

18

Practical 2 Different file according
demonstrates the use of different file objects and some methods.

Algorithm

Step 1: Create a file object using open method
and use the writing class mode followed
by writing done, consider onto the file
and then closing the file.

Step 2: Now, open the file in read mode and
then use read(), reading and reading()
and store the output in variable and
finally displaying the contents of variable.

Steps 3: Now, will the file object for finding
the name of the file, the file name, the
file mode in which it's opened whether
the file is still open or close and finally
the output of the is附录 attribute

write method

```
fileobj = open("abc.txt", "w")
fileobj.write("Subjects in " + "n(s)")
fileobj.write("\n" + "\n Calculus Instas In Python")
fileobj.close()
```

read method

```
fileobj = open("abc.txt", "r")
str1 = fileobj.read()
print("The output of read method is : ", str1)
fileobj.close()
```

>>> The output of read method is : Subjects in(s

Calculus

Instas

In Python

readline method

```
fileobj = open("abc.txt", "r")
str2 = fileobj.readline()
print("The output of readline method is : ", str2)
```

```
fileobj.close()
>>> The output of readline method is : Subjects in(s
```

readlines method

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

>>> The output of readlines method is : Subjects in(s
In. CIn : Calculus\n! Instas\n! Python

Ex. # file attributes

```
a = fileobj.name  
b = fileobj.closed  
c = fileobj.mode  
  
d = fileobj.softspace  
print("Name of file ::",a)  
print("Closed ::",b)  
print("File mode ::",c)  
print("Softspace ::",d)  
  
''' Name of file : abc.txt and two lines  
''' Closed : True  
''' File mode : r  
''' Softspace : 0  
  
# write mode  
  
fileobj = open("abc.txt","w")  
fileobj.write("C")  
fileobj.close()  
  
#. read mode  
fileobj = open("abc.txt","r")  
x = fileobj.read()  
print("The output of read mode is ::",x)  
  
# w+ mode  
fileobj = open("abc.txt","w+")  
fileobj.write("Skewin")  
fileobj.close()
```

Step 4: Now open the fileobj in write mode
write some contents close, subsequently then
again open the fileobj in 'wt' mode that
is the update mode and write contents

Steps : open fileobj in read mode display the
whole written contents and close
open again in 'rt' mode that is with
parameter passed and display the output
Subsequent

Step 5: Now open file in append mode open
write method, write contents close
the fileobj again Open the fileobj in read
mode and display the appended output

Step 7:- Open the fileobj read mode declare a variable and perform file objects dot tell method and store the output consequently in variable

Step 8:- Use seek method with the arguments with opening the file obj in read mode and closing it subsequently.

Steps:- Open fileobj with read mode also use the readlines() method and print the same. For reading the length use for conditional statement and displaying the length

```

# r+
fileobj = open ("abc.txt", "r+")
r1 = fileobj.read(10)
print ("The output of r+ is", r1)
>>> The output of r+ is: division by zero
# append method
fileobj = open ("abc.txt", "a")
fileobj.write ("In FOSS")
fileobj.close()
fileobj = open ("abc.txt", "r")
s1 = fileobj.read()
print ("The output of append mode is", s1)
fileobj.close()
>>> Tell method
fileobj = open ("abc.txt", "r+")
pos = fileobj.tell()
print ("Tell", pos)
fileobj.close()
>>> Tell 0

```

~~Java
Django~~

Program

```
rustyke = ("Bianka", "Abhinav", "Malikom", "Vedant")
nijiten = iden ("rustyke")
print (nijiten)
print (nijiten)
print (nijiten)
print (nijiten)
print (nijiten)
print (nijiten)
```

Output:

```
Bianka
Abhinav
Malikom
Vedant
```

Program:

```
rustyke = ("Bianka", "Abhinav", "Malikom", "Vedant")
for a in rustyke:
    print (a)
```

Output:

```
Bianka
Abhinav
Malikom
Vedant
```

Practical 2

27.

Aim :- To display elements of a tuple using iteration method.

Algorithm :-

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

Step 2:- Use ~~for loop~~ with tuple and assign it to a variable.

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

Aim :- To use item 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 element of tuple

15

QUESTION 15. Define a class named Odd.

Define a class within it that defines display method which will underline the first character within the container object.

Now we need method g Define the logic for displaying the odd values.

Program

Class obj

def __init__(self):

self.num=1

return self

def __next__(self):

num, self += 1

return num

def __next__(self):

num = self * num

self, num += 1

return num

myobj=obj()

myobj = obj() # created object

num = int(input("Enter the no:"))

for i in myobj:

if i < num:

print(i)

Output :-
Enter the no 5

1
2
3

(A) Program 85

try

f = open ("Kushal.txt", "w")
f.write ("My name is Kushal")

except IOError:

print ("Environment error")

else

print ("Successful")

Output: Successful

b)

Program

try

a = int (input ("Enter"))

except ValueError:

print ("Arithmatic error")

else

print ("Successful")

Output

Enter: 14

Arithmatic error

Practical 9Exception Handling

- 1) Algorithm
use the "try block" to defined the normal block of action.
- 2) Example:- Define a file object and open the file in the write and some content ends the file.
- 3) Use an example block with to errors as an environment error and convey the appropriate message to the user. else display the message that the exception or operation is carried out successfully

~~Incomplete
Topic~~

48

Program to demonstrate the multiple exception
I/O error and value exception.

S1:- Use the try block and define the file object and open the file in write mode or read mode and write some content onto the file.

S2:- Also i accept the value from the user and if it is a valid value display the entered value and terminate the condition.

Practical -4

Topic:- Regular expression

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

Step3:- Import re module declare pattern with meta character use the split() and print the output.

Step4:- Import re module declare string and accordingly declare pattern replace the blank Space with no-Space use sub() with 3 arguments and print the String without Spaces

Step5:- Import re module declare a Sequence use search method for finding subsequently use the group() with dot operator as search() gives memory creation using group() it will show up the matched string.

Ex.

