

```

fileobj = open ("abc.txt", "w")
fileobj.write ("science subjects" + "\n")
fileobj.write ("physics / chemistry / n states \n")
fileobj.close()

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

>>> ('The output of read mode : ', 'science subject' + 'physics'
      + 'chemistry \n maths\n')
# readline()
fileobj = open ("abc.txt", "r")
str1 = fileobj.readline()
print ("The output of readline method = ", 'science'
      'subject')

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

>>> ('output : ', [ 'science subjects' + 'physics' + 'chemistry'
      '\n maths'])

```

29/11/19

PRACTICAL - 01

020

Aim :- Demonstrate the use of different file accessing mode under different attributes lead method.

Algorithm :- Create a file object using open method and use the write accessing mode followed by writing some contents onto the file & than closing the file.

2) Now open the file in word and then

Q5

```
c = file  
print()  
>>> c  
d = f  
print  
>>> c
```

Step 4 : Now open the file obj in write mode , write some another content subsequently . Then again open the file obj in 'wt' mode that is the update mode & write content .

Step 5 :- Open filobj in read mode display the update written content & close . Open again in 'rt' mode with parameter passed and display the output subsequent

Step 6 :- Now open filobj in append mode open write method write contents close the file obj again open the file obj in read mode & display the appending output

```
c = fileobj . mode  
print (" file mode ", c)  
>>> c . filemode  
d = file obj . softspace.  
print (" softspace ", d)  
>>> ('softspace :', 0).
```

wt mode.

```
file obj = open ("abc . txt", "wt")  
file obj . write ("Mustafa").  
file obj . close () .
```

write mode 021
file obj = open ("abc . txt", "w")
file obj . write ("Mustafa")
file obj . close () .

rt mode

```
file obj = open ("abc . txt", "rt")  
S1 = file obj . read (s)  
print (" output of u+", 'Mustafa')  
# read mode
```

file obj = open ("abc . txt", "r")
S = file obj . read ()
print ("output:", s)
>>> ('output of mode:', 'Mustafa')

append mode.

```
fileobj = open ("abc . txt", "a")  
fileobj . write ("data structure")  
fileobj . close ()  
fileobj = open ("abc . txt", "r")  
S3 = fileobj . read ()  
print ("output:", S3)  
fileobj . close ()
```

("output", 'Mustafa data structure')

tell()
fileObj = open("abc.txt", "r").
pos = fileObj.tell().
print("tell():", pos).
fileObj.close().
>>> ('tell():', 0)

seek()
fileObj = open("abc.txt", "r").
st = fileObj.seek(0, 0).
print("seek(0, 0) is :", st).
fileObj.close().
>>> ('seek(0, 0) is :', None).
fileObj = open("abc.txt", "r").
str2 = fileObj.seek(0, 2).
print("seek(0, 2) is :", str2).
fileObj.close().
>>> ('seek(0, 2) is :', None).

II finding length of different lines exist within lines
fileObj = open("abc.txt", "r").
stat = fileObj.readlines().
print("Output : ", stat)
for line in stat:
 print(len(line)).
fileObj.close().
>>> ('Output : ', ['Abhay data structure'])

Step 7 :- Open the file obj in read mode declare a variable & perform file obj dot tell method & store the output consequently in variable.

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

6/12/19

SSO

PRACTICAL NO - 2

Objective :- To demonstrate the use of iterator & iterable.

odd Numbers.

Algorithm :-

Step 1 :- Define a __iter__ method with an argument and to initialize the value and return the value.

Step 2 :- Define the next method with an argument and compare the upper limit by using a conditional statement. Increase the value by 2.

Step 3 :- Now create an object of the given class and pass this object in the __iter__ method.

class odd

def __iter__(self) :

self.num = 1

return self.

def __next__(self) :

if self.num <= 0 :

num = self.num

self.num += 2

return num

else

raise StopIteration

y = iter(odd())

~~while next()~~

while True:

print(next(y))

Output

1

3

5

7

9

11

13

15

17

19.

```

class myiter :
    def __iter__(self):
        Pow = 1
        Pow * n1 = int(input("Enter the number"))
        Pow * n2 = int(input("Enter [\"m Maximum limit of Power \"])")
        return Pow
    def next__(self):
        if Pow * n <= Pow * n2:
            num = Pow * n1 * Pow * n
            Pow * n += 1
            return num
        else:
            raise StopIteration
y = iter(myiter())
while True:
    print(next(y))

```

Output :-

```

>>> Enter the number = 2
>>> Maximum limit of power = 4
2
4
8
16

```

power
Algorithms :

Step 1 :- Define its method with 3 arguments.
Initialize the first argument at 1. Initialize the other two arguments as "Enter the number" and "Maximum limit of power" respectively.

