

Aim:-

Demonstrate the use of different file accessing mode, different attributes like read method.

Algorithm

Step 1: Create a file object using open method and use the write accessing mode followed up by writing some contents on the file and closing the files.

Step 2: Now open the file in read mode and than use read(), readLine(), readLines() and store output in variable and display the contents of 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 is still opened or closed and finally the value of the soft space attribute

IS.

Step4:- Now open the file obj in write mode
write some another content close subsequently
then again open the file obj in wt mode
i.e. the update mode and write contents

Step5: Open file objects in read mode, display the
update written contents & close open
again in rt mode with parameters
passed & display the output subsequently

Step6: Open the file obj in read mode declare a
variable & perform file obj . tell() and
store the output consequently.

Step7:- Use the seek() with arguments with
opening the file obj in read mode & closely
subsequently.

Step8:- Open the file obj with read mode
also use the readlines() &
store the output consequently.

```
file obj = open('abc.txt', 'w')
file obj . write("Computer science subj "+"AI")
file obj . write("DBMS In Python In DS In ")
file obj . close()
file obj = open("abc.txt", "r")
file obj . read()
str1 = file obj . read()
print("The output of read method: ", str1)
file obj . close()
file obj
Output:-
```

computer Science subject
DBMS
PYTHON
DS

```
* Readlines
file obj = open('abc.txt', 'r')
str2 = file obj . readlines()
print("The output of readlines method: ", str2)
file obj . close()
file obj
Output of readlines method: Computer
Science subject
```

```
* readline
file obj = open('abc.txt', 'r')
str3 = file obj . readline()
print("The output of readline method: ", str3)
file obj . close()
file obj
Output of readline method:
Computer Science subject:  
DBMS  
PYTHON
```

```

#file attribute
a = file obj . name
Print ("Name of file : ", a)
>>> Name of file : abc.txt
b = file Obj . closed
Print ("(close) attribute : ", b)
>>> (close) attribute , True

c = file obj . mode
Print ('File Obj mode : ', c)
>>> file obj mode : 'r'

#wt mode
File obj = open ('abc.txt' , 'wt')
File obj . write ('Abhay')
File obj . close()

#rt mode
File obj = open ('abc.txt' , 'rt')
File obj . read (r)
Print ("Output of rt : ", stri)
File obj . close()
>>> Output of rt : Abhay

# write mode
File obj = open ('abc.txt' , 'w')
File obj . write ('DBMS')
File obj . close ()

```

	# read mode
File obj . write	
File obj = open ('abc.txt', 'r')	
stri 2 = file . read()	
Print ("Output of read", stri)	
>>> Output of read, Abhay	

* append mode

```
file obj = open('abc.txt', 'a')
file obj.write('Data structure')
file obj.close()
```

```
file object = open('abc.txt', 'r')
str3 = file obj.read()
```

```
print('Output off append mode:', str3)
file obj.close()
```

»» Output of append mode: Savabh, Datastructure

tell()

```
file obj = open('abc.txt', 'r')
pos = file obj.tell()
```

```
print('tell():', pos)
file object.close()
```

»» tell(): 11, pos

seek()

```
file obj = open('abc.txt', 'r')
str4 = file obj.seek(0, 0)
```

```
str8 = file obj.read(0)
print('The beginning of line is:', str8)
```

Practical 2:

Aim: Demonstrate the use of iterables & iterators

Theory: In Python iterator is an object which implements `__iter__` class which has 2 methods namely `__iter__` & `__next__`. list, tuple, dictionary & set all represents a iterable objects

(Q) WAP using iterable objects for displaying the odd nos. in range 1 to 10.

Algorithm

Step 1: Define a iter() with argument and initialize the value & return that value

Step 2: Define the next() with an argument & compare the upper limit by using a conditional statement

Step 3: Now create an object of the given class & pass the object in the iter method

```

code:
class odd:
    def __iter__(self):
        self.num = 1
        return self
    def next(self):
        if self.num <= 10:
            num = self.num
            self.num += 2
            return num
        else:
            raise StopIteration

```

```

>>> y = count()
>>> z = iter(y)
>>> z.next()
1
>>> z.next()
2
>>> z.next()
3
>>> z.next()
4
>>> z.next()
5
>>> z.next()
6
>>> z.next()
7
>>> z.next()
8
>>> z.next()
9
>>> z.next()
10

```

code:

class power:

```
def __init__(self):  
    self.num = 0  
    return self
```

```
def next(self):
```

```
    if self.P <= 10:  
        num = self.P
```

```
        self.P -= 1
```

```
P0 = 2**num
```

```
print('2**', self.P-1, '=', P0)
```

```
return P0
```

```
else:
```

```
    raise StopIteration
```

```
>>> p = power()
```

```
>>> x = iter()
```

```
>>> n = next()
```

$$2^{**} 0 = 1$$

$$2^{**} 1 = 2 \quad \text{--->}$$

$$n = next()$$

$$2^{**} 2 = 4$$

$$n = next()$$

$$2^{**} 3 = 8$$

$$n = next()$$

$$2^{**} 4 = 16$$

Q2) WAP using iterator for calculating the power of a given no. For instance no. entered is then value calculated should be
 $1, 2^1, 2^2, 2^3, 2^4$

Algorithm: The following steps define the algorithm:

Step 1: Define `iter()` with argument and initialize the value & return the value.

Step 2: Now define `next()` with an argument and compare the upper limit by using conditional statement.

Step 3: Now create an object of the given class & pass this object in the `iter` method.

Q3) ~~WAP using iterable concepts to find the factorial of no. in range, ito~~

Algorithm:

Step 1:- Define an `iter()` with arguments & initialize the value and return the value.

Step 2:- Define the `next()` with arguments and compare the upper limit by using conditional statement.

Q5

Step 3: Now create an object of the given class & pass this object in its method.

Q4] Write a program using iterable concept to display multiple of 2 in range 1 to 10.

Algorithm :

Step 1:- Define a `__iter__()` with argument & initialize the value and return the value.

Step 2:- Define the `__next__()` with an argument and compare the upper limit using `if` statement.

Step 3:- Now create an object of the given class & pass this object in the `iter` method.

~~odd~~ fact:
~~class~~ def - - i der -- (self):
~~def~~ self.f = 1
~~return~~ self

def next (self):
~~if~~ self.f <= 10:

num = self.f

self.f += 1

fac = 1

for i in range(1, num+1):

 fac *= i

Print (self.f-i, '!', fac)

else:

 raise StopIteration

7) fac = fact()

7) n = i der (f)

7) n.next()

1! = 1

n.next()

2! = 2

n.next()

3! = 6

```

## code:
class mult:
    def __init__(self):
        self.m = 1
    return self

    def next(self):
        if self.m <= 10:
            num = self.m
            self.m += 1
            table = 2 ** num
            print("2^", num, " = ", table)
        else:
            raise StopIteration

```

```

>>> m = mult()
>>> n = iter(m)
>>> n.next()
2**1 = 2
>>> n.next()
2**2 = 4
>>> n.next()
2**3 = 6
>>> n.next()
2**4 = 8

```

Practical 3:-

Aim: Demonstrate the use of exception handling

Theory:- An exception is an event which occurs during execution of program which disrupts the normal flow of program. The exception represents the object which represents an error. This object is derived from given class & when Python script raises an exception it must be handled immediately otherwise it will terminate & close the program.

Q] WAP to check the range of the age of the student in a given class & if age does not fall in given range use value error exception otherwise return the value.

Step1:- Define a function which will accept the age of the student from standard input

Step2:- Use 'condn' statement whether to input age falls in range or not so return value

* code
def accept age ():
 age = int(input("Enter your age:")) 28
 if age > 30 or age < 15:
 raise ValueError
 else:
 print ("Your age is ", age)

valid = False
while not valid:
 try:
 age = accept - age
 valid = True
 except ValueError:
 print ("Your age is not in range")

>>> Enter your age: 15

your age is not in range

Enter your age : 32

your age is not in range

Enter your age : 17

your age is 17

code:

While True:

try:

a = int(input("Enter a no."))

Print("Valid no.")

break

except ValueError:

Print("Not a valid no.")

>>> Enter a no: 17.2

Not a valid no.

>>> Enter valid no.: 172

Valid No.

Step 3: Use except with value error to print the message "not a valid range".

Q) WAP to check whether the no. is given class if the no. is a float point use Value Error as Exception.

