

```

15
flobj = open("abc.txt", "w") # file open(write mode)
flobj.write("Computer Science Subject" + "\n")
flobj.write("DBMS in python in DS\n") # file write
flobj.close() # file close

flobj = open("abc.txt", "r") # read mode
# Read()
str1 = flobj.read()
print("The output of Read method: ", str1)
flobj.close()
>>> ('The output of Read method: ', 'Computer Science Subjects
in DBMS in python is DS\n')

# Readline()
flobj = open("abc.txt", "r")
str2 = flobj.readlines()
print("The output of Readline method: ", str2)
flobj.close()
>>> ('The output of Readline method: ', 'Computer Science
Subject\n')

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

>>> ('The output of Readlines method: ', ['Computer Science
Subject\n', 'DBMS\n', 'python\n', 'DS\n'])

```

MON
25/11/19

25

PRACTICAL NO: 1

Objective: Demonstrate The Use of different file access modes, different attributes Read method.

Step 1:- Create a file object using Open method and use the write access mode followed by writing some Content into the file.
Then closing the file.

Step 2:- Now open the file in Read mode & Then use Read(), Readline() & Readlines(), & Store the Output in variable & finally Display content of the variable.

Step 3:- Now use The file object for finding the name of the file, the file mode in which it is opened whether the file still open or Close & Finally The output of the the softspace delimiter.

Step 5: Now open the File object in write mode & write some another content. Close the file object again. The file object in 'wt' mode is the update mode & write content.

Step 5: Open file object in Read mode & display update written content & close. Open again in 'wt' mode with parameter passed and display the output subsequently.

Step 6: Now open file object in append mode & write method write content. Close the file object again. Open the file object in Read mode & display append output.

File attributes

```

o = filobj.name
print("name of file (name of attributes) : ", o)
>>> ('name of file (name attributes)', 'abc.txt')
b = filobj.closed
print("(closed) attributes : ", b)
>>> ('closed attributes : ', True)
c = filobj.mode
print("file mode : ", c)
>>> ('file mode ', 'w')
d = filobj.softspace
print("softspace : ", d)
>>> ('softspace : ', 0)

```

w mode

```

filobj = open('abc.txt', "wt")
filobj.write("DBMS")
filobj.close()

```

r+ mode

```

filobj = open('abc.txt', "r+")
str1 = filobj.read(4)
print("output of r+ mode : ", str1)
filobj.close()
>>> ('output of r+ mode : ', 'Abhishek')

```

write mode

```

filobj = open('abc.txt', "wt")
filobj.write("DBMS")
filobj.close()

```

Read mode

```

filobj = open('abc.txt', "r")
str2 = filobj.read()
print("output of Read mode : ", str2)
>>> ('output of Read mode : ', 'Abhishek')

```

open mode

```
fileobj = open('abc.txt', 'a')
fileobj.write("data structure")
fileobj.close()
fileobj = open('abc.txt', 'r')
str3 = fileobj.read()
print("Output of append mode: ", str3)
fileobj.close()
```

```
>>> ('Output of append mode: ', 'Abhishek', 'data structure')
```

tell()

```
fileobj = open('abc.txt', 'r')
pos = fileobj.tell()
print("tell(): ", pos)
fileobj.close()
>>> ('tell(): ', pos)
```

seek()

```
fileobj = open('abc.txt', 'r')
str4 = fileobj.seek(0, 0)
str8 = fileobj.read(10)
print("The beginning of the file: ", str8)
```

finding length of different lines exists within lines

```
fileobj = open('abc.txt', 'r')
str9 = fileobj.readlines()
print("Output: ", str9)
for line in str9:
    print(len(line))
fileobj.close()
```

```
>>> ('Output: ', ['College database'])
```

MON
2/12/19

18

PRACTICAL NO: 2

Aim: The use of Iteration & Iterable Method.

- 1) Create a tuple & use iter.
- 2) Next Method to display values.
- 3) for assign the individual value. For the given obj can use for loop.

CODE :-

28

```
mytuple1 = ("Banana", "Orange", "apple")
myiter1 = iter(mytuple1)
print(next(myiter1))
for i in mytuple1:
    print(i).
```

Output:

```
for i in mytuple1:
    print(i)
```

```
>>> banana
>>> Orange
>>> Apple.
```

Dr. Ali M.

PRACTICAL NO : 5.

Aims:- programs to demonstrate Exception Handling

(1) program to demonstrate the use of I/O Errors.

- Step 1 :- Use the try block to define the Normal course of action for eg Define the file object & open the file in the write or read mode & write some content onto the file,
- Step 2 :- Use the except Block with I/O Errors, as environment is convey the appropriate message to user, else display the message that the operation is carried successfully.

(2) Program to demonstrate multiple exception viz.

I/O Error & value Error

- Step 1 :- Use the try Block & define file object, the file object the read mode & write some content onto the file.
- Step 2 :- Also accept the value from the user & if it is a valid value display the Entering value & terminate the condition by using the break statement.