Steps:- Import re module declare list with number
we use the conditional statement here we have
used up the for conditional statement
here we have used up the for
condition Statement

re-space :

import re

string = 'abc def ghi'

pattern = r'[]+'

replace = ""

U1 = re.sub(pattern, replace, string)

print(U1)

>>> abcdefghi

group()

import re

sequence = "I python is an interesting language"

U = re.search("Apython", sequence)

print(U)

U1, U.group()

print(U1)

>>> <_re.Match object at 0x0281 DF00>

By Python:

Verifying the given set of phone numbers import re

list1 = ['8004567891', '9145673210', '1234567890',
'9876543210']

for values in list1:

if re.match(r'[8-9]\d{1}\d{9}'): {0-9} d9}:

value or en(value) == (0):
value or en(value) == (0):
value or en(value) == (0):
value or en(value) == (0):

print("Criteria matched for cell number!")

else:

print("Criteria failed")

77) criteria matched for cell number
88) criteria matched for cell number
criteria failed!
Criteria matched for cell number

vessels

import re

str1 = "plant is life overall"

```
output = re.findall(r'1 Is [a-zA-Z0-9]+', str1)
```

print (output)

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

Just 8 domain

airport re

seq = 1abc. fcsc @ edu. com x 42 @ g mail. com

pattern = r' [\w{.} -] + [' \w{.} -] +

Output = re.findall(pattern, seq)
[begin, ..., end]

pattern
abc.tcsdc@edu.com xyz@gmail.com

counting of first 2 letters.
in box

import 're'

$$S = \{ \text{nor. } a, \text{ns. } b, \text{ns. } c, \text{nor. } d \}$$

$$P = \pi^* [(\ln s) / \ln \eta] +$$

Ore. finfall (b/s)

print(0)

$$f'' = 0$$

1

for $n \in \mathbb{N}$:

Step 7: Import re module declare a string use the module with.findall() for finding the vowels in the string and declare the same.

Step 8: Import re module declare the host and domain name declare pattern for separating the host & domain name use the.findall() and print the output respectively.

Step 9: Import re module enter a string use pattern to display only two elements of the particular string use.findall() declare two variables with initial value as zero use for condition and subsequently use the if condition.



if ($j == \text{ms}_1$):

$f = f + 1$

else

$n = n + 1$

print ("No of males is : " + str(n))

print ("No of females is : " + str(f))

print ("No of males is : " + str(m))

print ("No of females is : " + str(t))

print ("No of females is : " + str(r))

Ans

Q8

PROGRAM:

```
from tkinter import *
```

```
root = Tk()
```

```
T1 = Text(root)
```

```
T1.insert(END, "Hey there! my name is Akash")
```

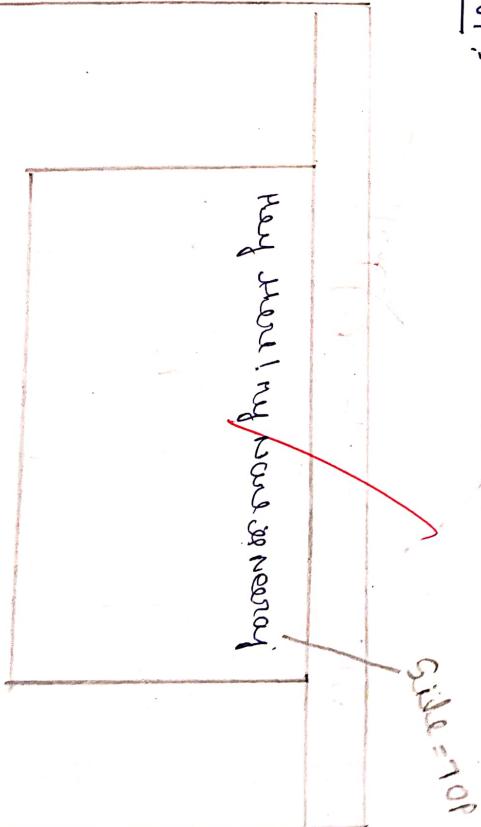
```
T1.pack(side=TOP, padx=20, pady=30, ipadx=40, ipady=40)
```

```
L1 = Label(root, text="Label", bg="red", fg="blue")
```

```
L1.pack(side=LEFT, padx=10, ipadx=20, ipady=10)
```

```
root.mainloop()
```

OUTPUT:-



Practicals

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

Algorithm:

Step 1) Use the Skinter library for importing the features of text widget.

Step 2) Create a variable from a text variable & position it onto the parent window

Step 3) Use the pack() along with the object created from text method & use the parameter
1) 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 arguments.

- Name of parent window
- The background colour
- The foreground colour
- Now use pack() with relevant attributes

Ques:- To make use of Radiobutton widgets
Selection of one of the option.

Algorithm:

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

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

Step 3:- Use the config method along with self method to call the variable as an argument.

Step 4:-

Now define the parent win and define option using control variable

Step 5:-

Now create object of Radio B button
 1) set variable take following ways
 2) Define variable argument

Step 6:- Now call the pack method for corresponding arguments as an anchor attribute.

Step 7:- Now define a label object and place it onto parent window using pack method.

PROGRAM:

36

```
from tkinter import *
def sel():
    selection = "No Kursus"
    label.config(text=selection)
def sel2():
    selection = "Ei mark"
    label.config(text=selection)
label = Label(config(text="Selection"))
label.pack()
sel()
sel2()
selection = "Abhinav"
label.config(text=selection)
label = Label(config(text="Selection"))
label.pack()
selection = "Ganesh"
label.config(text=selection)
label = Label(config(text="Selection"))
label.pack()
root = Tk()
root.title("Python GUI")
L1 = Label(root, text="Select and number")
L1.pack(side="top")
L1 = pack(side="top")
R1 = Radiobutton(root, text="1963", variable=var,
                 value=0, command=sel)
R1 = pack(side="left")
R1 = pack(side="right")
R2 = Radiobutton(root, text="1967", variable=var,
                 value=1, command=sel2)
R2 = pack(side="left")
R3 = Radiobutton(root, text="1971", variable=var,
                 value=2, command=sel3)
R3 = pack(side="left")
R4 = Radiobutton(root, text="1975", variable=var,
                 value=3, command=sel4)
R4 = pack(side="left")
```

Output

4A

- X

Select any three numbers

01763

01762

01772

01773

01774

01775

01776

01777

01778

Kusnud

make a note who of certain buttons works for selection
of one of the options.