Step 2 :- Define the next method with an argument and compare it by using a conditional statement. Increment the value by 1 -

Step 3 :- Now create an object of the given class and pass this object in the its method & use the while conditional statement to print

Range Algorithm

Step 1 : Define a iter method with an argument & initialize the value & return that value.

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

Step 3 : Now create a object of the given class & pass this object to the iter method & use the for statement to print.

class myrange :

025

def __iter__(self) :

self.a = 1

return self

def __next__(self) :

if self.a <= 30

x = self.a

~~self~~

self.a += 1

return x

else:

raise StopIteration

myclass = myrange()

mydata = iter(myclass)

for x in mydata:

print(x)

Output :

1	16
2	17
3	18
4	19
5	20
6	21
7	22
8	23
9	24
10	25
11	26
12	27
13	28
14	29
15	30

~~Not~~

~~Dr~~
~~2012~~

Aim : Program to demonstrate exception handling.

Write a program using the exception method of the native Arithmetic error.

Step 1 : Use the try block and except block to input using the raw_input method and convert into the integer datatype and subsequently terminate the block.

Step 2 : Use the except block with the exception name as value error and display the appropriate message if any suspicious code is part of try block.

Write a program for accepting the file in a given mode and use IOError as an exception for given input.

Step 1 : Within the try block open the file using the write mode and write some content on the file.

Step 2 : Use the except block with IOError and display the message regarding

#1.

while True :

try :

x = int(input("Enter class :"))

break

except ValueError:

print("Enter numeric value").

Output :

Enter class : 457.

#2.

try :

f0 = open("abc.txt", "w")

f0.write("Skilled")

except IOError:

print("Error writing on the file")

else :

print("Operation carried out successfully")

f0.close()

Output :

Operation carried out successfully

Ques #3

```
def assert_(n) :  
    assert len(n) == 0  
    print ("list is empty")  
  
var1 = []  
print (assert_(var1))
```

Output : list is empty.

#4

```
def acceptage () :  
    age = int(input("Enter age"))  
    if age >= 30 or age < 16 :  
        raise ValueError  
    return age  
  
valid = False  
while not valid :  
    try :  
        age = acceptage()  
        valid = True  
    except ValueError :  
        print ("not a valid age")
```

Output :

```
Enter age : 17.  
Not a valid age  
Enter age : 19
```

missing node of the file or incompatibility of the message that the operation is carried out successfully.

- 3) Write a program using the assert() to check if the list elements are empty.

Step 1 : Define a function which accepts an arguments and check using the assert statement whether the given list is empty list and accordingly return the message.

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

- 4) Write a program to check the range of the age of the students in given class and if the age do not fall in given range else the value user exception - otherwise return the valid value

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

350

Step 2 : Use the if condition to check whether two input age falls in the range and so return the age else use the value error exception

Step 3 : Define the while loop to check whether the boolean expression holds true, use the try block to accept the age of student and terminate the looping condition

Step 4 : Use the value error and print the message not a valid input

Xat
Xor

850

code

import re

```
string = "hello 1234 abc 456"
result = re.findall ("1\d", string)
result = re.findall ("1", string)
print (result)
print (result)
```

output

```
>>> ['1234', '4567']
>>> ['hello', 'abc']
```

Aim :- Demonstrate the use of regular expression.

Theory :- Regular expression represents the sequence of characters which is mainly used for finding & replacing the given pattern in a string and for this use import module re and common usage of regular expression involves following functions.

Searching a given string.

Finding a string.

Finding a string into smaller substrings.

Replacing a part of string.

Write a regular expression segregating numeric and alphabetic value from a given string

Algorithm:

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

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

Q3.

⇒ Write a regular expression for finding match string at the beginning of given the sequence

Algorithm

Step 1 : Import re module and apply a string

Step 2 : Use Search() with "A python" and string as two parameters

Step 3 : Now display the output.

Step 4 : Now use if conditional statement for use to know whether match is found or not.

code : 031

```
import re
string = "Python is an important language"
result = re.search('Python', string)
print(result)
if result :
    print("match found")
else :
    print("match not found")
```

output

```
>>> result object . span = (0, 6)
      match = "Python"
>>> match found
```

169

```
# code
import re
li= ["8850897674", "9910256089"]
for element in li:
    result = re.match("[8-9]{1}\d{9}", element)
    if result:
        print("correct mobile no")
        print(result.group())
    else:
        print("incorrect mobile no:")
# output
>>> correct mobile no.
9910256089
correct mobile no.
```

Q.3) Write a regular expression to check whether a given mobile number starts with 8 or a 9 in the total length.

Algorithm

Step 1 : Import re module and apply string of mobile no. s.

Step 2 : Now use for conditional statement to find if the number starts with 8 or a 9 and the total number should length 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 to match or not if we have use group () to display the output and if we don't display incorrect mobile no.

4) Write a regular expression for extracting word and subsequently extract the word without the space.