- Step 3:- Define the except Block for I/O Error and value Error

1. try:

```
fileobj = open("abc.txt", "w")
fileobj.write("python is interpreted language in python is identified language")
```

30

except IOError:

```
print("There is an Environment Error")
```

else:

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

>>> Operation Successfull

2. while True:

try:

```
fileobj = open("abc.txt", "r+")
fileobj.write("python is identified language in python is an interpreted language")
```

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

break

except IOError:

```
print("There is Environment Error")
```

except ValueError:

```
print("The value is invalid")
```

>>>

```
Enter a number: abc
```

```
The value is invalid
```

```
Enter a number:pqr
```

```
The value is invalid
```

```
Enter number: 73
```

```

## match()
import re
pattern = r'(\d+)'
Sequence = ('FYS represents Computer Sequence String',
            'Sequence = (\d+)' if re.match(pattern, Sequence),
            'if \d+ matches pattern found!') if
            print('Matched pattern found!') else:
            print("NOT FOUND!")

```

else:
print("NOT FOUND!")
''' matched pattern found!
'''
Numerical values (Stringification)

```

import re
pattern = r'\d+'
String = ('Hello123', 'howdy789', '45howzo')
Output = re.findall(pattern, String)
print(Output)

```

'''
[123, 789, 45]

```

## split()
import re
pattern = r'\d+'
String = ('Hello123', 'howdy789', '45howzo')
Output = re.split(pattern, String)
print(Output)

```

'''
[Hello', 'howdy', ', ', 'howzo']

Step 4:- Import re module declare string and accordingly declare pattern. Replace the Blank Space within string. use `re.replace()` with 3 argument & print the string without spaces.

Step 5:- Import re module & declare a sequence. Use search method for finding subsequence. only use the `Group()` with dot operator as `search()` gives memory loc on using group() if with shows the matched string.

Step 6:- Import re module declare list with numbers. Use the Conditional Statement. Here we have use up for Conditional Statement used if Conditional statement to use if condition for checking first number are in range either 8 or 9 & Next Number are in range 0 to 9 & check whether the Entered numbers are equal to 10. If criterias matches print all numbers. matches. otherwise print failed.

```
# no_Space!
import re
String = 'abc def ghi'
pattern = r'\s+'
Replace = ''
v1 = re.sub(pattern, Replace, String)
print(v1)
>>> abcdeFghi

# groups()
import re
Sequence = 'python is interesting language'
v = re.search('A python', Sequence)
print(v)
v1 = v.group(0)
print(v1)
>>> <_ SRE_match object at 0x0281DF00>
python
# verifying the given set of phone.numbers.
import re
list = ['8000567891', '9145673210', '7865432981',
        '19876543201']

for the values in list:
    if re.match(r'[8-9]\d{3}\d{3}', value):
        print("Criteria matched for all numbers!")
    else:
        print("Criteria Failed!")

```

>>> Criteria matched for all numbers
 Criteria matched for all numbers,
 Criteria for 101
 Criteria matched for all numbers.

vowels.

```
import re
str = 'plant is life overall'
output = re.findall(r'[aeiouAEIOU]', str)
print(output)
>>> ['i', 'o', 'a', 'a', 'u']
```

host & domain

```
import re
seq = 'abc.tesla.com, xyz@gmail.com'
pattern = r'([a-zA-Z]+)\.([a-zA-Z]+)\.([a-zA-Z]+)'
output = re.findall(pattern, seq)
print(output)
```

>> [('abc', 'tesla', 'com'), ('xyz', 'gmail', 'com')]

Counting of first 2 letters.

```
import re
s = 'mr. a, ms. b, ms. c, mm. t'
p = r'([mr\ms\mm]\.([a\c\t]))'
o = re.findall(p, s)
print(o)
m = 0
f = 0
for i in o:
    m += 1
    f += 1
    print(m, f)
```

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

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

Step 9: Import re module Enter a string use pattern to display only two elements of the part of the string Use.findall() declare two variable with initial values as zero use for condition & sequentially use the if conditional check whether conditions satisfy add up the or else increment value. & display the values subsequently

Jyoti

if ($C_4 == rms$)
 $f = f + 1$

else :

$m = m + 1$

point C₄ No. of molecules is : ", m")

point C₅ No. of female is : ", n")

else C₄ ", rms", "rms", "rms", "rms")

(No. of males is : ", 2)
(No. of females is : ", 2)

PRACTICAL NO:5

Aim:- To make use of GUI Application along with Basic pack method.

Algorithms:-

Step 1:- Use the Tkinter library for importing the feature of text widgets

Step 2:- Create a variable from a text variable & position it into the parent window

Step 3:- Use the pack() along with the object created from text method & we the parameters,