Method:

Method will use standard method to support the
forward method.

Method defining a function which takes user chosen
given suggestion made from suitable option

method for the marking method along with label
method. Using the variable for arguments
within method

label will now define the parent window & defining
option using central variable.

New create object of Radio Button which
will take following arguments

pushing on parent window

and variable variable.

Defining variable argument
for given border value and triggering the
given function.

Method now call the back method for corresponding
radio object to update and specify
arguments on an another attribute.

Method will define a Radio object and places on
the parent window using back method
accordingly we can do it.

Aim: To make use of scroll bar widget of my application

Algorithm:

Step 1: Import tkinter library to use scroll bar widget

Step 2: Create an object corresponding to scroll parent window & create an object for scroll bar & place it on the parent window created.

Step 3:

Create an object of `Label` method to have a heading and place it on parent window pack method along with object of scroll bar method as the argument of `fill`

Step 4: Create an object of `List Box` method & place it onto parent window method an attribute `yscroll` window with use for scroll bar command.

Object of `List Box` insert values in the window by using `insert` method

Step 5: ~~Bind config method along with scroll bar object of use command attribute the main loop method~~

Program

38

from skidder import
tk

root = tk()

root.geometry("450x400")

l = Label(root, text = "Batch Reel numbers:", bg="black", fg="white")

l.pack()

scroll = Scrollbar(root)

scroll.pack(side = RIGHT, fill = Y)

listbox = Listbox(root, yscrollcommand = scroll.set, bg = "light blue")
for num in range(1, 10):

my_list.insert(END, "# Reel numbers: " + str(num))

mylist = Listbox(root, side = LEFT, fill = BOTH)

scroll.config(command = mylist.yview)

root.mainloop()

def exit

+K	B Batch Reel Numbers	-	X
✓			

* Reel Number: 1761
 * Reel Number: 1762
 * Reel Number: 1763
 * Reel Number: 1764
 * Reel Number: 1765
 * Reel Number: 1766
 * Reel Number: 1767
 * Reel Number: 1768
 * Reel Number: 1769

Airplane Components

~~Step 2:- Import relevant methods from Heintzen library~~

~~Step 2:- Define the object corresponding to parent window and define the size of parent window in terms of no of pixels~~

~~Step 3:- Now define the frame object from the method and place it on the parent window~~

~~Step 4:- Create another frame object derived os the left frame and put it on the parent window on its LEFT side.~~

~~Step 5:- Similarly define the RIGHT frame and subsequently define the button object placed onto the given frame with the attribute as text, active background and foreground.~~

~~Step 6:- Now use the pack() along with the side attribute~~

~~Step 7:- Similarly create the bottom object corresponding to the MODIFY operation but it will frame object on side = right~~

4.8

Steps:- Create another button object & place it
on to the Right hand & label the
button as ADD.

Step 9:- Add another button object & place it on
the top of frame and label it as ERASE.
Step 10:- Use the pack() simultaneously for all
the objects & finally save the main
file as my first GUI program.

Code:-

```
from Tkinter import *
root = Tk()
root.title("My First GUI Program")
root.geometry("350x350")
label = Label(root, text="Hello World").grid(row=0, column=0)
button = Button(root, text="Add").grid(row=1, column=0)
button = Button(root, text="Delete").grid(row=1, column=1)
button = Button(root, text="Exit").grid(row=2, column=0)
root.mainloop()
```

from Skripten auf Seite 4

window = TK()

window.geometry("600x500")

frame = Frame (window)

frame.pack ()

left frame = Frame (window)

left frame . pack (side: "left")

right frame = Frame (window)

right frame . pack (side: "right")

b1 = button (frame , text = "Submit" , active background

= "red" , fg = "blue")

b2 = button (frame , text = "cancel" , active background

= "yellow" , fg = "black")

b3 = button (frame , text = "ADD" , active background

= "blue" , fg = "red")

b4 = button (frame , text = "EXIT" , active background

= "red" , fg = "green")

b1 . pack (side: "LEFT" , padx = 20)

b2 . pack (side: "RIGHT" , padx = 50)

b3 . pack (side: "Bottom" , pady = 20)