Algorithm :

Step 1 : Import re-module and apply a string

Step 2 : Use findall () to extract a given word from given string.

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

Step 4 : Now display the output

5) Write a regular expression for extracting first and last .

Step 1 : Import re-module and apply a string

Step 2 : Use findall () in which " /\w+" as one parameter to find first word of string there are " /\w+ \$1 " as parameter to find last word of string

Step 3 : Display the output.

```
# code.
import re.
string = "Python is a lang"
result1 = re.findall("w+", string)
result2 = re.findall("a/w+", string).
print(result1)
print(result2)
```

Output

>>>

code.

```
import re.
string = "Python is a lang"
result = re.findall("a/w+", string)
result1 = re.findall("l/w+", string) -
print(result)
print(result)
```

output

```
>>> ['Python']
>>> ['lang']
```

code

import re

string = "Bla Bla Bla 201 24-12-2019"

result = re.findall(["\d{2}-\d{2}-\d{4}", string])
print(result)

output

>>> [24-12-2019]

code.

import re

string = "abc @ fcsc.edu"

result1 = re.findall("\w+", string)

result2 = re.findall("+\w+", "\w+\\$")

result3 = re.findall("[\w.-]+", string)

print(result1)

print(result2)

print(result3)

output

>>> ['abc']

>>> ['fcsc.edu']

>>> ['abc', 'fcsc.edu']

034

With a regular expression for extracting the
data where the string has followed format $24-12-2019$.

Algorithm

Step 1 : Import re-module and apply string.

Step 2 : Use.findall() method and use \d{2}-\d{2}-\d{4} as a parameter.

Step 3 : Now display the output.

With a re for extracting file:

① Username from email id.

② Hostname from email id.

③ Both Username and Hostname

~~Both~~
~~Both~~

Algorithm

Step 1 : Import re module and apply a string.

Step 2 : Use.findall() to find Username, hostname
and both as mail-id.

Step 3 : Display the output.

PRACTICAL - OS : (A)

Topic : Gui Components

Step 1 :- Use the tkinter library importing the features of text widgets.

Step 2 :- Create an object using Tk().

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

Step 4 :- Use the mainloop() function of the corresponding events.

2 :-

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

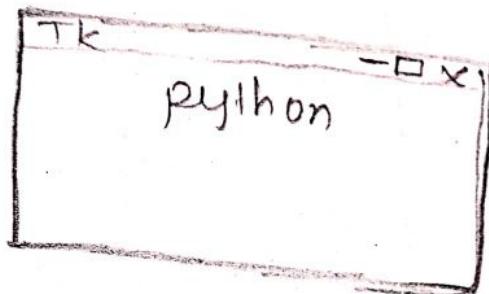
Step 2 :- Create a variable from the parent window and position it on

Gui
From
e
l
s
Output

Creation of parent window.
from Tkinter import *.
root = Tk()
l = Label(root, text = "python")
l.pack()
root.mainloop()

035

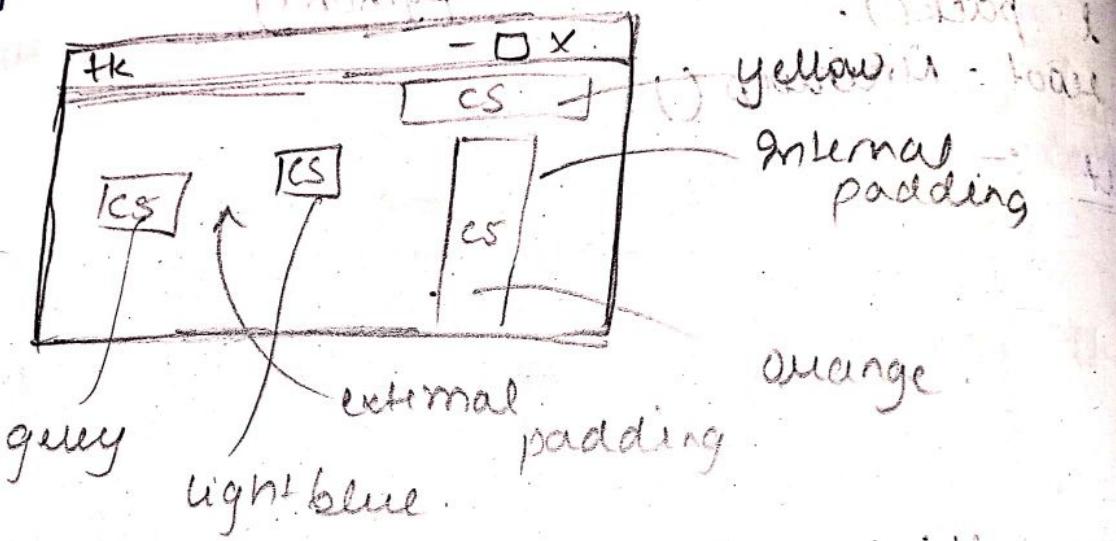
Output :-



#2 :

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")
l2.pack(side = LEFT, pady = 40)~~
root.mainloop()

Output :-



Step 3 :-

(8) ~~20 min~~
Use the pack object created along with the and use the parameters from the `txt()` method.
 1) `side = LEFT`, `padx = 20`.
 2) `side = LEFT`, `pady = 40`.