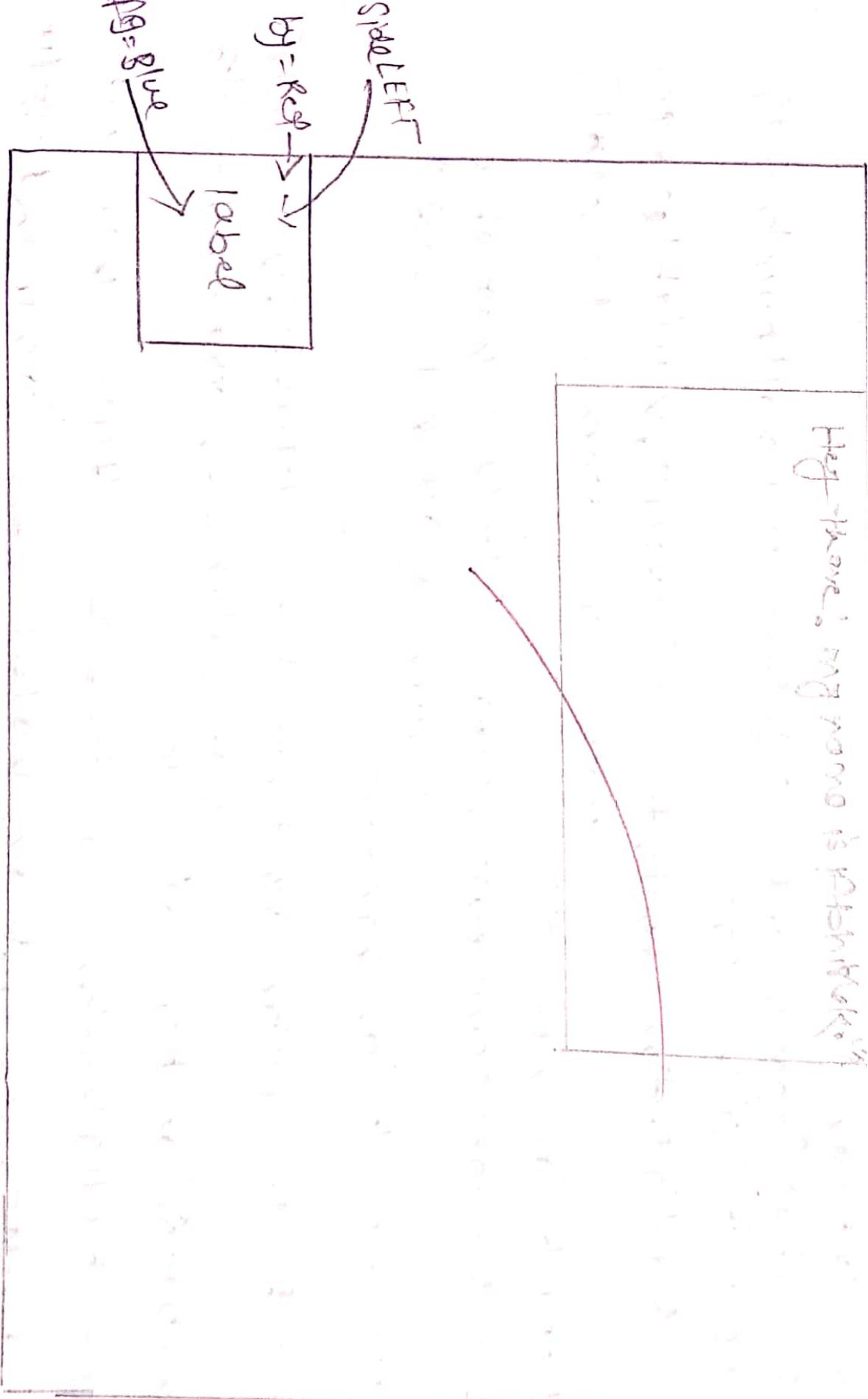
\downarrow side = top , pack = 20 , ipadx = 40 ,
ipady = 50

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

- 1) Name of parent window
 - 2) To text attributes which defines the size
 - 3) The Background colour(bg)
 - 4) The foreground colour(fg)
- Now use the pack command with relevant attributes

Hey there! My name is Peter. Welcome!



```
from tkinter import *
def Sel1():
    Selection = "Abhishek"
    label.config(text=selection)
def Sel2():
    Selection = "Raj"
    label.config(text=selection)
def Sel3():
    Selection = "Parinay"
    label.config(text=selection)
def Sel4():
    Selection = "Sachin"
    label.config(text=selection)
root = Tk()
var = IntVar()
L1 = Label(root, text="Select any roll number")
L1.pack(side=TOP)
R1 = Radiobutton(root, text="1741", variable=var, value=0, command=Sel1)
R1.pack(anchor=N)
R2 = Radiobutton(root, text="1742", variable=var, value=1, command=Sel2)
R2.pack(anchor=N)
R3 = Radiobutton(root, text="1743", variable=var, value=2, command=Sel3)
R3.pack(anchor=N)
R4 = Radiobutton(root, text="1744", variable=var, value=3, command=Sel4)
R4.pack(anchor=N)
label = Label(root)
label.pack(side=Bottom)
```

Step 4: Now define a label object & place it in parent window using pack method & finally use mainloop method.

```
from Tkinter import *  
  
root = Tk()  
  
label = Label(root, text="Hello World")  
  
label.pack()  
  
root.mainloop()
```

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 bar parent window & Create an object from Scrollbar & place it on it on the parent window so created.