Algorithm:-

Step 1:- Use try block & accept the input using input() & convert it into integer datatype & subsequently handle the block.

Step 2: Use the except block with exception as ValueError & display appropriate message.

Q) WAP to demonstrate use of zero division error.

Algorithm:

Step 1: Use the try block & accept the input using input() & then convert it into int datatype.

Step 2 - Define a function which with 2 parameters to divide the no. given by user.

Step 3 - Define while loop to check whether the boolean expression holds true

Step 4 - Use except with ZeroDivisionError & print the message

```
def divide(a, b):
    ans = a / b
    return ans
```

```
while True:
    try:
        a = int(input("Enter 1st no.:"))
        b = int(input("Enter second no.:"))
        ans = divide(a, b)
        print('division of', a, 'and', b, 'is', ans)
        break
    except ZeroDivisionError:
        print("Error!")
```

1) Enter 1st no.: 1

2) Enter 2nd no.: 1

Division of 1, 1, is 1

1) Enter 1st no.: 1

2) Enter second no.: 0

error!

```
#CODE1
import re
string = "ALL Might! 1234"
result = re.findall("\d+", string)
result1 = re.findall("\D+", string)
print(result)
print(result1)

#output:
>>> ['1234']
>>> ['ALL Might']
```

✓ m

Practical 4:-

Aim:- Demonstrate the use of regular expression

Theory:- R.E. represents the sequence of character which is mainly used for finding & replacing the given pattern in the string & for this we import re module & common usage of re expression involved foll. functionalities:-

- Searching a given string
- Finding a given string
- Breaking a string into smaller sub string
- Replacing part of string

WAP on R.E. separating numeric & alphabetical values from a string.

Algorithm:-

Step 1:- Now apply string & pattern in.findall() & display the output.

Step 2 :- \d is used for matching all decimal digits whereas D is used to match non-decimal digits.

Q2]. W.A.R.E. for finding match string at the beginning of the sequence.

Algorithm :-

Step 1 :- Import re module & apply a string in pattern.

Step 2 :- Use `re.search()` with " \A Python " and string as 2 parameters.

Step 3 :- Now display the output.

Step 4 :- Now use `if` statement to check the match is found or not.

```
code:2
import re
string = "Python is an Amazing language" *32
result = re.search('IAmazing', string)
print(result)
if result:
    print('Match found')
else:
    print('Match not found')

# output
>>> m = re.match(r'IAmazing', string)
>>> m
match object: span=(0, 6);  
match='python'>
```

>>> match found

QD3:

import re

Li = ['98704507687', '8169260459', '9011869675',
"9920105706", "7030734259"]

for

element in Li :

result = re.match("[8-9] \{1\} [0-9] \{9\}", element)

if result :

Print ("Correct mobile")

Print (result.group(0))

else :

Print ("Incorrect mobile no:")

Output :

>>>

Correct mobile no.

98 6765543210

Correct mobile:

816 9260459

Correct mobile:

9011869675

Incorrect mobile

Correct mobile.

WAP on R.E. to check whether the given mobile no. starts with 8 or 9 & total length of digits should be 10.

Algorithm :

Step 1:- Import re module & apply a s.t.y of mobile nos.

Step 2:- Now use for conditional statement to find if the starts with 8 or 9 and the total no.s. should be of 10. Use match inside for statement to find the "match" in given string.

Step 3:- Use if conditional statement to know whether we have a match or not so if we have use group(1) to display the output and if false display incorrect mobile no.

Q4] W.R.E. for extracting a word from given string along words between words.

Algorithm :-

Step 1:- Import re module & apply a string

Step 2:- Use find all() to extract a word from the string

Step 3:- Use "\w*" to extract word along with space & use "\w+" to extract word without space.

Step 4:- Now display the output.

Q5] W.A.R.E. for extracting first & last word from a string

Step 1:- Import re module and apply a string

Step 2:- Use find all() in which use "\w+" as one parameter to find first word of string then use "\w+\\$" as parameter to find last word of string

Step 3:- Now display ...

#10de4:

```
import re  
string = "Python is Sugoi, subaashi"  
result1 = re.findall("\w+", string)  
result2 = re.findall("\w*", string)  
print(result1)  
print(result2)
```

34

output:

```
>>> ['Python', ' ', 'is', ' ', 'sugoi', 'subaashi']  
['Python', 'is', 'sugoi', 'subaashi']
```

#10DE5:

```
import re  
string = 'Python is important'  
result = re.findall("\w+", string)  
result1 = re.findall("\w+\$+", string)  
print(result)  
print(result1)
```

output

```
>>> ('Python')  
>>> ['important']
```

code 6

```
import re  
string = 'Amit: 2019-12-24 12:23:45 - [d2y-1d82y] shig'  
result = re.findall('(\d{2}y-\d{2}y) shig')  
print(result)  
  
# output  
→> ['24-12-2019']
```

Q6] W.R.E for extracting the date information dd-mm-yyyy by using the find all() where the string has full format Amit 2011-24-120-2019

Algorithm :-

Step 1: Import re module & apply str

Step 2: Use findall method and use 'dd' - 'mm' - 'yyyy' as a parameter.

Step 3: Now display the output.

Q7] W.A.R.E. for extracting :-

- ① Username from email-id
- ② Host name from email-id
- ③ Both host name & username from email-id

Algorithm

Step 1: Import re module & apply a str

Step 2:- Use findall() to find username & host name & both of email id

Step 3: Use "lwt" for user name use "t\w+\w+"
for host name or user name
"[\w+-]+\w+" for both as
parameter in find all C:\

Step 4:- Display the output

Code :-

```
import re
string = 'abc @ tsc.edu'
result1 = re.findall("^\w+", string)
result2 = re.findall("\t\w+\.\w+\$", string)
result3 = re.findall("[\w\.-]+\.", string)

print(result1)
print(result2)
print(result3)
```

Output

```
>>> ['abc']
>>> ['tsc.edu']
>>> ['abc', 'tsc.edu']
```

07/01/2022

Creation of parent window
From tkinter import *

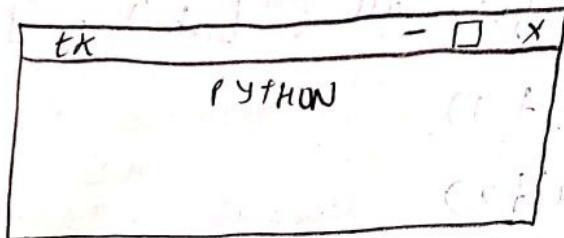
root = Tk()

l = Label(root, text="Python")

l.pack()

root.mainloop()

Output:



2 label, attributes

From tkinter import *

root = Tk()

l = Label(root, text='Python')

l.pack()

l1 = Label(root, text="CS", bg="grey", fg="black", font=10)

l1.pack(side=LEFT, padx=20)

l2 = Label(root, text="CS", bg="light blue", fg="black", font=20)

(~~l2.pack(side=LEFT, pady=30)~~)

l3 = Label(root, text="CS", bg="yellow", fg="black", font=10)

~~(l3.pack(side=TOP, ipadx=40))~~

= Label(root, text="CS", bg="orange", fg="black", font=10)

~~(l4.pack(side=TOP, ipadx=50))~~

root.mainloop()

Practical 5:-

37

Topic: GUI components

Step 1: Use the tkinter library for importing the features of text widget.

Step 2: Use the create() method using the TK()

Step 3: Create a variable using the widget label and use the text method

Step 4: Use the mainloop() for triggering of correspondingly above mentioned events

#2 Step 1: Use the tkinter library for importing the features of text widget.