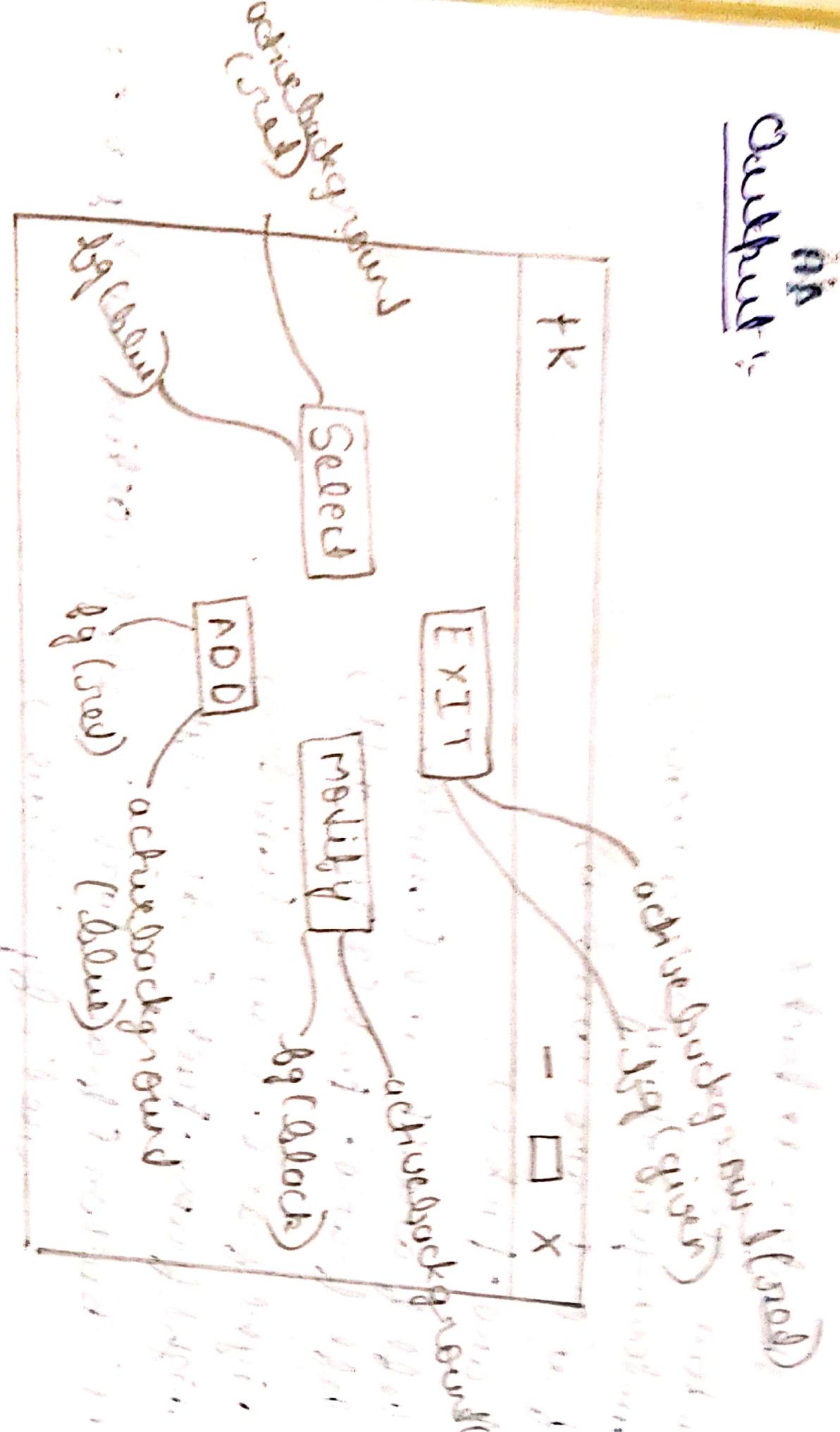
~~b4 . pack (side: "top")~~

~~b4 . pack (side: "top")~~

Dr. 10

Outlines

3



Practicals

Aim:- To make use of message box method of the
GUI application

Algorithm

Step 1:- Import relevant methods from tkinter library

Step 2:- Define a function and use message box
along with different methods available
which contains one or more arguments

Step 3:- Create an object from button method
and place it onto the parent window
with text and command attribute specified

Step 4:- Use pack method and finally use the
~~mainloop~~ method.

from Tkinter import *

from Tkinter import messagebox

def neighbor():

message box. Show error ("Action", "Do you want to

message box. Show error ("Error", "Can't load the
continued?")

process!"

root = Tk()

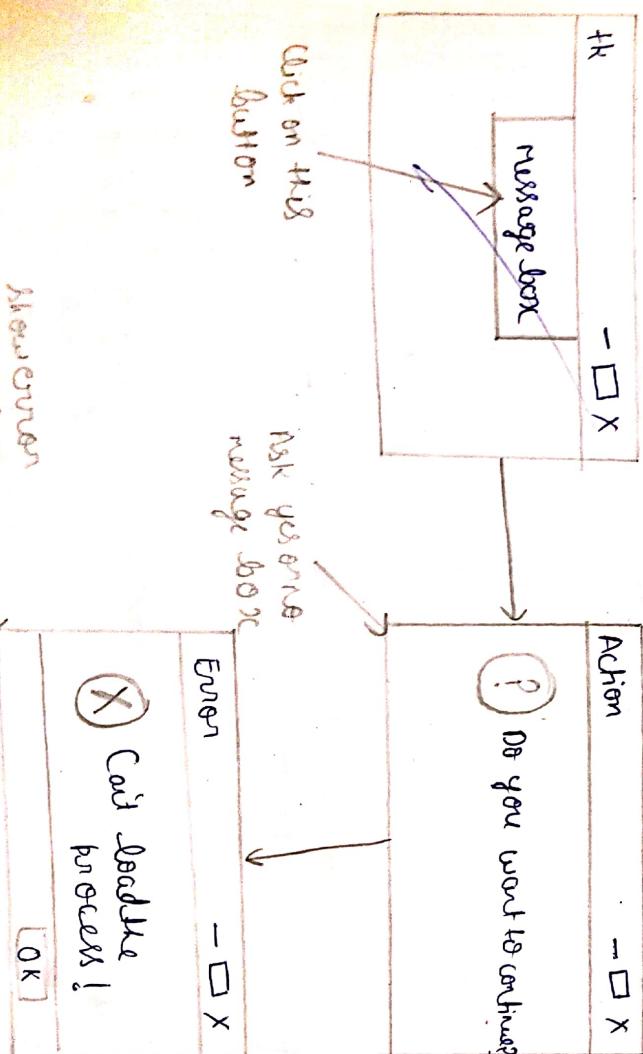
root . config (bg = "green")

b1 = button (root , text = "message box", bg = "red", command = error)

b2 = pack ()

root . mainloop ()

OUTPUT :-



Aim: To program to traverse various windows using button's window gif.

Algorithm:

Step 1: Import the relevant method from tkinter library.

Step 2:-

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

Step 3:-

Define a button object which will be placed on the current window to traverse and define another button which will be used to exit from the window

Step 4:- Define another function which will use the quit method to terminate the program.

