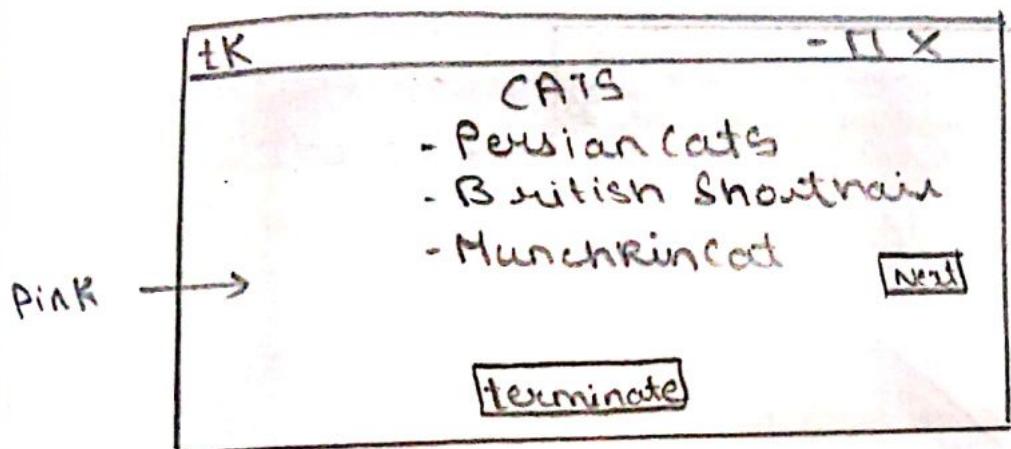
def ter1():
 quit()

bu=Button(root,text="PET CATS AND DOGS")
bu['command']=main

bu.pack()
root.mainloop()

40

OUTPUT:



* Spinbox:

Step-1:

→ Create an object from the `tk` method and subsequently create an object from the `Spinbox` method

Step-2:

→ Make the object so created onto the parent window and trigger the corresponding events.

Step-3:

→ Use the ~~anchor~~ `pack` method to provide the direction using `anchor` method.

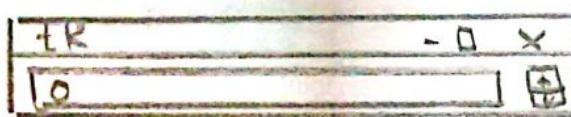
Step-4:

→ Use the `mainloop` method to terminate.

Source Code:-

```
from tkinter import *
root = Tk()
s1 = Spinbox(root, from_=0, to_=10)
s1.pack(anchor=s)
root.mainloop()
```

OUTPUT:-



* Paned Window:

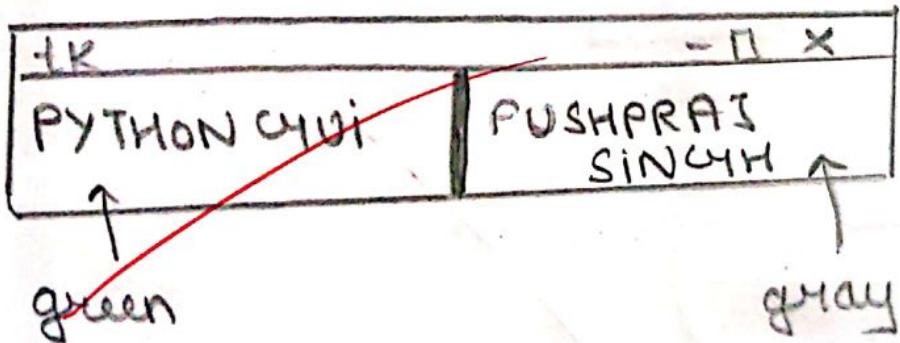
- Step 1: Create an object from paned window and use the pack method with the attribute fill and expand.
- Step 2: Create an object from the label method and put it onto the paned window with the text attribute and use the add method to embed the new object.
- Step 3: Similarly create a second paned window object and add it onto the 1st paned window with orientation specified.
- Step 4: Now Create another label object and place it onto the 2nd paned window object and add it onto the 2nd pane.
- Step 5: Now use the mainloop method to terminate.

Source Code:-

42

```
from tkinter import *
root=TK()
p=PanedWindow(bg="red")
p.pack(fill=BOTH, expand=1)
l1=Label(p, text="PYTHON GUI", bg="green")
p.add(l1)
p1=PanedWindow(p, orient=VERTICAL, bg="blue")
p.add(p1)
l2=Label(p1, text="PUSHPRAJ SINH", bg="gray")
p1.add(l2)
root.mainloop()
```

OUTPUT:



* Canvas Widget:

Step 1:

- Use the tkinter method and create an object from the canvas method and use the attribute height, weight, bg colour and the parent window object.

Step 2:

- Use the method create oval, create line and create arc along with the canvas object so created and use the co-ordinate value. Also use the fill attribute to assign various colours.