~~Step 2: Create a variable from the text method & position it on the parent window.~~

Step 3: - Use the pack() along with the obj created from the text() several parameters

1) side = LEFT, padx = 20

2) side = LEFT, pady = 30

3) side = TOP, i) padx = 40

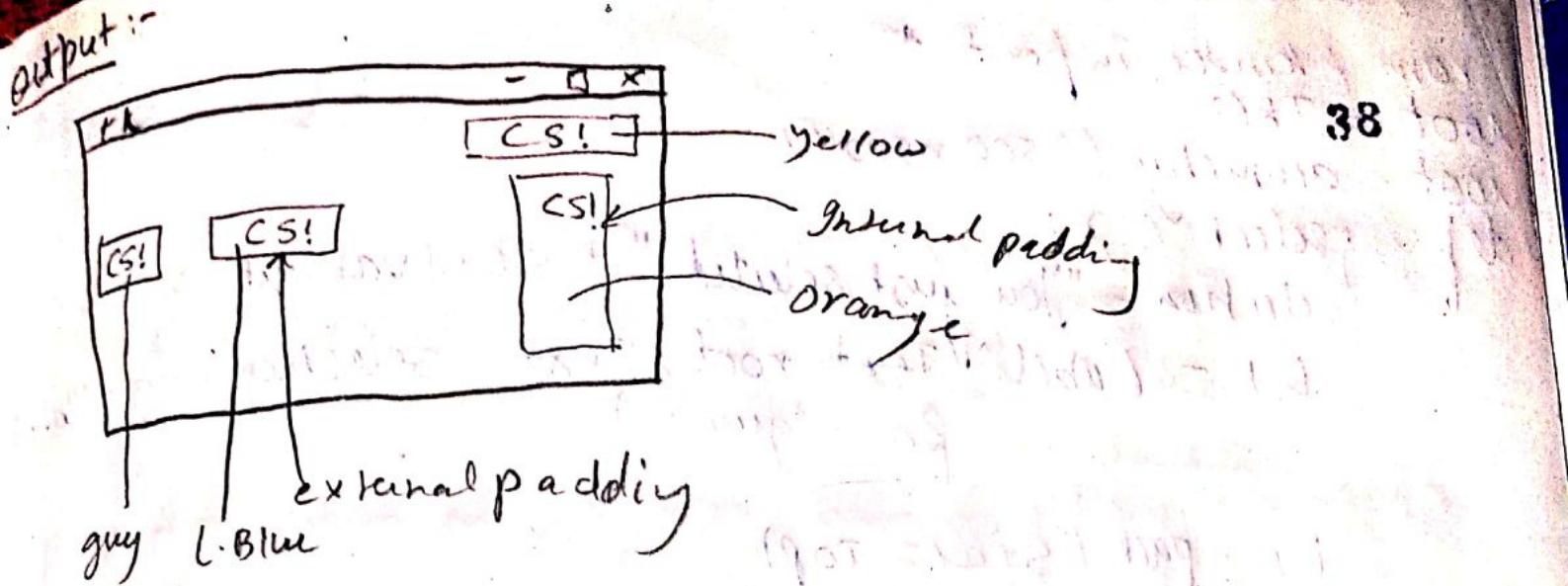
4) side = TOP, i) pady = 50

58

Step 4:- Use the mainloop() for the triggering
of the corresponding events

Step 5:- Now repeat above steps with
label() which takes the foll arguments:

- 1) Name of the parent Window
- 2) Text attribute which defines the string
- 3) The bg color
- 4) The foreground bg & then use the
pack()



#= Radiobutton
from tkinter import *
root := Tk()
root.geometry ("500x500")
def select ():
 selection = "you just selected " + str (var.get ())
 l1 = Label (root, Text = selection, bg = "white",
 fg = "green")
 l1 . pack (side = top)
var = StringVar ()
t1 = listbox
t1 . insert (0, "option1")
t1 . insert (1, "option2")
t1 . pack (anchor = N)
d1 = Radiobutton (root, text = "option 1", variable = var,
 value = "option1", command = select)
d2 = Radiobutton (root, text = "option 2", variable = var,
 value = "option2", command = select)
d2 . pack (anchor = N)
root . mainloop()

Aim :- GUI components

"What" #1 GUI

Step1:- Import relevant objs from tkinter library
create an obj from parent window.

Step2:- Use the parent windows' object along with
the geometry() declaring specific pixels
size of the parent window.

Step3:- Now define a function which tells the
users about the given selection made from
multiple options available.

Step4:- Now define the parent window &
define the option with control variable

~~Step5~~:- Use the listbox() & insert options on the
parent window along with the pack()
with specifying other attributes.

Step6:- Create an object from radio button which
will take following arguments
• parent window object
• var variable which will take the value
option no 1, 2, 3, ...
• Variable arguments
• Corresponding value to trigger the
function declared.

Step 7:- Now call pack() for radio obj so as to specify the argument using anchor attribute.

Step 8:- Finally make use of the mainloop() along with parent object.

#2 [Scoll bar:-]

Step 1: Import relevant methods from tkinter lib.

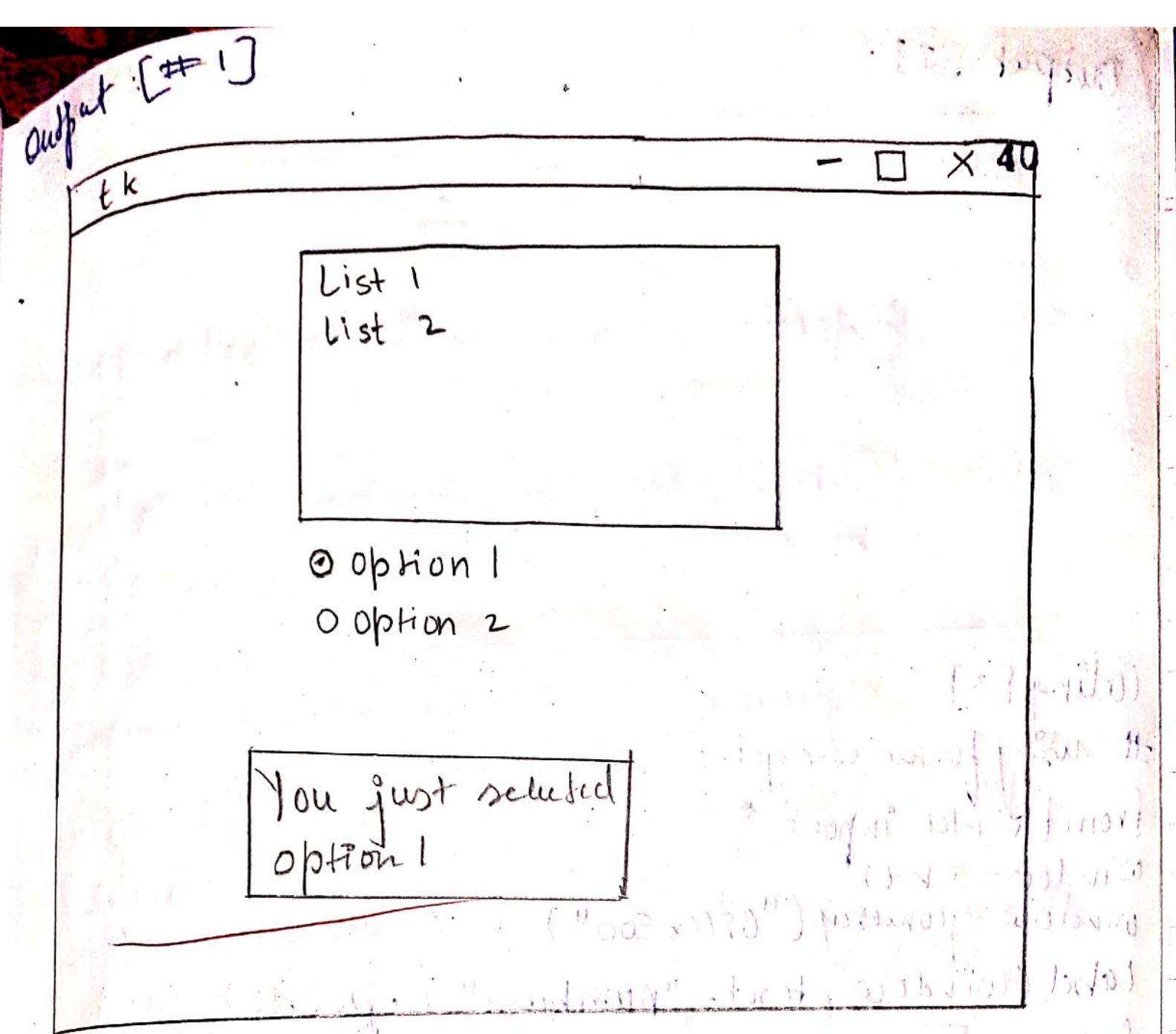
Step 2:- Create a parent obj corresponding to the parent window.

Step 3: Use the geometry() for laying of the windows.

Step 4: Create an obj & use the scrollbar()

Step 5: Use the pack() along with the scrollbar()

Step 6: Use the mainloop() with the parent obj.