Step 4 :-

Use the mainloop() for the triggering of the corresponding events.

Step 5 :-

Now repeat the above steps with label() which takes the following arrangements:

- 1) Name of the parent window.
- 2) Text attributes which defines the string.
- 3) The Background colour (bg).
- 4) The foreground colour (fg) and then use the pack() with a relevant padding attribute.

Jr 2017

Practical - 05 (B)

Topic :- Gui Components

#1 :-

Step 1 :- Import the relevant methods from tkinter library to create an object for the parent window.

Step 2 :- Use the parent window object with the geometry () declaring pixel size of parent window.

Step 3 :- Now define a function which tell user about the given selection from multiple option available.

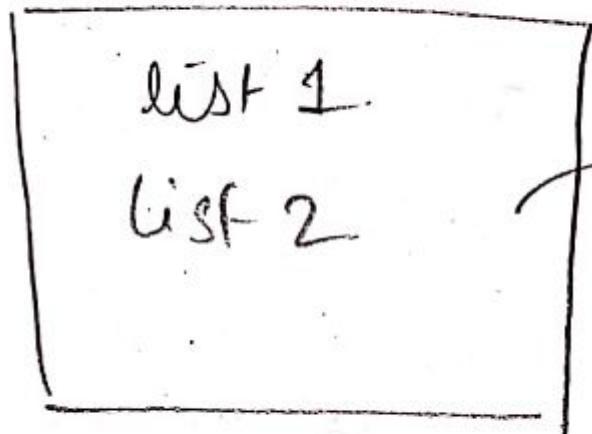
Step 4 :- Now define the parent window object with variable .

Step 5 :- Use the linkbox () and insert on the parent window along with pack () specifying anchor attribute.

Step 6 :- Create an object from radio button following arguments

tk

- □ X



① option 1

② option 2

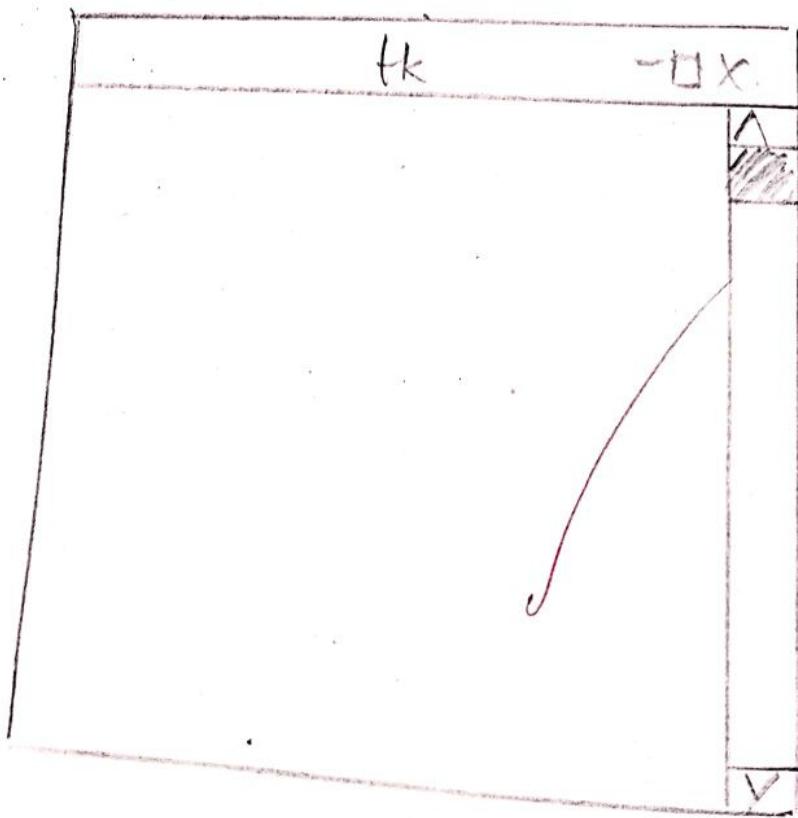
you just selected no. 1.

189

2.5

```
Scrollbar()
from tkinter import *
root = Tk()
root.geometry("500x500")
s = Scrollbar()
s.pack(side="right", fill="y")
root.mainloop()
```

Output :



Step 7 :- Trigger the function method.