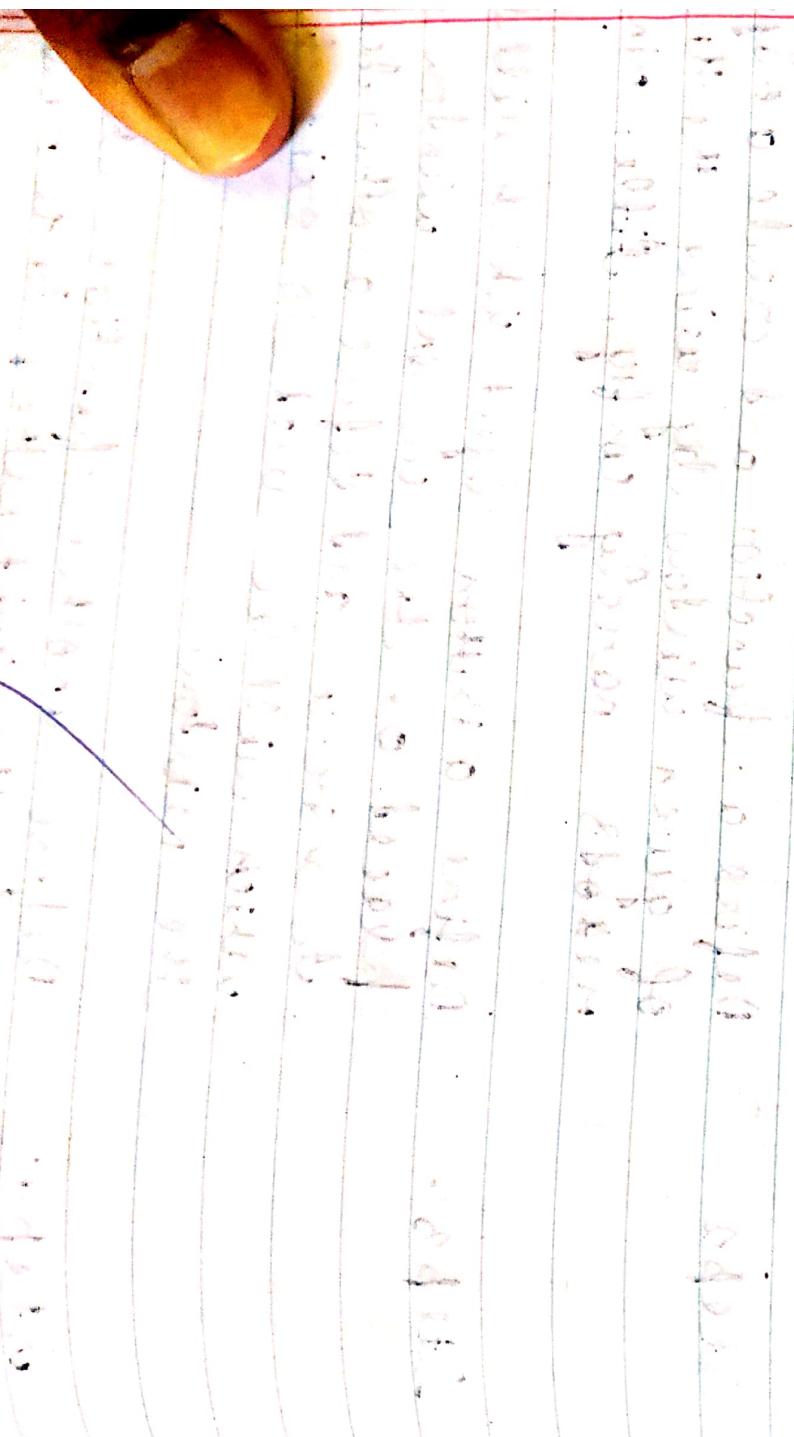
Step 5:- Now create an object of main window and use various methods like config title, geometry etc.

Step 6:- Define two buttons which will be placed on the main window; one to traverse another window and the other to terminate the programs.

Ex

Step 7:- Define another function on which will carry various buttons placed on this window. Define two buttons respect and use the grid method along with the two buttons.

Step 8:- Finally call the mainloop method.



44

Coding
 from scratch import *
 from main C:
 need = tk()
 need · geometry ("400x500")
 need · config (bg = "light green")
 need · title ("window 1")
 B_1 = button (need, text = "Next", command = main)
 B_2 = button (need, ipadx = 50, ipady = 40, pack = 10, pady = 50)
 B_2 = button (need, text = "Exit", command = return)
 B_2 = button (need, ipadx = 50, ipady = 40, pack = 20, pady = 50)
 del need()

quit() = Tk()

TOS = geometry ("450x500")

TOS · config (bg = "purple")

TOS · title ("rain window")

B_3 = button (TOS, text = "Continue", command = main)

B_3 = grid (ipadx = 50, ipady = 40, padx = 20, pady = 30)

B_2 = button (TOS, text = "Exit", command = return)

B_2 = grid (ipadx = 50, ipady = 40, padx = 20, pady = 30)

def rain (C):

 C = 0

 top · geometry ("490x500")

 top · config (bg : "purple")

 top · title = ("window 2")

b_4 = button (top, text = "main page", command = rain)

b_4 · grid (column x=90, ipady = 40, padx x=20, had y=30)

b_2 = button (top, text = "exit", command = term)

b_2 · grid (ipadx=50, ipady = 40, padx x=20, pady=30)

mainloop (C)

output

Main window - □ X

continue

exit

Takes focus from window 2

Window 2 - □ X

next

exit

dg = purple

Takes focus
back to first
window

window 2 - □ X

main page

37.