#Coding [2]

#scrollbar

from tkinter import *

root=tk()

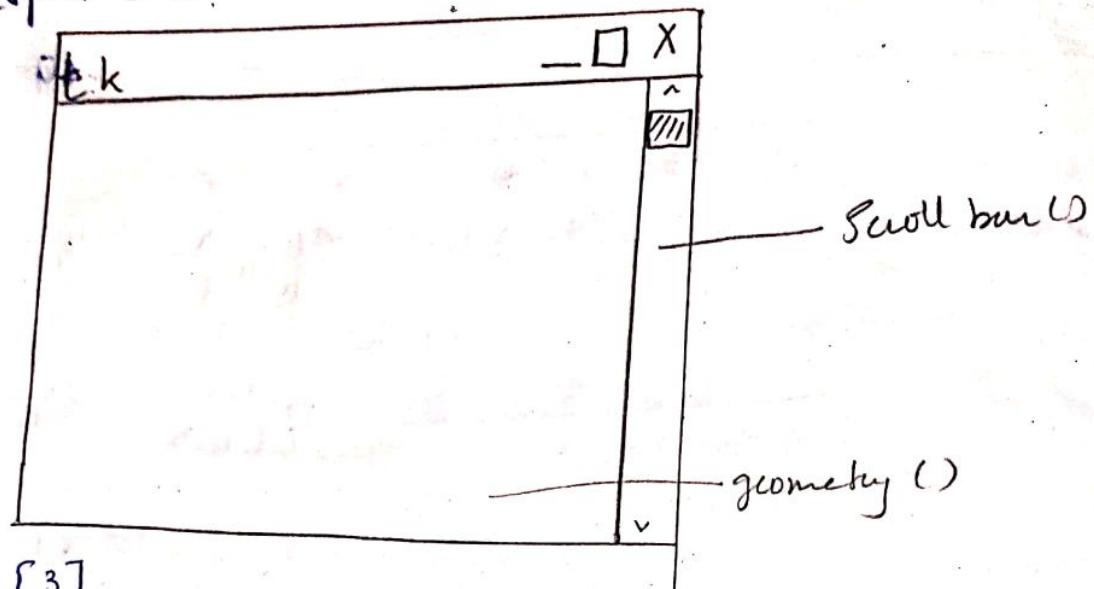
root.geometry ("500x500")

s=scrollbar()

s.pack(side="right", fill="y")

root.mainloop()

Output [2]:



Coding [3]

```
# Using frame widget
from tkinter import *
window = Tk()
window.geometry("680x500")
label(window, text="numbers:").pack()
frame = Frame(window)
frame.pack()
listNodes = Listbox(frame, width=20, height=20, font="Times New Roman", 10)
listNodes.pack(side="left")
scrollbar = Scrollbar(frame, orient="vertical")
scrollbar.config(command=listNodes.yview)
scrollbar.pack(side="right", fill="y")
for x in range(100):
    listNodes.insert(END, str(x))
window.mainloop()
```

#3: [Using frame Widget]

41

Step 1:- Import the relevant libraries from tkinter.

Step 2:- Create an corresponding obj of the parent window.

Step 3:- Use the 'geometry ()' with pixel size ("680x500") or any other suitable pixel value

Step 4:- Use the label widget alongwith the parent obj created & subsequently use the pack()

Step 5:- Use the frame widget along with the parent object created and else pack()

Step 6:- Use the listbox () alongwith the attr. like width, height, font. Do create a listbox() obj use pack() for the same.

Step 7:- Use the scrollbar () with an obj use the attr. of vertical. Then configure the same with obj created from scrollbar () & use pack()

Step 8:- Trigger the events with help of mainloop()

4:-

Step1:- Import relevant methods from tkinter library.

Step2:- Define the obj corresponding to parent window.

Step3:- Now define frame obj from the method & place it onto the parent window.

Step4:- Create another frame obj termed as left frame & put it on the left side of the parent window.

Step5:- Similarly define the Right Frame & subsequently define the button obj placed onto the given frame with the attr. as text, active bg & foreground.

Step6:- Now use the pack() along with the side attr.

Step7:- Similarly create a button obj corresponding to the Modify operation put it into the frame obj on side = "right".

Step8:- Create another button obj & place it on ADD. Right frame & label the button.

Coding : [#=4]

```

from tkinter import *
window = Tk()
window.geometry("680x500")
frame = Frame(window)
frame.pack()
leftframe = Frame(window)
leftframe.pack(side="left")
rightframe = Frame(window)
rightframe.pack(side="right")
b1 = Button(Frame, text="Select", activebackground="red",
            fg="blue")
b2 = Button(Frame, text="Modify", activebackground="yellow",
            fg="black")
b3 = Button(Frame, text="ADD", activebackground="blue",
            fg="red")
b4 = Button(Frame, text="Exit", activebackground="red",
            fg="green")
b1.pack(side="left", padx=20)
b2.pack(side="right", padx=30)
b3.pack(side="bottom", pady=20)
b4.pack(side="Top")

```

Coding : [#4]

42

```
from tkinter import *
window = Tk()
window.geometry("680x500")
frame = Frame(window)
frame.pack()
leftframe = Frame(window)
leftframe.pack(side="left")
rightframe = Frame(window)
rightframe.pack(side="right")
b1 = Button(Frame, text="Select", activebackground="red",
            fg="blue")
b2 = Button(Frame, text="Modify", activebackground="yellow",
            fg="black")
b3 = Button(Frame, text="ADD", activebackground="blue",
            fg="red")
b4 = Button(Frame, text="Exit", activebackground="red",
            fg="green")
b1.pack(side="left", padx=20)
b2.pack(side="right", pady=30)
b3.pack(side="bottom", padx=20)
b4.pack(side="Top")
```

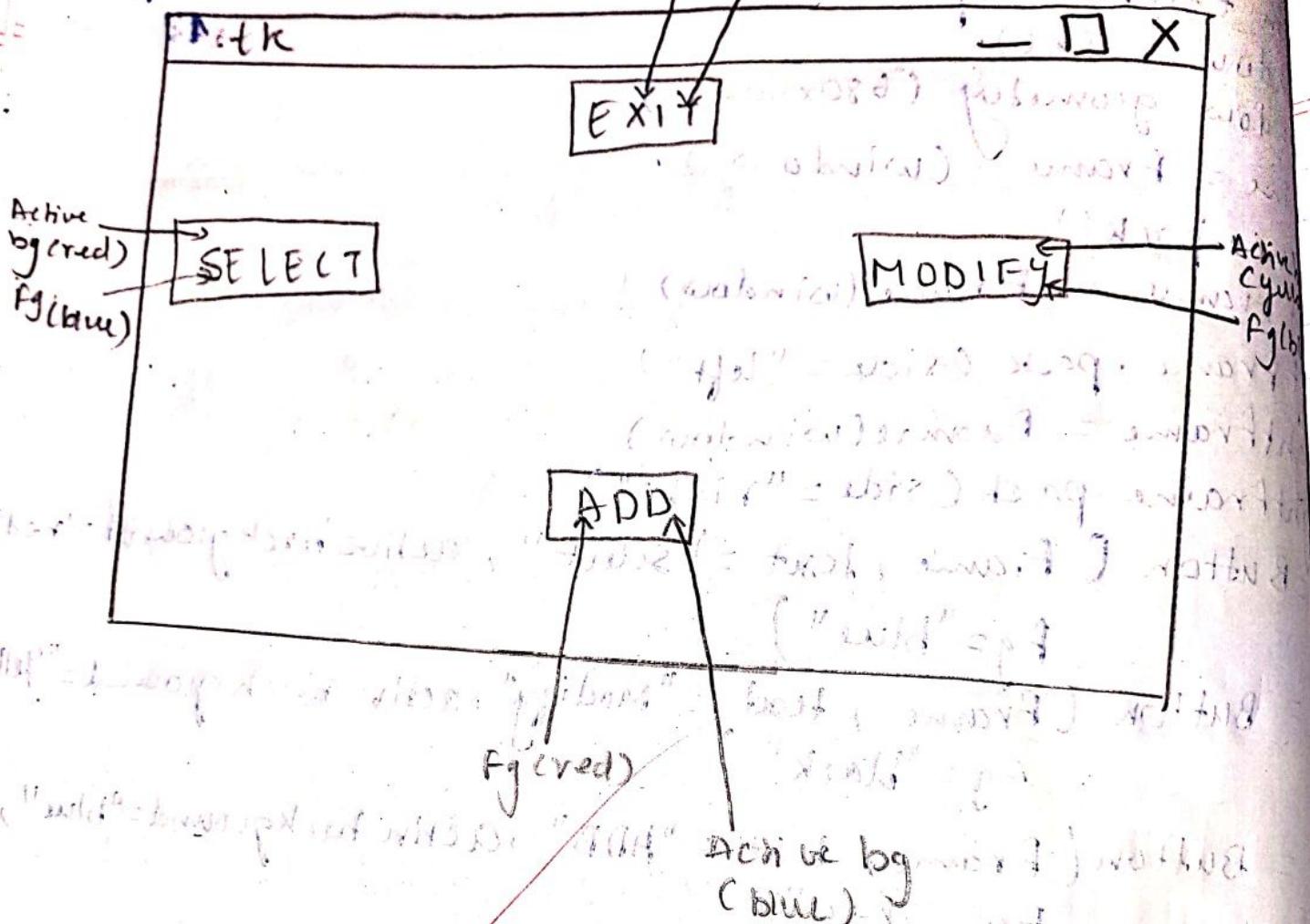
Coding : #4

```

from tkinter import *
window = Tk()
window.geometry("680x500")
frame = Frame(window)
frame.pack()
leftFrame = Frame(frame)
leftFrame.pack(side="left")
rightFrame = Frame(frame)
rightFrame.pack(side="right")
b1 = Button(Frame, text="Select", activebackground="red",
            fg="blue")
b2 = Button(Frame, text="Modify", activebackground="yellow",
            fg="black")
b3 = Button(Frame, text="ADD", activebackground="blue",
            fg="red")
b4 = Button(Frame, text="Exit", activebackground="red",
            fg="green")
1.pack(side="left", padx=20)
2.pack(side="right", padx=30)
3.pack(side="bottom", pady=20)
4.pack(side="Top")