Step 8 :- Now call the back() for radio object also created and specify the argument using `anchor` attribute.

Step 9 :- Finally make use of the Mainloop() along with parent object.

2 :-

Step 1 :- Import relevant method from the tkinter library.

Step 2 :- Create a parent object corresponding to the root or the parent window.

Step 3 :- Use the geometry() for laying the window.

Step 4 :- Create an object and use the scroll bar().

Step 5 :- Use the pack() along with the scrollbar object with side and fill attributes.

Step 6 :- Use the mainloop with parent object.

- relevant libraries for
- #3 :- Import the `tkinter` module.
- Step 2 :- Create the parent window as corresponding object of
- Step 3 :- Use the geometry manager with `size(680x400)` or any other suitable pixel value
- Step 4 :- Use the label widget along with the parent object created and subsequently use the `pack` method.
- Step 5 :- Use the frame widget along with the parent object created and use the `pack` method.
- Step 6 :- Use the listbox method along with its attribute like `width`, etc.

3

039

Using frame widget.

```
from tkinter import *  
window = Tk()  
window.geometry("680x500")  
label = Label(window, text = "numbers : ")  
label.pack()  
frame = Frame(window)  
frame.pack()  
listnodes = Listbox(frame, width = 20, height = 20,  
                    font = ("Times New Roman", 10))  
listnodes.pack(side = "left", fill = "y")  
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()
```

#4

from tkinter import *

window = Tk()

window.geometry("680x500")

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 = "Select",
 = "red", fg = "blue")

b2 = Button(frame, text = "Modify",
 = "yellow", fg = "black")

b3 = Button(frame, text = "ADD",
 = "blue", fg = "red")

b4 = Button(frame, text = "EXIT",
 = "red", fg = "green")

b1.pack(side = "left", padx = 20)

b2.pack(side = "right", pady = 40)

b3.pack(side = "bottom", ipady = 20)

b4.pack(side = "top", ipady = 40)

4 :-

Step 1 :- Import relevant methods from tkinter 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 termed as 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,~~

put it into the frame object
side = "Right".

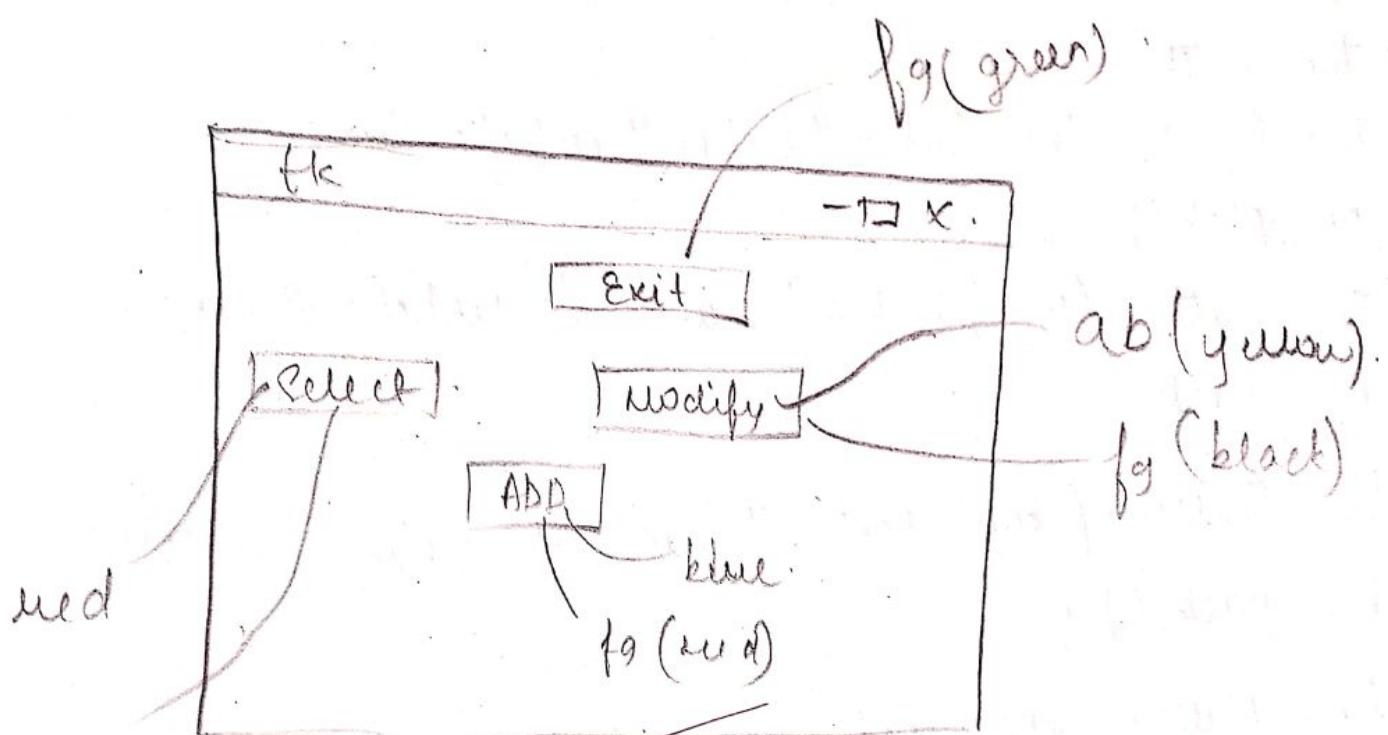
Step 8 :- Create another button object & is on to the Right frame & the button as ADD.