Step 3 :- Create an object of Label method to provide a heading & place it on parent window

Step 4 :- Use pack method along object or Scrollbar method & use argument sides & fill

~~Step 5 :- Create an object of listbox method & place it into parent window with attribute
→ yscroll command~~

~~Step 6 :-~~

Step 6 :- Use the 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 & use command attribute

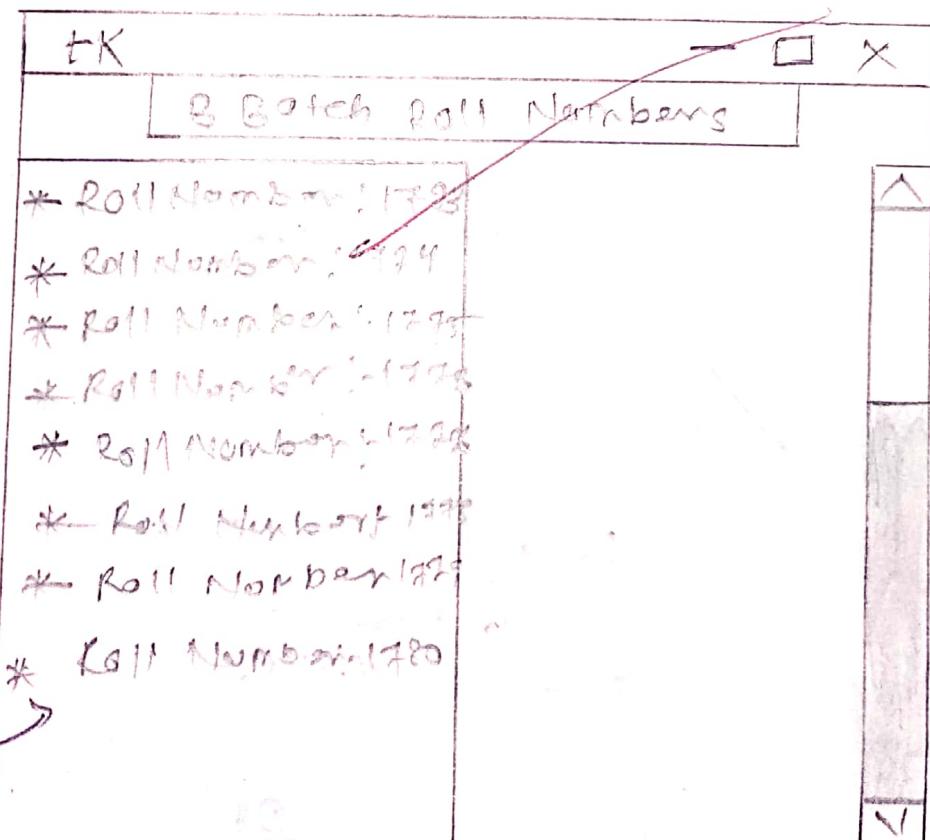
Step 8 :- Finally Call the mainloop method.

PROGRAM:-

40

```
from tkinter import *
root = Tk()
root.geometry('400x400')
L = Label(root, text = "B Batch Numbers", bg = "Black", fg = "white")
L.pack()
scroll1 = Scrollbar(root)
scroll1.pack(side = RIGHT, fill = Y)
mylist = Listbox(root, yscrollcommand = scroll1.set, bg = "light blue")
for num in range(4, 81):
    mylist.insert(END, "* Roll Number : " + str(num))
mylist.pack(side = LEFT, fill = BOTH)
scroll1.config(command = mylist.yview)
root.mainloop()
```

OUTPUT:-



PROGRAM

```

from tkinter import *
from tkinter import messagebox
def msgbox():
    messagebox.showaskyesno("Action", "Do you want to Continue?")
    messagebox.showerror("Error", "Can't load the project")

```

root = Tk()

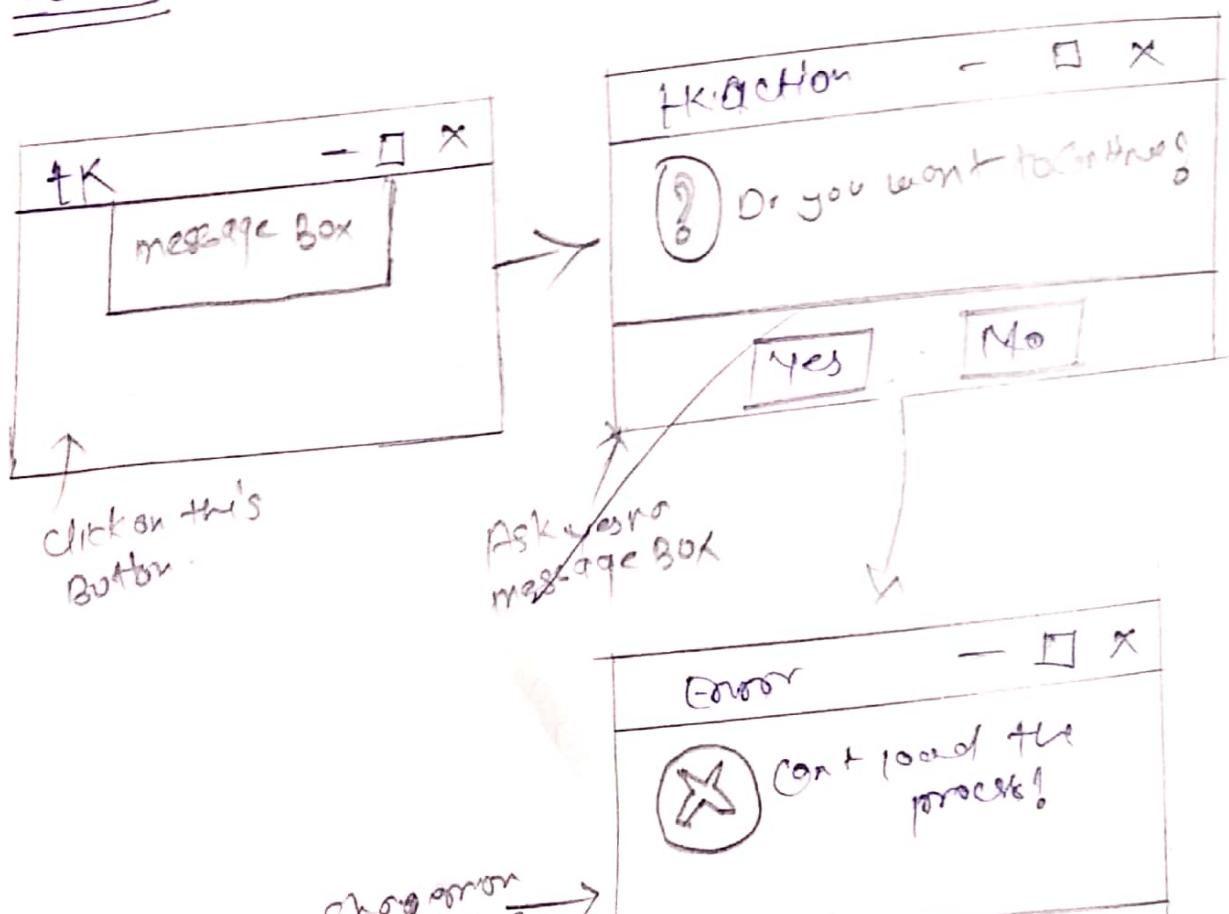
root.config(bg="grey")
 root.config(bg="grey", text="message Box", bg="light blue", command=msgbox)