Step 3:

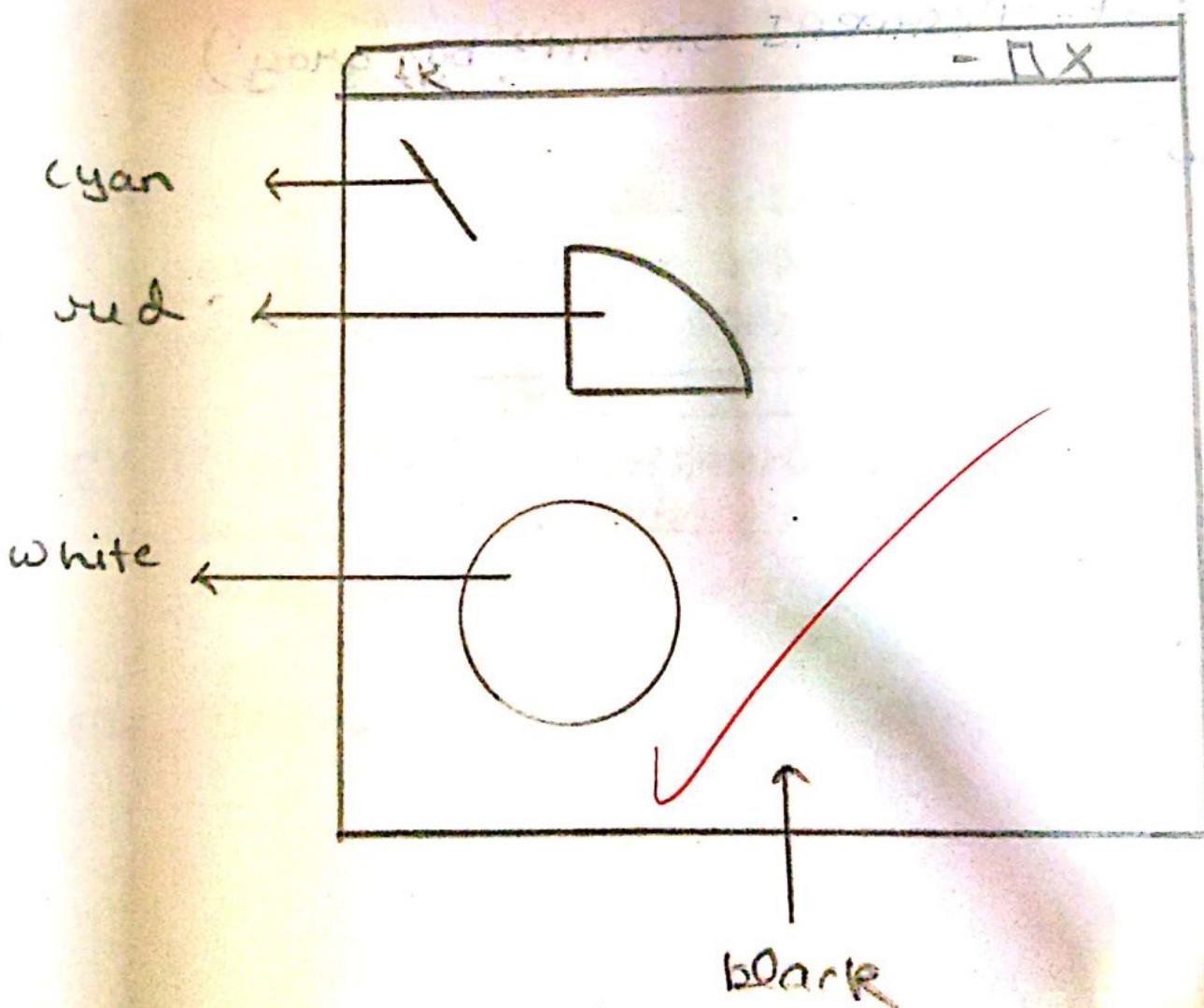
- Now call the pack method and mainloop method.

Jn 17/12.

Source Code:

```
from tkinter import *
root = Tk()
c1 = Canvas(root, height=400, width=400, bg="black")
oval = c1.create_oval(20, 140, 150, 250, fill="white")
line = c1.create_oval(30, 40, 50, 60, fill="cyan")
arc = c1.create_oval(20, 140, 150, 60, fill="red")
c1.pack()
root.mainloop()
```

OUTPUT:



Practical: 6

Step.1:

- Import db library and use the open method for creating the database by specifying name of the database along with the corresponding regular.

Step.2:

- Use the objects for accessing to given web size and the corresponding regular for the web size.

Step.3:

- Check whether the given URL address with the regular of the pages is not equal to None then display message from URL address else not found.

Source Code:-

```
import dbm  
db=dbm.open("database", flag="c")  
if db["www"]!=None:  
    print("good")  
else:  
    print("Not good")
```

OUTPUT:

good.

b]

Step 1:

→ Import the corresponding library taking of database connection.

Step 2:

→ Now create connection objects using sqlite library and connecting method to create the new database.

Step 3:

→ Now create the cursor objects using cursor method from the connection objects create in steps.

Step 4:

→ Now use the executing method for creating the table with the column name and respective datatype.

Step 5:

→ Now with the cursor objects we insert elements for entering the values co-ordinating into the different field considering the datatypes.

Step 6:

→ Use the commit method to complete the transaction use the connection objects.

Program:-

```
import os, Sqlite3  
connection = Sqlite3.connect("student.db")  
c1 = connection.cursor()  
c1.execute('Create table student (Name, RNo, DOB)')  
c1.execute('insert into student values ("Rakesh", 1840,  
23-06-2002)')  
c1.execute('insert into student values ("Monu", 1774, 05-10-2001)')  
c1.execute('insert into student values ("Sawan", 1790, 10-01-2000)')  
connection.commit()  
c1.execute('Select * from student')  
c1.fetchall()  
c1.execute('Drop table Student')
```

OUTPUT:-

```
[('Rakesh', 1840, 23-06-2002), ('Monu', 1774, 05-10-2001),  
('Sawan', 1790, 10-01-2000)].
```

Step 7:

→ Use the execute statement along with the cursor object for inserting the value the data base using selecting from where clause.

Step 8:

→ finally use the fetchall method for display the value for the table using the cursor objects.

Step 9:

→ Use the execute method and the drop table syntax for terminating the database finally use the close method.