```

Output 4:-



Step 9:- Add another button & put it on top of frame & label is as EXIT.

Step 10:- Use the pack() simultaneously for all the objects. finally use the mainloop().

Practical 5[C]

Aim:- GUI Components

Algorithm:- #1 [Message box]

Step1:- Import the relevant methods from tk library.

Step2:- Import tk messagebox

Step3:- Define a parent window object alongwith parent window.

Step4:- Define a function which will use tkmessagebox with showinfo() alongwith info window attribute.

Step5:- Declare a button with parent window object alongwith the command attr.

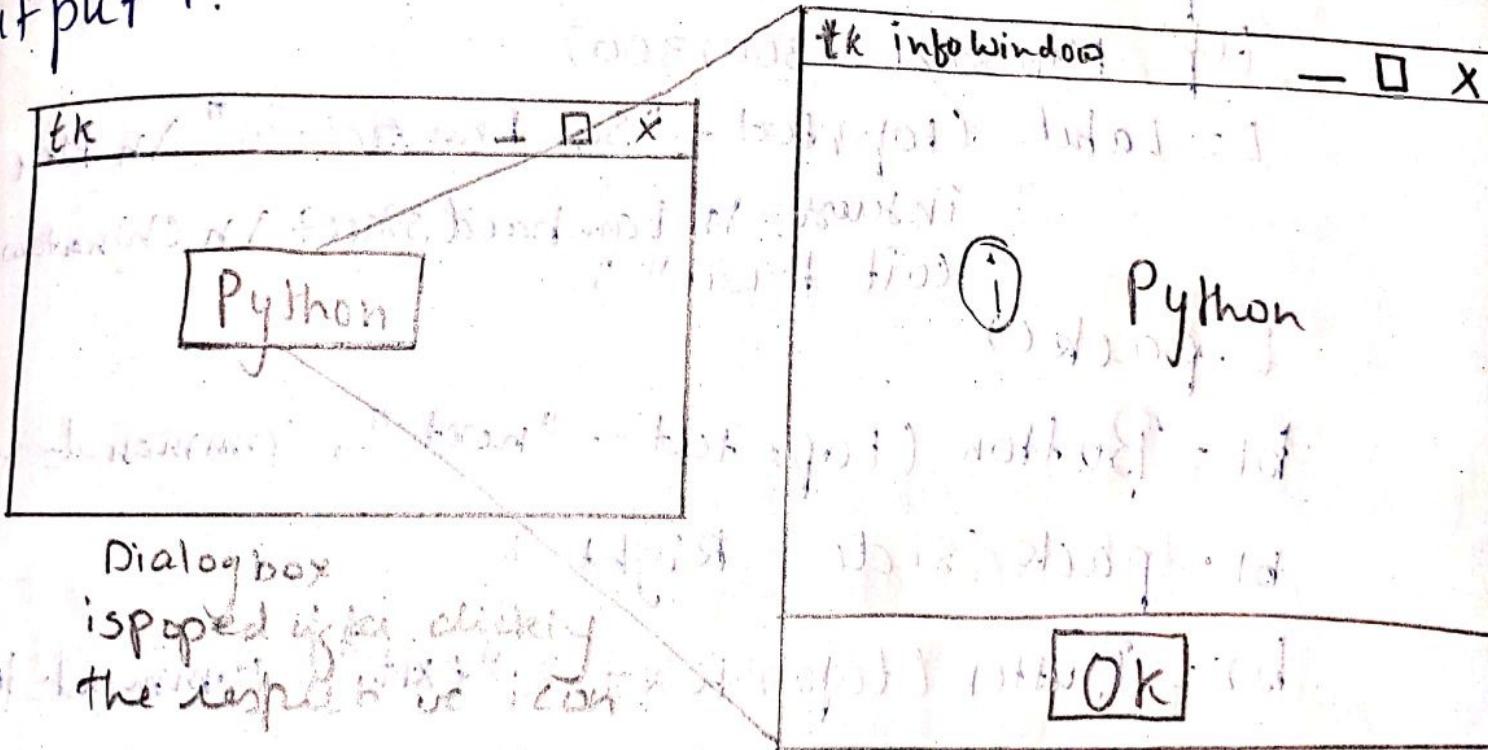
Step6:- Place the button widget onto the parent window & finally call mainloop() for triggering of events called above.

```

coding : [Ex 1]
# message box
from tkinter import *
import tkinter.messagebox
root = Tk()
def function():
    messagebox.showinfo("info window", "Python")
    b1 = Button(root, text="Python", command=function)
    b1.pack()
root.mainloop()

```

Output :-



Coding : 2

Multiple Window
Different button (relief " ")

from tkinter import *

root = Tk()

root . min size (300, 300)

def main ():

top = Tk()

top . config (bg = "black")

top . title ("Home")

top . min size (300, 300)

L = Label (top, text = "San Francisco" \n Place
in west in Lombard street in Chinatown
city town")

L . pack ()

b1 = Button (top, text = "next", command = second)

b1 . pack (side = Right)

b2 = Button (top, text = "exit", command = exit)

b2 . pack (side = LEFT)

top . mainloop ()

def second () :

top 2 = Tk ()

top 2 . config (bg = "orange")

#2 [Multiple Windows ; Different button (Relief)]

Step1:- Import the relevant method from the `tkinter` library along with parent window object declared.

Step2:- Use parent window object alongwith `minsize` for window size.

Step3:- Define a function `main`, declare parent window object & use `config()`, `title()`, `minsize()`, `label()` as well as `button()` & use `pack()` & `mainloop()` simultaneously.

Step4:- Similarly define the function `second` & use the attributes accordingly.

Step5:- Declare another function `button` alongwith parent obj & declare button with attr. like FLAT, RAISED, GROOVE, SUNKEN alongwith relief widget.