b1 = Button(root, text="message Box")

b1.pack()

root.mainloop()

OUTPUT:



PROGRAMMING

from tkinter import *

def main():

root = Tk()

root.geometry("450x500")

root.config(bg = "light green")

root.title("window1")

root.minsize(300, 200)

B1 = Button(root, text = "Next", command = next1)

B2 = Button(root, text = "Exit", command = form)

B2.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B1.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

A2

Steps: Define program to transverse various window win
from the Bottom widget

Algorithm:

Step 1: Import the relevant method from tkinter library.

Step 2: Define a Bottom object which will be placed on the current window to transverse &

Define another Bottom which will be used to But from the windows & place it in current window.

Step 3: Define function & create objects of

given window by using the these method
namely config, title, minsize

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

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

steps:- Define two Buttons which will be placed on the main window; one to transverse
Another window and the other to form

B1 = Button(top, text = "main page", command = main)

B1.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B2 = Button(top, text = "Exit", command = form)

B2.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B3 = Button(top, text = "Continue", command = main)

B3.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B4 = Button(top, text = "Exit", command = form)

B4.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B5 = Button(top, text = "Continue", command = main)

B5.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B6 = Button(top, text = "Exit", command = form)

B6.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B7 = Button(top, text = "Continue", command = main)

B7.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B8 = Button(top, text = "Exit", command = form)

B8.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B9 = Button(top, text = "Continue", command = main)

B9.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B10 = Button(top, text = "Exit", command = form)

B10.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B11 = Button(top, text = "Continue", command = main)

B11.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B12 = Button(top, text = "Exit", command = form)

B12.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B13 = Button(top, text = "Continue", command = main)

B13.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B14 = Button(top, text = "Exit", command = form)

B14.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B15 = Button(top, text = "Continue", command = main)

B15.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B16 = Button(top, text = "Exit", command = form)

B16.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B17 = Button(top, text = "Continue", command = main)

B17.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B18 = Button(top, text = "Exit", command = form)

B18.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B19 = Button(top, text = "Continue", command = main)

B19.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B20 = Button(top, text = "Exit", command = form)

B20.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B21 = Button(top, text = "Continue", command = main)

B21.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B22 = Button(top, text = "Exit", command = form)

B22.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B23 = Button(top, text = "Continue", command = main)

B23.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B24 = Button(top, text = "Exit", command = form)

B24.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B25 = Button(top, text = "Continue", command = main)

B25.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B26 = Button(top, text = "Exit", command = form)

B26.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B27 = Button(top, text = "Continue", command = main)

B27.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B28 = Button(top, text = "Exit", command = form)

B28.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B29 = Button(top, text = "Continue", command = main)

B29.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B30 = Button(top, text = "Exit", command = form)

B30.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B31 = Button(top, text = "Continue", command = main)

B31.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B32 = Button(top, text = "Exit", command = form)

B32.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B33 = Button(top, text = "Continue", command = main)

B33.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B34 = Button(top, text = "Exit", command = form)

B34.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B35 = Button(top, text = "Continue", command = main)

B35.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B36 = Button(top, text = "Exit", command = form)

B36.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B37 = Button(top, text = "Continue", command = main)

B37.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B38 = Button(top, text = "Exit", command = form)

B38.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B39 = Button(top, text = "Continue", command = main)

B39.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B40 = Button(top, text = "Exit", command = form)