Source code:

from tkinter import *
root = Tk()

P = Toplevel window (bgc = "red")

P.pack (fill : BOTH, "PYTHON GUI"), bgc = "green")

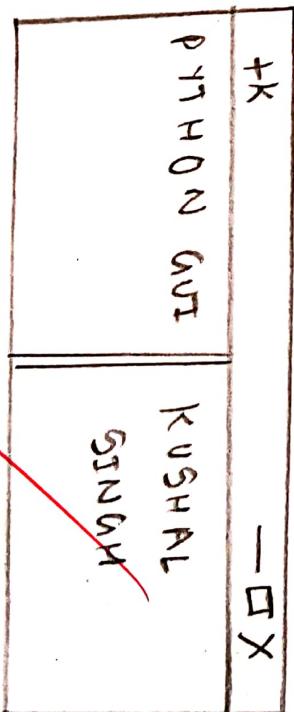
P.add (L)

L1 = Toplevel window (P, orient=VERTICAL , bgc = "blue")

P.add (P1)

L2 = Label (P1, text = " KUSHAL SINGH ", bgc = "gray")
P1 . add (L2)
root . mainloop ()

Output:



Paned window

- *
step) Create an object from paned window and use the pack method with the attributes fill and expand.
- *
step) Create an object from the label method and put it onto the paned window with the text attribute and use the add method to add the new object to the paned window.
~~similarly create a second paned window object and add it on to the 1st paned window with orientation specified~~
- *
~~now create another label object and place it onto the 2nd paned window object and add the onto the 2nd part.~~
- *
~~now use the mainloop method to terminate.~~

* canvas widget

Step 1
→ use the tkinker method and create an object from the canvas method and use the attributes high weight , say below and the parent window object.

Step 2
use the method create oval , create arc and create arc along with the canvas object so created and use the co-ordinate values. Now use the fill attribute to assign various colors.

Step 3:

Now call the pack method and mainloop method

source code

from "sketchbook"

root = KC()

l1 = canvas (x:0, y:0, width:400, height:400, fill:"white")

l1.create - oval (20,140, 150, 250, fill:"white")

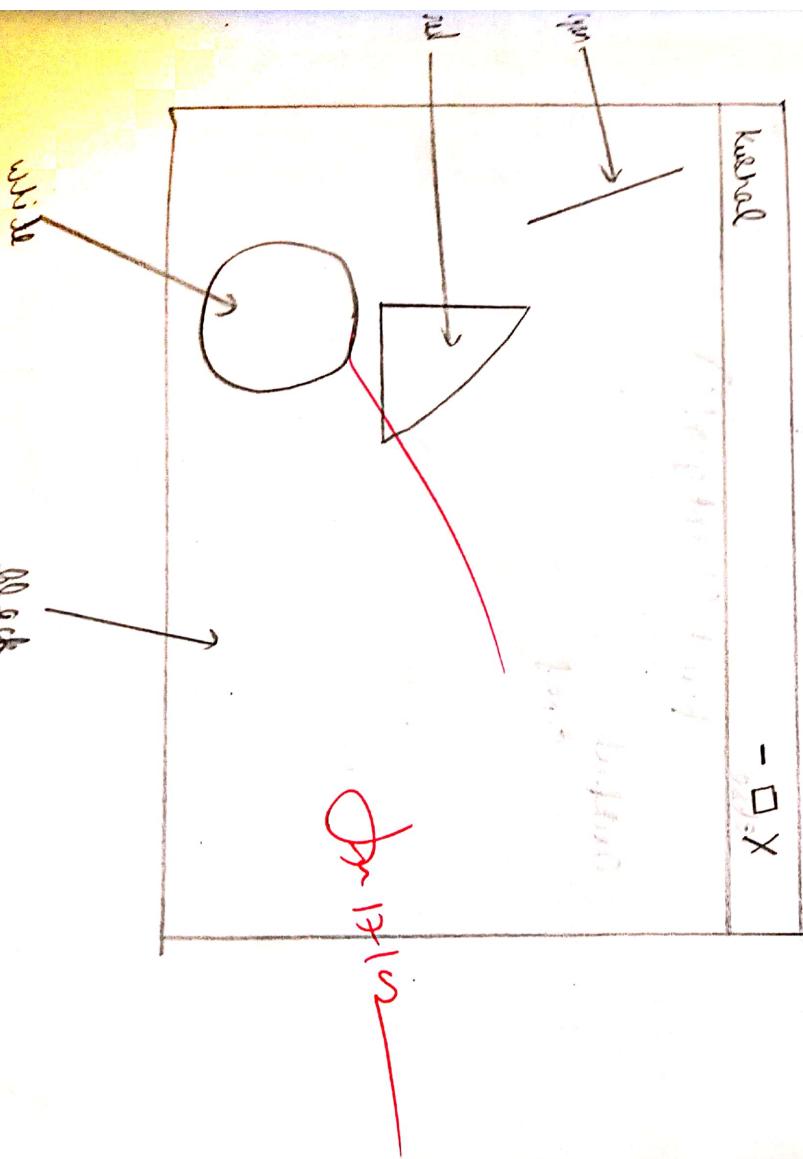
line = l1.create - line (50,140, 150, 160, fill:"cyan")

arc = l1.create - oval (20, 140, 150, 160, fill:"red")

l1.pack ()

root.mainloop ()

output



Ques. Write a python program to read data from a file named 'data.txt'. The file contains a single line of text. If the file contains 'good', print 'not good' else print 'good'.

```
import dbn  
db = dbn.open ("data base", "r")  
if db["w w w"] == None:  
    print ("good")  
else:  
    print ("not good")
```

Output:-
Good.

Practical-6

Aim:- To learn the use of database in python

- Step 1:- Import db library and use the open method for creating the data base but specify none of the data base along with the corresponding flag.

Step 2:- Use the objects for accessing the given WAP site and the correct regular for the web site

Step 3:- Check whether the given URL address with the regular of the pages is not equal to None then display the message from URL address else not found.

P.A.

Q)

Aim:- To demonstrate the use of cursor object
Step 1:- Import the corresponding library taking
of data base connection

Step 2:- Now Create Connection object using
SQLite library and connecting method
for create the database.

Step 3:- Now Create the Cursor objects using
cursor method from the connection
object. Create in steps.

Step 4:- Now we're executing method for
creating the table with the column
name and respective data types

Step 5:- Now with the cursor object we
insert different the entering the
values co-ordinating into the
different fields considering the
data types.

Step 6:- Use the commit method to complete
the transaction use the connection
objects.