Step6:- Finally called the `mainloop()` for event driven program.

b5
top 3
def

top 2 = Tk()
top 2 . title ("About us!")
top 2 . minsize (300, 300)
L = Label (top 2 , text = "Created by Abhay S.
In for more details contact (the
our official account")
L . pack ()
b3 = Button (top 2 , text = "Prev" , command =
mainloop)

b3 . pack (side = LEFT)

b2 = Button (top 2 , text = "exit" , command =
quit)

b2 . pack (side = RIGHT)

def button ():

top 3 = Tk()
top 3 . geometry ("300x300")

b1 = Button (top 3 , text = "Flat button" ,
relief = FLAT)

b1 . pack ()

b2 = Button (top 3 , text = "Groove button" ,
relief = GROOVE)

b2 . pack ()

b3 = Button (top 3 , text = "Raised button" ,
relief = RAISED)

b3 . pack ()

b4 = Button (top 3 , text = "Sunken button" ,
relief = SUNKEN)

b4 . pack ()

```

    b5 = Button (top 3) text = "Ridge button", relief = RIDGE)
    .pack()
    mainloop()

top 3. terminate()
def b5 = quit()

b5 = Button (root, text = "TOUR DETAILS", command =
main)

b5.pack()

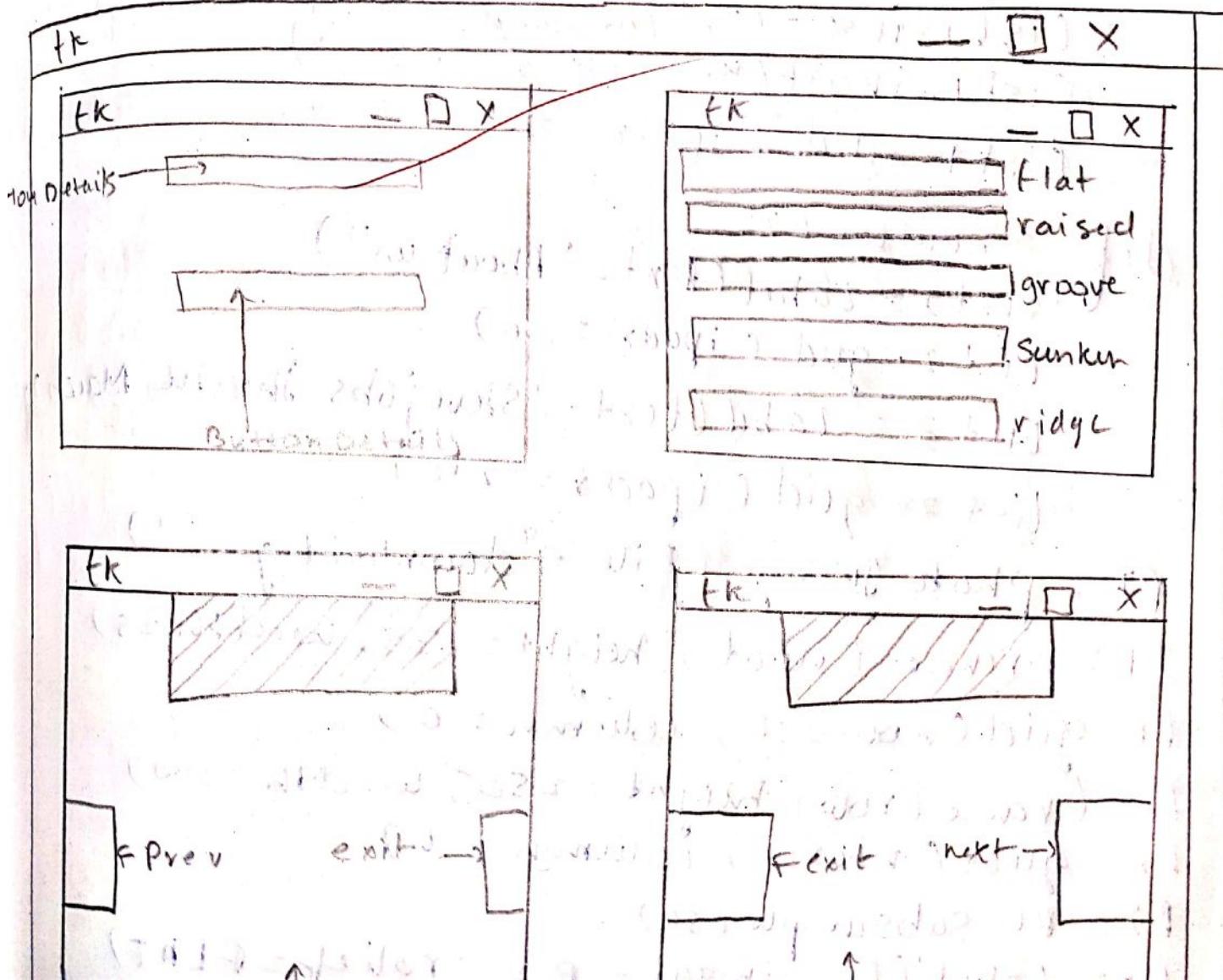
b6 = Button (root, text = "Button DETAILS",
command = button)

b6.pack()

root.mainloop()

just put #2

```



Coding : [# = 1]

```
from tkinter import *
root = Tk()
root.config(bg = "grey")
def finish():
    message_box = askokcancel("Warning", "This will"
                               + " quit the program")
def info():
    list1 = Listbox()
    list1.insert(1, "Co.name : Apple")
    list1.insert(2, "Products : iphone")
    list1.insert(3, "Language : SWIFT")
    list1.insert(4, "OS : iOS")
    list1.grid(ipadx = 30)
```

def aboutus():

```
List2 = Label(text = "About us")
list2.grid(ipadx = 30)
```

```
List3 = Label(text = "Steve jobs died in March 2011")
list3.grid(ipadx = 24)
```

P1 = PhotoImage(file = "download.gif")

F1 = Frame(root, height = 35, width = 5)

F1.grid(row = 1, column = 0)

F2 = Frame(root, height = 250, width = 500)

P2 = P1.subsample(54)

h1 = Label(F1, image = P2, relief = FLAT)

Practical No 5, [D]

47

Aim :- Gui Components

Algorithm : ++ .

Step 1:- Import relevant method from Tkinter library

Step 2:- Create Parent Window obj & use the config method alongwith background colour attribute specified.

Step 3: ~~Define a function finish with the message box widget which will display a message [i.e. A warning Message & subsequently terminate the program] .~~

Step 4:- Define a function info use a listbox widget alongwith the object of the same. Use the listbox object alongwith insert() & insert the s and & finally use the grid() with ipadx attr.

Step 5: Define a function about us with label widget & text attribute & subsequently use the grid()

Step 6:- Use the Photoimage widget with file
as filename with gif attr.

Step 7:- Create a frame object along with
the Frame() along with parent
window object height & width specify
it subsequently use the grid()
with row & column attribute spec-

Step 8:- Similarly, Create another frame
object as declared by step 7

Step 9:- Create another object S1 sub-

Step 10:- Use label widget along with the frame
object relief attribute as subsequent
else use grid()