B40.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B41 = Button(top, text = "Continue", command = main)

B41.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B42 = Button(top, text = "Exit", command = form)

B42.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B43 = Button(top, text = "Continue", command = main)

B43.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B44 = Button(top, text = "Exit", command = form)

B44.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B45 = Button(top, text = "Continue", command = main)

B45.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B46 = Button(top, text = "Exit", command = form)

B46.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B47 = Button(top, text = "Continue", command = main)

B47.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B48 = Button(top, text = "Exit", command = form)

B48.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B49 = Button(top, text = "Continue", command = main)

B49.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B50 = Button(top, text = "Exit", command = form)

B50.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B51 = Button(top, text = "Continue", command = main)

B51.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B52 = Button(top, text = "Exit", command = form)

B52.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B53 = Button(top, text = "Continue", command = main)

B53.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B54 = Button(top, text = "Exit", command = form)

B54.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B55 = Button(top, text = "Continue", command = main)

B55.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B56 = Button(top, text = "Exit", command = form)

B56.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B57 = Button(top, text = "Continue", command = main)

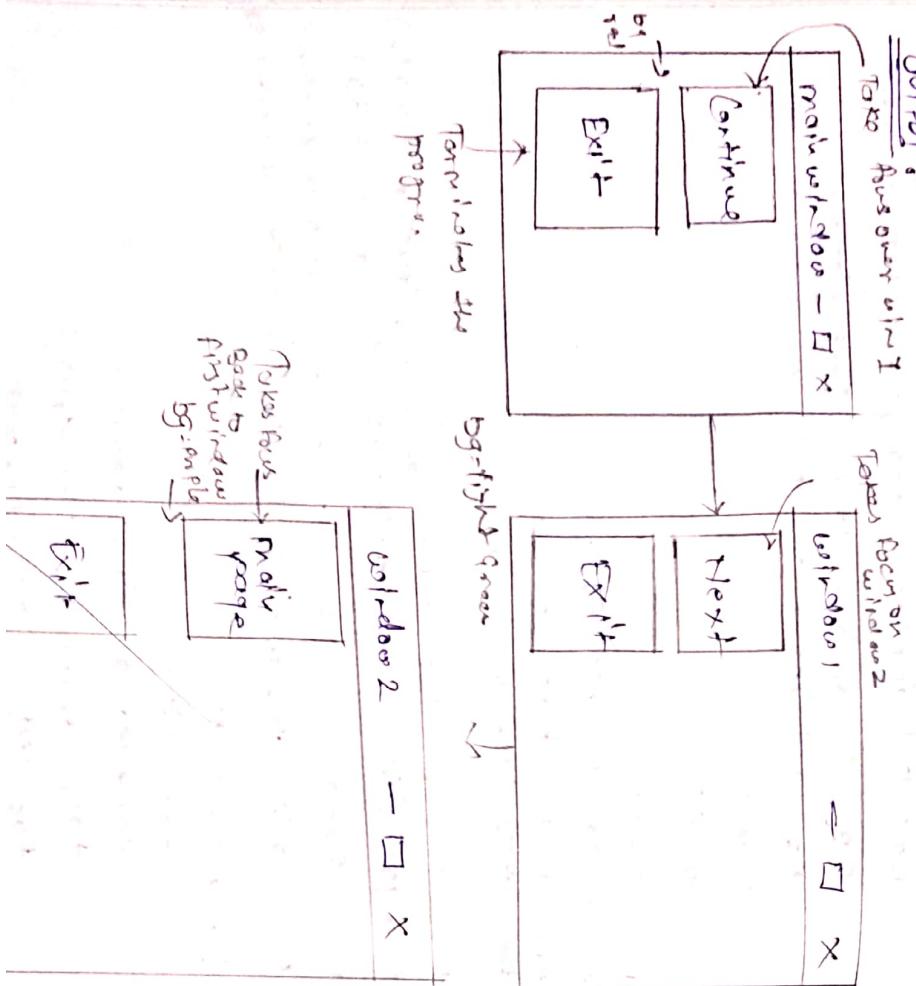
B57.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

B58 = Button(top, text = "Exit", command = form)

B58.grid(ipadx = 50, ipady = 40, padx = 20, pady = 30)

42

B2 = Bottom (Top, text = "Exit", command = term) B2 • Top (Top, text = "Exit", ipady = 10, padx = 20, pady = 10) mainloop()



43

Step 7: Define another function which will
call various button placed on third window.
Define two buttons respectively & use the
grid method along with two button.

Step 8: finally call the main loop method.

Main

Spinbox

→ Create an object from the method & ~~get~~
only Create an object from the Spinbox
method

Steps:

→ Make the object & created onto the parent
window & trigger the corresponding event.

Steps:

→ Use the another pack method to provide the
direction using anchor method.

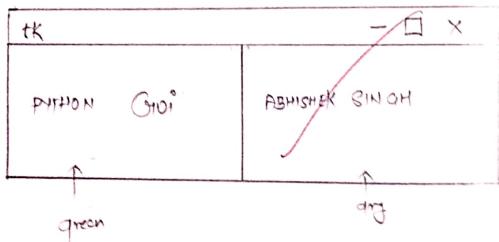
Steps:

→ Use the mainloop method to terminate

44
SOURCE CODE:-

```
from tkinter import *
root = Tk()
P = PanedWindow(bg = "red")
P.pack(fill = BOTH, expand = 1)
L1 = Label(P, text = "PYTHON GUI"), bg = "green"
D.add(L1)
P1 = PanedWindow(P, orient = VERTICAL, bg = "blue")
P1.add(P1)
P1.add(L2)
root.mainloop()
```

OUTPUT:-



45

* paned window.