Program -

import oracle.jdbc.*;

connection = Database.connect("student", "db")

c1 = connection.cursor()

c1.execute("create table student(name, rno, age);")

c1.execute("insert into student values('Rishabh',
123, 23, 01-2019)")

c1.execute("insert into student values('Sachin',
1362, 13, 01-2019)")

c1.execute("insert into student values('Ranji',
1892, 02, 02-2019)")

connection.commit()

c1.execute("Select * from student")

c1.executeQuery()

c1.execute("Drop table student")

50

03

Output :-

[("Sukhbir Kaur", 1763, 23-01-2020), ("Sukhbir", 1762, 23-01-2020),
("Guru", 1842, 02-02-2020)]

Mrs. Amrit

iii

Step 3: use the execute statement along with the cursor objects for accessing the values the data base using selecting from where clause

step 4: finally use the fetchall method for displaying the value for the table using the cursor objects

step 5: use the execute method and the drop table lighter for removing the database finally use the close method

cursor object

execute

fetchall

close

drop table

close

cursor object
use the cursor object

HUI-Project

Currency converter

- Label widget is used for giving the heading.
- Configure method is used for giving the background image.

②

- Canvas method is used to create a line
- Radio button is used for selecting the particular purpose.

Then message box is used to give the user a message.

Define function is used for the respective radio button with respect to the define function the label method is used to point the output.

Entry widget is used to take the inputs from user.

Submit button is used for further calculation of a inputted data in entry widget.

Message widget is used for giving a message to the user.

CURRENCY CONVERTER

52

Please select the conversion you want

- DOLLOR to RUPEES
- POUND to RUPEES
- SWISS to RUPEES
- EURO to RUPEES

Note :
PLEASE
ENTER
VALID INPUT

tk

- □ ×

CURRENCY CONVERTER

Please select the conversion you want

- DOLLOR to RUPEES
- POUND to RUPEES
- SWISS to RUPEES
- EURO to RUPEES

Enter The Currency In SWIS

SUBMIT

Note :
PLEASE
ENTER
VALID INPUT

tk

CURRENCY CONVERTER

Please select the conversion you want

- DOLLAR to RUPEES
- POUND to RUPEES
- SWISS to RUPEES
- EURO to RUPEES

Enter The Currency In POUNDS

5

SUBMIT

The Conversion Of POUNDS to Rupee is 482.85 Rs

Note :
PLEASE
ENTER
VALID INPUT

kushal final

```
from tkinter import*
root=TK()
#radiobutton 1#
def SUBMIT():
    num1=StringVar()
    def enter():
        f=float(num1.get())
        c=(f*74.01)
        label=Label(root,text=" The Converstion of DOLLOR to Rupee
Rs"%c,bg="white",fg="dark blue",font=15).pack(side=BOTTOM,ipadx=15,
root.configure(background="light blue",font=15).pack(side=BOTTOM,ipadx=15,
return
    l=Label(root,text="Enter The Currency In DOLLOR",font="Times 20
bold",fg="blue",bg="white")
    l.pack(anchor=CENTER,pady=10)

entry=Entry(root, textvariable=num1, bd=10, insertwidth=1, font=14, just
entry.pack()
    b=Button(root, text="SUBMIT", bg="white", fg="blue", font=14, comman
b.pack(anchor=CENTER)
#radiobutton 2#
def SUBMIT3():
    num1=StringVar()

    def enter():
        f=float(num1.get())
        c=(f*78.94)
        label=Label(root, text=" The Converstion Of SWIS To Rupee Is
Rs"%c,bg="white",fg="dark blue",font=15).pack(side=BOTTOM,ipadx=15,
return
        root.configure(background="light blue")
        l=Label(root, text="Enter The Currency In SWIS", font="Times 20
bold",fg="blue",bg="white")
        l.pack(anchor=CENTER,ipady=10)

entry=Entry(root, textvariable=num1, bd=10, insertwidth=1, font=14, jus
entry.pack()
    b=Button(root, text="SUBMIT", bg="white", fg="blue", font=14, comm
b.pack(anchor=CENTER)
#radiobutton 3#
def SUBMIT4():
    num1=StringVar()
    def enter():
        f=float(num1.get())
        c=(f*83.53)
        label=Label(root, text=" The Converstion Of EURO To Rupee Is
Rs"%c,bg="white",fg="dark blue",font=15).pack(side=BOTTOM,ipadx=15,
return
    root.configure(background="light blue")
    l=Label(root, text="Enter The Currency In EURO", font="Times 20
bold",fg="blue",bg="white")
    l.pack(anchor=CENTER,ipady=10)

entry=Entry(root, textvariable=num1, bd=10, insertwidth=1, font=14, ju
entry.pack()
    b=Button(root, text="SUBMIT", bg="white", fg="blue", font=14, comm
b.pack(anchor=CENTER)
#radiobutton 4#
def SUBMIT5():
    num1=StringVar()
```

```

kushal final
1=label(root,text="Enter The Currency In POUNDS",font="Times 20
bold",fg="blue",bg="white")
1.pack(anchor=CENTER,pady=10)

entry=Entry(root, textvariable=num1, bd=10, insertwidth=1, font=14, justify='center')
entry.pack()

b=button(root, text="SUBMIT", bg="white", fg="blue", font=14, command=enter)
b.pack(anchor=CENTER)

#first seen page#
root.configure(background="light blue")
1=Label(root, text="CURRENCY CONVERTER", font="Times 20
bold",fg="red",bg="yellow")
1.pack(anchor=CENTER, pady=10)

#line#
a=Canvas(root, bg="light blue", width=500, height=5)
a.pack()

a.create_line(0,20,500,20,fill="black")
12=Label(root, text="######
Please select the converstion you want
#####
", font="Helvetica 10", bg="light blue")
12.pack(anchor=NW,padx=15,pady=20)

v=IntVar()
r=Radiobutton(root, text="DOLLAR TO RUPEES", bg="light blue", font="Courier
variable=v, value=1, command=SUBMIT)
r1=Radiobutton(root, text="POUND TO RUPEES", bg="light blue", font="Courier
variable=v, value=2, command=SUBMIT2)
r2=Radiobutton(root, text="SWISS TO RUPEES", bg="light blue", font="Courier
variable=v, value=3, command=SUBMIT3)
r3=Radiobutton(root, text="EURO TO RUPEES", bg="light blue", font="Courier
variable=v, value=4, command=SUBMIT4)
r.pack(anchor=NW,padx=5)
r2.pack(anchor=NW,padx=5)
r3.pack(anchor=NW,padx=5)

#message box#
msg=Message(root, text="Note : PLEASE ENTER VALID INPUT", bg="light
blue", fg="black", font="Times 15 bold").pack(side=BOTTOM)
mainloop()