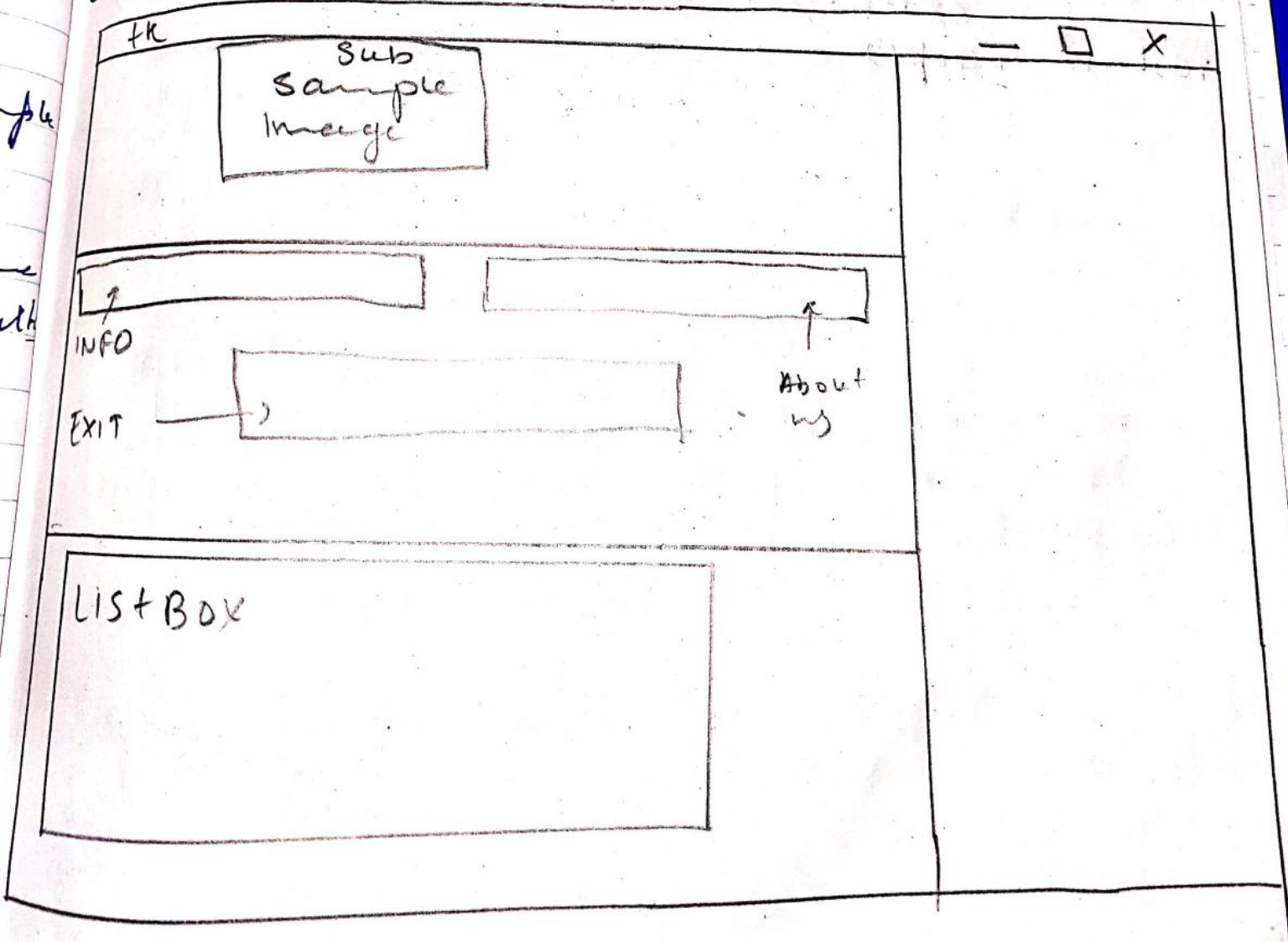
Step 11:- Now Create button obj dealing
with different section of frame

```

    f1 = Frame()
    l1 = Label(f1, image=pi, relief=SUNKEN)
    b1 = Button(f1, text="Information", relief=RAISED,
                command=info)
    b1.grid(row=1, column=0, padx=20, pady=15)
    b2 = Button(f1, text="About us", relief=SUNKEN,
                command=about_us)
    b2.grid(row=1, column=1, padx=5)
    b3 = Button(f1, text="Exit", relief=RAISED,
                command=finish)
    b3.grid(row=1, column=2, padx=5)
    f1.grid(row=2, column=1, ipadx=15)
    root.mainloop()

```

Output : #



Coding:- [] : [Human Face]
from tkinter import *

root = Tk()

C = Canvas(root, width=500, height=500)

c.pack()

face = C.create_oval(50, 50, 350, 350, outline="black", fill="yellow")

eye1 = C.create_oval(125, 125, 175, 175, fill="black")

eye2 = C.create_oval(125, 125, 275, 175, fill="black")

mouth = C.create_arc(125, 225, 275, 275, start=0, extent=-180, width=5, fill="red")

root.mainloop()

Practical 6:-

49

Aim :- WAP to draw a human face with GUI

Algorithm:-

Step 1:- Import relevant methods from tkinter library

Step 2:- Create a object corresponding to the parent window from TK()

Step 3:- Create an obj from Canvas() to place it onto parent window along with height & width

Step 4:- Now use pack() for positioning of widget onto the parent window.

Step 5:- Now create an object face & use object .Create oval() with 10-ordinates 50, 50, 350, 350 & outline = 'black', fill = 'yellow' as an attribute to create face.

Step 6:- Now create eye-1 object & again use obj.create oval() with appropriate coordinates along with fill as attribute to create left eye.

Step 2:- Now repeat the same step to create right eye

Step 3:- Create an object Mouth i.e. up object. Create an object with appropriate co-ordinates, start extent = -180 & fill = "red", width 5 as attribute to create mouth.

Step 4:- Finally use the methods

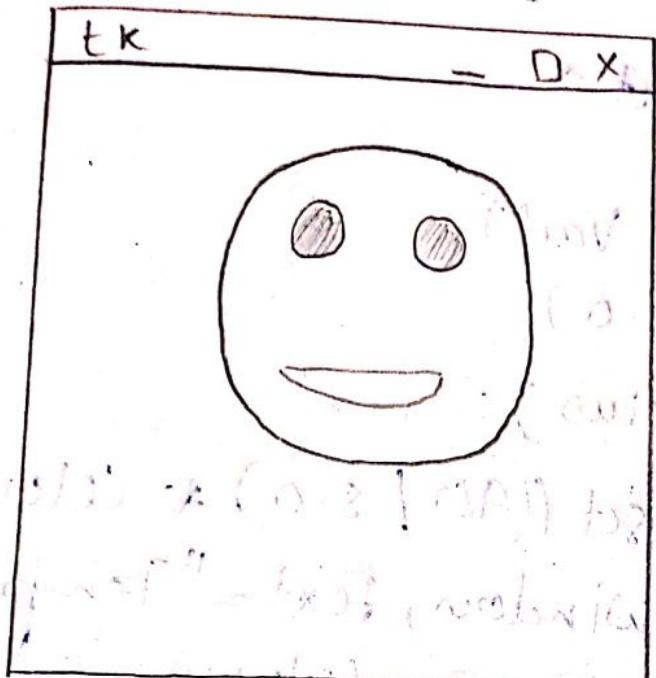
of Graphics class to draw the mouth. This will be filled with red color.

This will be the final output.

Outpus

Output

50



~~#2~~ Coding [Conversion of Farenheit From Celsius]

```
From tkinter import *
window = Tk()
Farenheit = DoubleVar()
Farenheit.set(32.0)
def convert(celsius):
    Farenheit.set((9.0 / 5.0) * celsius + 32)
    L1 = Label(window, text="Temp in Celsius")
    L1.grid(row=0, column=0)
    E = Entry(window, textvariable=celsius)
    E.grid(row=0, column=1)
    celsius = IntVar()
    L2 = Label(window, textvariable=farenheit)
    L2.grid(row=2, column=0, columnspan=2)
    B = Button(window, text="Calculate",
               command=lambda: convert(celsius.get()))
    B.grid(row=1, column=0, columnspan=2)
window.mainloop()
```

Conversion of fahrenheit from Celsius

Step1:- Import relevant method from tkinter library

Step2:- Make a parent Window & decline a function for conversion of one scale to another.

Step3:- Define fahrenheit & Celsius & give them variable which hold value

Step4:- Use Label Widget & place it on Parent window , its also use text to specify the type of conversion.

Step5:- Use grid() & specify rows & column attributes.

Step6:- Use Entry Widget & and use text variable foll the grid()

Step7:- Use button widget & grid() specifying rows & column

Step8:- Use mainloop() to terminate prog.