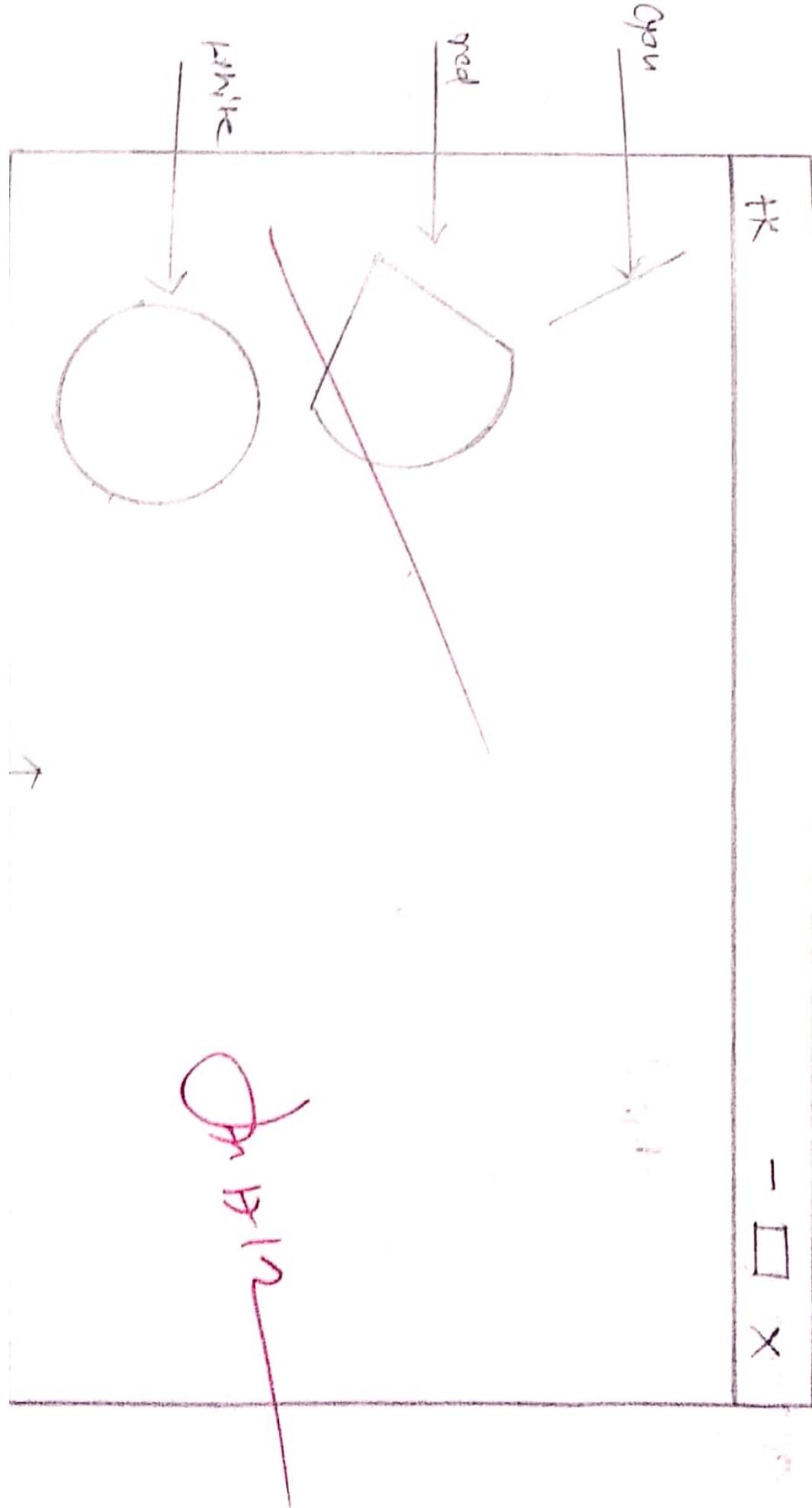
step 3 Create an object from paned window and use the pack method with the attribute fill & expand:

step 4 Create an object from the label method & put it onto the paned window with the text attribute & use the add method to Embed the new object

step 5 Similarly create second paned window object & add it onto the 1st paned window with orientation specified.

step 6 Now create label object & place it onto the 2nd paned window object and add it onto the 2nd pane

step 7 Now use the mainloop method to terminate



Step 1: Import db library & use the open method for creating the database by specifying name of the data base along with the corresponding flag.
db = db["C:\Users\user\Desktop\good"]
if db["C:\Users\user\Desktop\good"] == None:
 print("Good")

else:
 print("Not good")

Output:-

Good

Step 2: Use the object for accessing the cursor size & the corresponding register for the cursor size.

Step 3: Check whether the given URL address with the register of the page is not equal to none then display the message from URL address else not found.