```

55

Project :-
Database

The Database is made to show District wise population of each District

Python 3.7.4 (tags/v3.7.4:e09359112e, Jul 8 2019, 20:34:20) [MSC v.1916 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

```
>>> import os,sqlite3

>>> connection=sqlite3.connect("population.db")

>>> c=connection.cursor()

>>> c.execute("create table Andhrapradesh(District char,Population int,Area_kmsq int)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into Andhrapradesh values('Anantapur',4083315,19130)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into Andhrapradesh values('Chittoor',4170468,15152)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into Andhrapradesh values('EastGodavari',5151549,10807)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into Andhrapradesh values('Guntur',4889230,11391)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into Andhrapradesh values('YSR Kadapa',2884524,15351)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into Andhrapradesh values('Krishna',4529009,8727)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into Andhrapradesh values('Kurnool',4046601,17658)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into Andhrapradesh values('Nellore',296608,13076)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into Andhrapradesh values('Prakasam',3392764,17626)")
```

```
<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into AndhraPradesh values('Srikakulam',2699471,5837)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into AndhraPradesh values('Visakhapatnam',4288113,11161)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into AndhraPradesh values('Vizianagaram',2342868,6539)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into AndhraPradesh values('WestGodavari',3934782,7742)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("create table Kerala(District char,Population int,Area_kmsq int)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into Kerala values('Malappuram',4110956,3550)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into Kerala values('Thiruvananthapuram',3307284,2192)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into Kerala values('Ernakulam',3279860,3086)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into Kerala values('Thrissur',3110327,3032)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into Kerala values('Kozhikode',3089543,2344)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into Kerala values('Palakkad',2810892,4480)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("insert into Kerala values('Kollam',2629702,2491)")
```

```
<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("Insert into Kerala values('Kannur',2525637,29661)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("Insert into Kerala values('Alappuzha',2121943,1414)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("Insert into Kerala values('Kottayam',1979384,2208)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("Insert into Kerala values('Kasaragod',1302600,1992)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("Insert into Kerala values('Pathanamthitta',1195537,2637)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("Insert into Kerala values('Idukki',1107453,4358)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.execute("Insert into Kerala values('Wayanad',816558,2131)")

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> connection.commit()

>>> c.execute('select * from AndhraPradesh')

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.fetchall()

[[('Anantapur', 4083315, 19130), ('Chittoor', 4170468, 15152), ('EastGodavari', 5151549, 10807), ('Guntur', 4889230, 11391), ('YSR Kadapa', 2884524, 15351), ('Krishna', 4529009, 8727), ('Kurnool', 4046601, 17658), ('Mellore', 296608, 13076), ('Prakasam', 3392764, 17626), ('Srikakulam', 2699471, 5837), ('Visakhapatnam', 4286113, 11161), ('Vizianagaram', 2342868, 6539), ('WestGodavari', 3934782, 7742)]]

>>> c.execute('select * from Kerala')

<sqlite3.Cursor object at 0x000001985DF0A340>
```

```
>>> c.fetchall()

[('Malappuram', 4110956, 3550), ('Thiruvananthapuram', 3307784, 2192), ('Ernakulam', 3279860, 3086),
 ('Thissur', 3110327, 3032), ('Kozhikode', 3089543, 2344), ('Palakkad', 2810892, 4480), ('Kollam', 2629702,
 2491), ('Kannur', 2525637, 2966), ('Alappuzha', 2121943, 1414), ('Kottayam', 1979384, 2208), ('Kasaragod',
 1302600, 1992), ('Pathanamthitta', 1195537, 2637), ('Idukki', 1107453, 4358), ('Wayanad', 8165558, 2131)]
```

>>> c.execute('select District, Population from Andhraapradesh union select District,Population from Kerala')

```
<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.fetchall()

[(('Alappuzha', 2121943), ('Anantapur', 4083315), ('Chittoor', 4170468), ('EastGodavari', 535349), ('Ernakula
3 279860), ('Guntur', 4889230), ('Idukki', 1107453), ('Kannur', 2525637), ('Kasaragod', 130600), ('Kollam',
2629702), ('Kottayam', 1979384), ('Kozhikode', 3089543), ('Krishna', 4529009), ('Kurnool', 446601),
('Malappuram', 4110956), ('Nellore', 296608), ('Palakkad', 2810892), ('Pathanamthitta', 1195537), ('Prakasai
3 392764), ('Srikakulam', 2699471), ('Thiruvananthapuram', 3307284), ('Thrissur', 3110327), ('Visakhapatnam
4288113), ('Vizianagaram', 2342868), ('Wayanad', 816558), ('WestGodavari', 3934782), ('Vizianagaram', 288452
>>> c.execute('select count(District) from Andhraapradesh')

<sqlite3.Cursor object at 0x000001985DF0A340>

>>> c.fetchall()

[(13,)]
```

(14,)

```
>>> c.execute('select count(District) from Kerala')

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.fetchall()
```

(14,)

```
>>> c.execute('select sum(Population) from Andhraapradesh')

<sqlite3.Cursor object at 0x000001985DF0A340>
>>> c.fetchall()

[(46709302,)]
```

>>> c.execute('select sum(Population) from Kerala')

```
<sqlite3.Cursor object at 0x000001985DF0A340>
>>>c.fetchall()
[[33387676,]]
>>>c.execute("drop table Andhraapradesh")
<sqlite3.Cursor object at 0x00000269D76DB5E0>
>>>c.execute("drop table Kerala")
<sqlite3.Cursor object at 0x00000269D76DB5E0>
>>>c.fetchall()
```

✓ ✓
✓ ✓