Output 2 :-

tk → X

Temp in Celsius

12

CONVERT

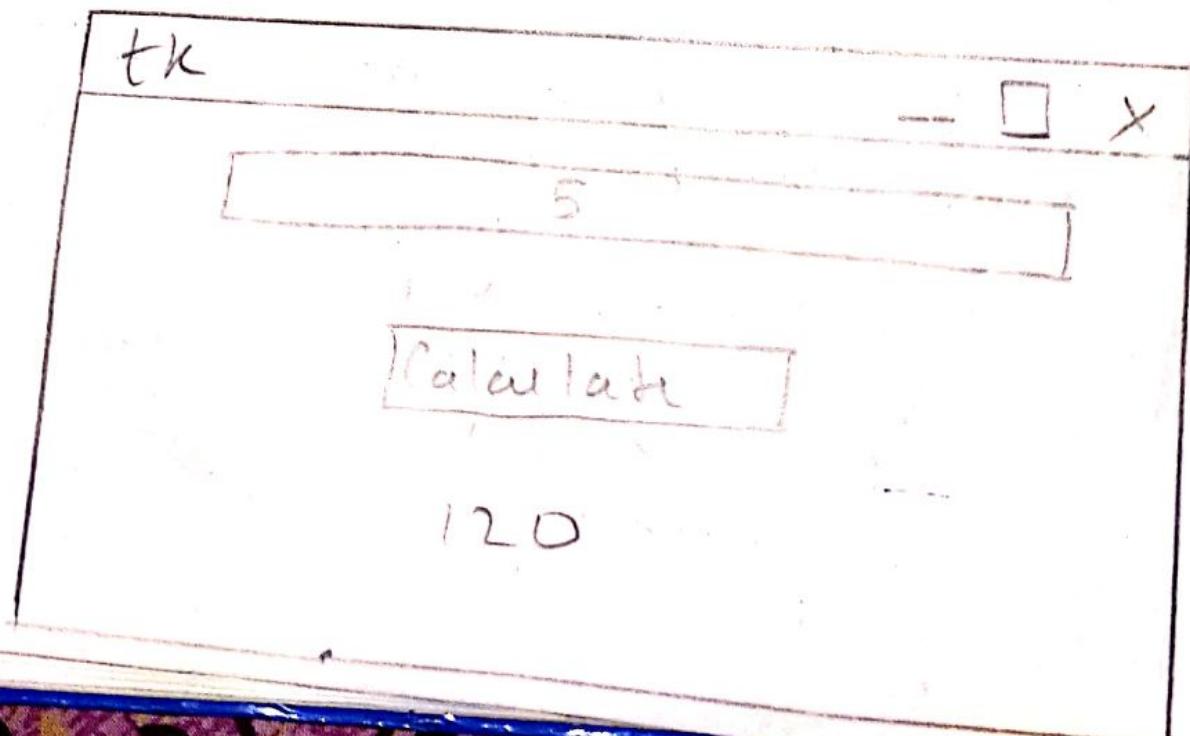
53.6

Coding :-

```
from tkinter import*
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n-1)

root = Tk()
def calculate():
    result = factorial(int(entry.get()))
    info.config(text=result)
entry = Entry(root)
entry.pack()
bt = Button(root, text="Calculate", command=calculate)
bt.pack()
info = Label(root, text="factorial")
info.pack()
root.mainloop()
```

Output:-



Practical 7 :-

5.3

Aim:- WAP to find factorial of no. & use arithmetic operations on 2 nos. using GUI.

Algorithm:-

Step 1:- Import relevant methods from tkinter library

Step 2:- Now Define a function factorial to calculate factorial using recursive function

Step 3:- Define another function to calculate to call factorial function.

Step 4:- Now create an obj with entry () And use pack () to position the obj

Step 5:- Now create a button () along with command attr & create a label () to show output.

Step 6:- Finally use the mainloop ()

88
#2 Code 2:- Arithmetic. opern on 2 nos.

Step 1:- Import Relevant Methods from Tkinter library

Step 2:- Now Create an obj corresponding to parent window

Step 3:- Now define a function calculate to carry out operation.

Step 4:- Now create obj with label 1) as Num 1 & Num 2 and use grid to place it onto parent window.

Step 5:- Create objects with entry() to take input from user()

Step 6:- Now initialise u as integer using int()

Step 7:- Now Create 4 objects with Radiobutton() to choose any one of Arithmetic operators & use grid

```

loading : # 2
from tkinter import *
def calculate():
    if int(u.get()) == 1:
        res = int(e1.get()) + int(e2.get())
        l3.config(text=res)
    elif int(v.get()) == 2:
        res = int(e1.get()) - int(e2.get())
        l3.config(text=res)
    elif int(u.get()) == 3:
        res = int(e1.get()) * int(e2.get())
        l3.config(text=res)
    else:
        res = int(e1.get()) / int(e2.get())
        l3.config(text=res)

```

```

root = Tk()
l1 = Label(root, text="Enter a no")
l1.grid(row=0, column=0)
e1 = Entry(root)
e1.grid(row=0, column=1)
l2 = Label(root, text="Enter 2 no")
l2.grid(row=1, column=0)
e2 = entry(root)
e2.grid(row=1, column=1)
v = IntVar()

```

r1 = Radiobutton (root, text = "Add", variable=4, value=1)

r1.grid(row=2, column=0)

r2 = Radiobutton (root, text = "Sub", variable=4, value=2)

r2.grid(row=2, column=1)

r3 = Radiobutton (root, text = "Multiply", variable=4, value=3)

r3.grid(row=2, column=2)

r4 = Radiobutton (root, text = "Div", variable=4, value=4)

r4.grid(row=2, column=3)

B = Button (root, text = "Calculate", command=calculate)

B.grid(row=3, column=1, columnspan=2)

L3 = Label (root)

L3.grid(row=4, column=1)

root.mainloop()

Output:-

TK

Enter No1:

Enter No2:

ADD SUB MULTIPLY DIV

Calculate

2.0

Step 8:- Now create an object with buttons along with command attribute to carry out the arithmetic operator of user's choice

Step 9:- Now Create a Object with label() to show input()

Step 10:- Finally use the mainloop()

Practical No:- 8

Aim:- Demonstrate the use of Socket and server, Client programs

Algorithm :-

Step 1:- Import the socket module with relevant methods

Step 2:- Define a function as Server() to get host name

Step 3:- Now get value for port variable to initialize port func above to 1234

Step 4:- Use Socket () to get instance

Step 5:- Now use bind () function to be bind host address & port together to configure how many client the screen can list simultaneously

Step 6:- Use accept () to accept new connection

Step 7:- Now print the address

Code:- Server

56

```
import socket  
def server_program:  
    host = socket.gethostname()  
    port = 5000  
    server_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)  
    server_socket.bind((host, port))  
    server_socket.listen(2)  
    conn, address = server_socket.accept()  
    print("Connection from: " + str(address))  
    while True:  
        data = conn.recv(1024).decode()  
        if not data:  
            break  
        print("from connected user: " + str(data))  
        data = input("→")  
        conn.send(data.encode())  
    conn.close()
```

Output :- \$ python3 socket_server.py

connection from: ('127.0.0.1', 57822)

from connected user: Hi

→ Hello

from connected user: How are you?

→ Good

from connected user: Are you?

→ Ok then, Bye!

```
# Code 2  
= import socket  
def Client_prog:-  
    host+= socket.gethostname()  
    port= 5000  
    Client_socket= socket.socket()  
    Client_socket.connect((host, port))  
    message= input("→")  
    while message.lower().strip() != 'bye':  
        Client_socket.send(message.encode())  
        data= Client_socket.recv(1024)  
        print("Received from server", data)  
        message= input("→")  
    Client_socket.close()
```

Output:

```
$ Python 3.6 socket-client.py  
→ Hi  
→ Received from server: Hello  
→ How are you?  
→ Received from server: Good  
→ Awesome.  
→ Received from server: Ok then, Bye!  
→ bye.
```

loop as well to
stream stream ?

the program :-

client program :-

at model to import
at are relevant

function client - program
be give port a value

Practical 9:-

Aim:- Demonstrate the use of database connectivity

Algorithm:-

Step 1:- Import gg lite 3 module to import relevant methods

Step 2:- Now initialise a variable conn to connect by using connect() to a new database using extension.db

Step 3:- Now initialise a variable to connect cursor()

Step 4:- Now use cur.execute() to create a table insert values into tables use DML,DDL, statements to manipulate the data in this database.

Step 5:- Use fetch all() to show the output.

Step 6:- Use commit to save all changes

Step 7:- Use close() to terminate the program.

code in Shell Environment :-

```
>>> import sqlite3  
>>> conn = sqlite3.connect("Student.db")  
>>> cur = conn.cursor()  
>>> cur.execute('create table student (roll_no int(5),  
      primary key, name varchar(50) not null, class varchar(50),  
      dob date)')  
<sqlite3.cursor object at 0x0322EBED>  
>>> cur.execute('insert into students values (101, "Abhay",  
      "Vasai", "FyCS", "24/01/2002")')  
<sqlite3.cursor object at 0x0322EBED>  
>>> cur.execute('insert into students values (102, "Venu",  
      "Borivali", "FyCS", "02/08/2002")')  
<sqlite3.cursor object at 0x0322EBED>  
>>> cur.execute("Select * from student")  
<sqlite3.cursor object at 0x0322EBED>  
>>> cur.fetchall()  
[(101, 'Abhay', 'Vasai', 'FyCS', '24/01/2002')]  
(102, "Venu", "Borivali", "FyCS", "02/08/2002)]
```

>>> cur.execute('Update student set dob = "13/01/1998" where roll-no = 101')

< sqllite3.cursor object at 0x0322EBE>

>>> cur.execute('Select * from student where address = "Kandivali"')

< sqllite3.cursor object at 0x0322EBE>

>>> cur.fetchall()

[C101, 'Abhay', 'Wasai', 'FyCS', 13108]

>>> cur.execute('commit')

< sqllite3.cursor object at 0x0322EBE>

>>> cur.close()

~~not
os~~