Good

Ex

Step 1: Import the Corresponding library taken
of data Base Connection

Step 2: Now Create Connection Object using
Connecting method for
Split library & Create the New data base

Step 3: Now create the Cursor object using
Cursor method from Connecting object
Create in step

Step 4: Now use the Executing method for Chaining the table with the column name
to respective data type

Step 5: Now with Cursor object we insert
Statement for Entering the values Consider
Putting into the different field Considering
the data type

Step 6: Use the Commit method to complete the
transaction use the Connection object.

Step 7: Use the execute statement along with
the Cursor object for executing the
values the Data base using the select
from other table

PROJECT TO DEMONSTRATE VARIOUS WIDGETS

Python project-1

Title : various widgets in python

```
from tkinter import *
from tkinter import messagebox
def Quit():
    top.mainloop()
def show():
    name=E1.get()
    messagebox.showinfo("Welcome", "Welcome" +name)
def error():
    messagebox.showwarning("OOPS...", "wrong password entered")
Quit()
def check():
    v1=E2.get()
    if v1=="abhishek":
        show()
    else:
        error()
top=Tk()
top.title("Login form")
L1=Label(top,text="enter Loginname")
L1.grid(row=0,column=0)
```

E1=Entry(top, bd=6)

```
E1.grid(row=0,column=1)
```

L2=Label(top,text="Enter your password")

```
L2.grid(row=1,column=0)
```

```
E2=Entry(top, bd=6, show="*")
```

```
E2.grid(row=1,column=1)
```

```
B1=Button(top,width=25,text="Login",command=check)
```

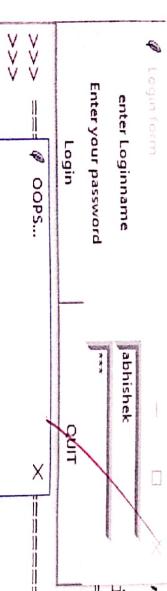
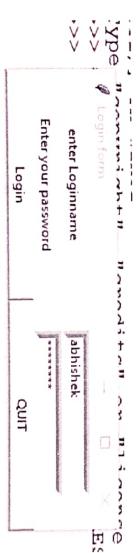
```
B1.grid(row=2,column=0)
```

```
B2=Button(top,width=25,text="QUIT",command=Quit)
```

```
B2.grid(row=2,column=1)
```

```
top.mainloop()
```

Output:



PROJECT TO DEMONSTRATE DATABASE:-

Project-2

```
import sqlite3  
con=sqlite3.connect('population.db')  
cur=con.cursor()  
cur.execute("DROP TABLE IF EXISTS PopByRegion")  
cur.execute("CREATE TABLE PopByRegion(Region TEXT,Population INTEGER)")  
ans="yes"  
count=0  
while ans=="yes":  
    x=input("ENTER THE REGION NAME...")  
    y=int(input("ENTER THE POPULATION FOR %s "%x))  
    cur.execute("INSERT INTO PopByRegion(Region,Population)VALUES('%s',%d)"%(x,y))  
    count=count+1  
    ans=input("Do you want to continue insertion?")  
print("You have insertion %d records "%count)  
con.commit()  
con.close()
```

Output:

```
ENTER THE REGION NAME...Nashik  
ENTER THE POPULATION FOR Nashik 675657  
Do you want to continue insertion?yes  
ENTER THE REGION NAME...Mumbai  
ENTER THE POPULATION FOR Mumbai 5645434  
Do you want to continue insertion?no  
You have insertion 2 records
```

#FETCHING

```
import sqlite3  
con=sqlite3.connect('population.db')  
cur=con.cursor()  
sql="SELECT* FROM PopByRegion WHERE Population>10000"  
cur.execute(sql)  
results=cur.fetchall()  
for row in results:  
    rn=row[0]  
    po=row[1]  
    print(rn)  
    print(po)  
    con.close()
```

OUTPUT:

```
Nashik  
675657  
Mumbai  
5645434
```

#UPDATE

```

import sqlite3
con=sqlite3.connect('population.db')
cur=con.cursor()
sql="UPDATE PopByRegion SET Population=Population+50000 WHERE Region='Nashik'"
try:
    cur.execute(sql)
    print("Record Updated...")
    sql2="SELECT* FROM PopByRegion where Region='Nashik'"
    cur=con.cursor()
    cur.execute(sql2)
    result=cur.fetchone()
    print("Region %s"%result[0])
    print("Population %d"%result[1])
    con.commit()
except:
    con.rollback()
    con.close()

```

OUTPUT:

Record Updated...
 Region Nashik
 Population 6806265

#DELETE

```

import sqlite3
con=sqlite3.connect('population.db')
cur=con.cursor()
sql="DELETE FROM PopByRegion where Region='Pune'"
try:
    cur.execute(sql)
    print("Record deleted...")
    sql2="SELECT* FROM PopByRegion"
    cur=con.cursor()
    cur.execute(sql2)
    result=cur.fetchall()
    for row in result:
        ro=row[0]
        po=row[1]
        print("Region",result[0])
        print("Population",result[1])
        con.commit()
except:
    con.rollback()
    con.close()

```

OUTPUT:

Record deleted...
 Region ('mumbai', 656546)

22/11