Step 9 :- Add another button & put it on top frame and label as its exit.

Step 10 :- use the pack() simultaneously for all the objects & finally use the Mainloop()

041

Output :-



189

from tkinter import *

top = Tk()
b1 = Button(top, text = "sunken", relief = SUNKEN)

b1.pack()

b2 = Button(top, text = "raised", relief = RAISED)

b2.pack()

b3 = Button(top, text = "Groove", relief = GROOVE) 2)

b3.pack()

b4 = Button(top, text = "ridge", relief = RIDGE)

b4.pack()

b5 = Button(top, text = "flat", relief = FLAT) 3)

b5.pack()

top.mainloop()



Components of GUI (Button, Attribute, Message Box)

Write a program on various attributes which a button widget may assume related to relief attribute.

- 1) Define a button object and place it on to the corresponding parent window.
- 2) Use the text attribute for specifying title to the button object.

SPG

* Showwarning() :-

- 1) Define a function which will use the showwarning() derived from message box.
- 2) The attributes which a given method takes will specify the two strings one related to the message displayed corresponding to message.
- 3) Now create an object from the method and place it on the parent window with the title of the button object specified and finally use the command attribute to execute.
- 4) Use mainloop() to terminate.

```
from tkinter import *
```

```
top = Tk()
```

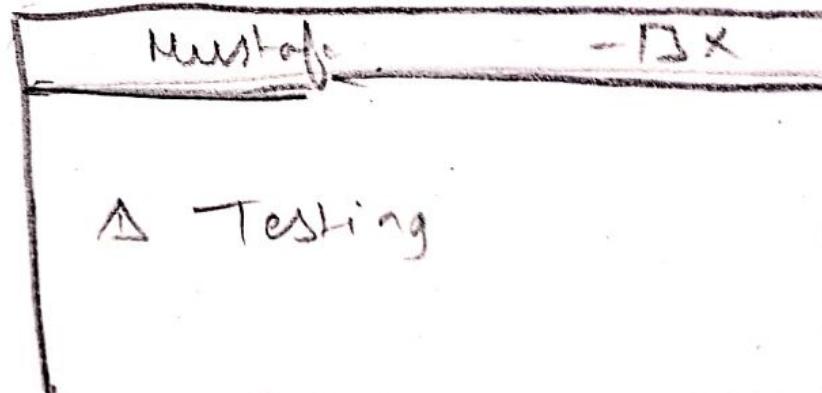
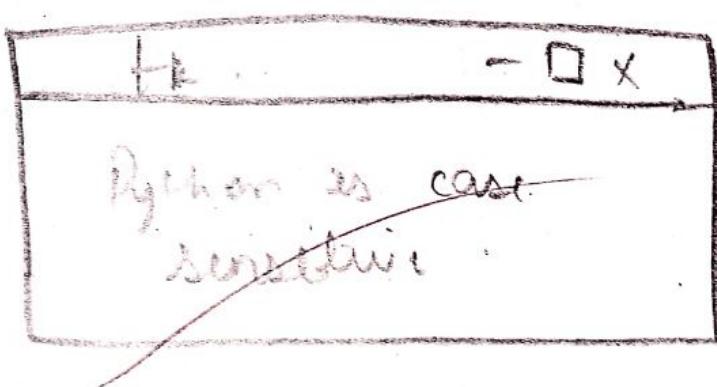
```
messagebox
```

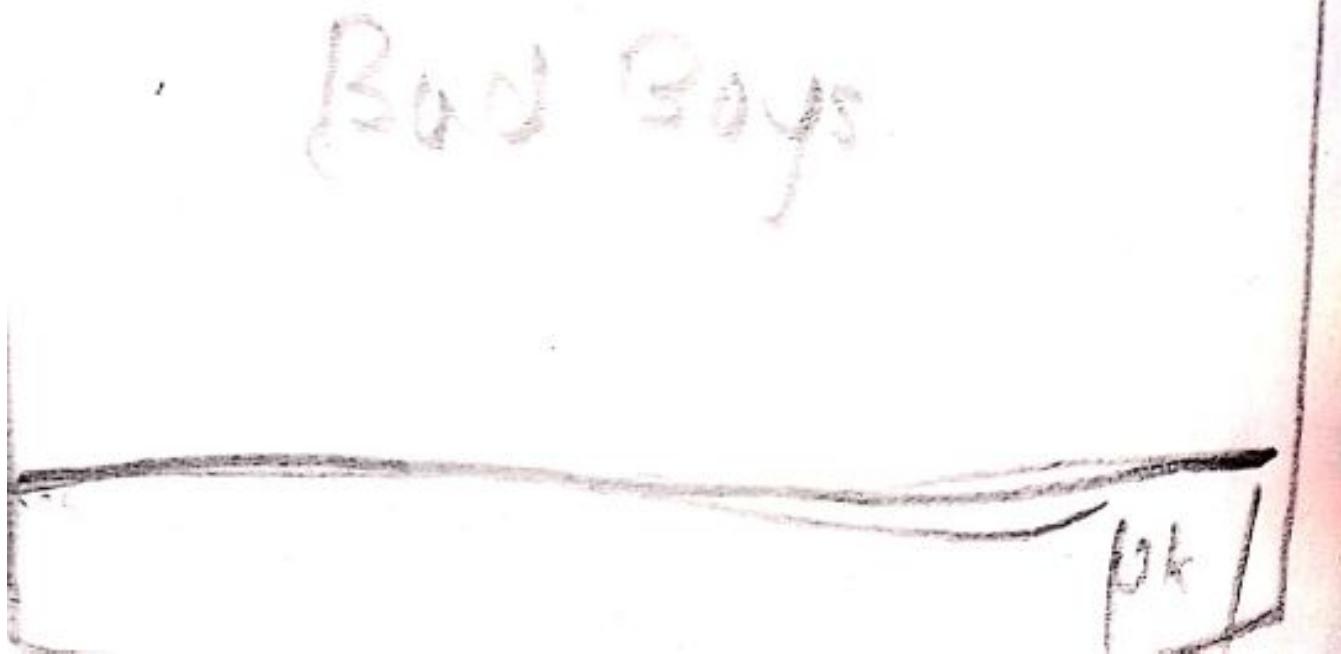
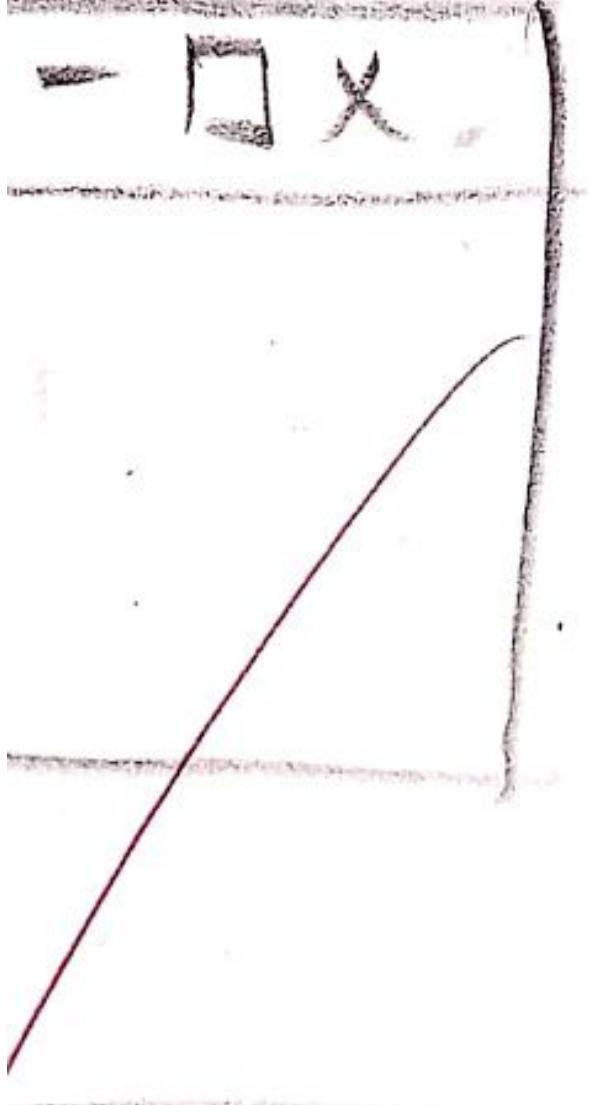
```
def mgb():
```

 messagebox.showwarning("Mustafa", "Testing")
 b1 = Button (top, text = "Python is case sensitive", com
 = mgb)

```
b1.pack()
```

```
top.mainloop()
```





* Show Info :-

- 1) Define function derived from which will use the showInfo() method will use the message box library.
- 2) The attributes which a given method takes will specify two string. One related to the title can correspond to message displayed.
- 3) Now create an object from button method and place it on parent window with title of button object and finally use command attribute.
- 4) Terminate by using a mainloop() method.

* Ask yesno()

4) Define a function which will use askyesno
derived from messagebox library.

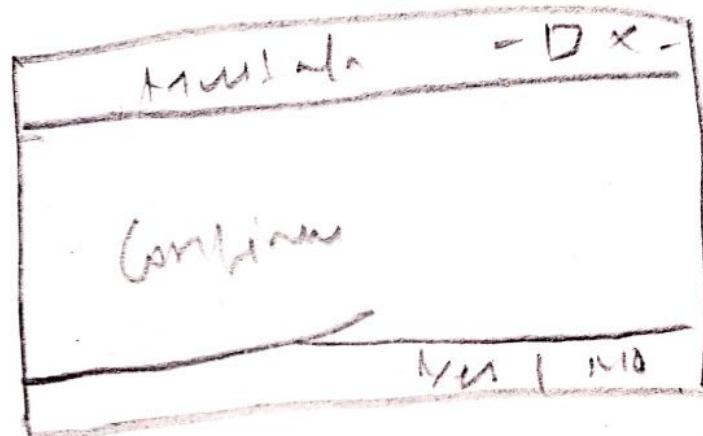
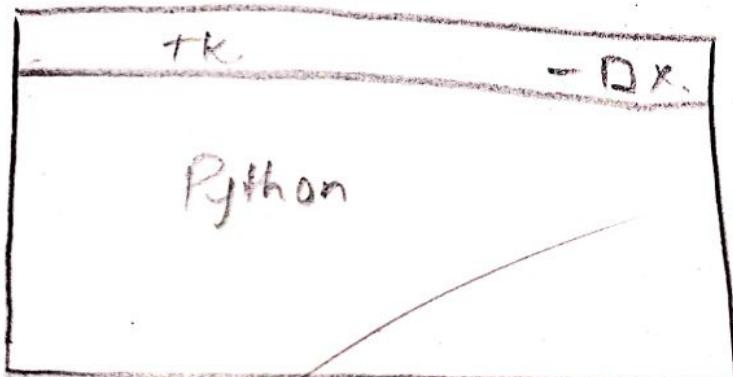
2) The attribute gives from method takes
of the window i) Related to title of
window ii) Corresponding to message displayed

3) Now create an obj from the button
method and place to parent window with
title object specified and finally use the
command attribute to create function.

(4) Terminate by calling mainloop() method.

Q45

```
from tkinter import *
top = tk()
def msgb():
    messagebox.askyesno("Mustafa", "Continue")
b1 = Button ( top, text = "Python", command = msgb)
b1.pack()
#mainloop()
top.mainloop()
```



219

```
from tkinter import *
Top = Tk()
messagebox()
def msg():
    messagebox.showerror("mustafa", "Syntax error")
b = Button(top, text = "python", command = msg)
b.pack()
top.mainloop()
```

Q46

Showerror() :-

- 1) Define a function which will call use the Show error () derived from the message box library.
- 2) The attribute which a given method take will specify two string.
 - i) Related to title.
 - ii) Corresponding to the message.

Ques:-

* Askquestion () :-

- 1) Define a function which will use the askquestion() from message box library.
- 2) The attributes which a given method will take will specify two strings :-
 - i) Related onto the title .
 - ii) Corresponding to message displayed.
- 3) Now create obj from button method and place onto the parent window.
- 4) Terminate by using mainloop () .

#

* Ask ok cancel :-

- 1) Define function which use the askokcancel () from message box library
- 2) The attribute takes which gives to method will specify two string :-
 ① Related to title .
 ② Corresponding to message displayed.
- 3) Now create an obj from button method and place it onto parent window
- 4) Terminate by using mainloop ()

Task

Write a program to move from one window to another window with the help of button widget.

Step 1 :- Define a function and create a parent window object and use the config, title, minsize function.

Step 2 :- Now define a button object place it on to the parent window with a suitable title and use command attribute to call the next function using the grid method specifying external padding.

Step 3 :- Now define a function corresponding to second window and create another parent window object with config method and again calling the place, title, and minsize function again and calling the next function.

Page

```
from tkinter import *
def selection():
    root = Tk()
    root.config(bg = "red")
    root.title("One window")
    root.minsize(200, 200)
    b1 = Button(root, text = "Next window", command =
    new1).pack(padx = 10, pady = 20)
    root.mainloop()
def new1():
    top = Tk()
    top.config(bg = "pink")
    top.title("Second window")
    top.minsize(170, 170)
    b2 = Button(top, text = "Second window", command =
    exit).pack(padx = 20, pady = 30)
    top.mainloop()
def exit():
    quit()
selection()
```

One window = X

next window

Step 4 :- Similarly create the function ⁰⁵⁰ and use the button widgets and finally create a function which will terminate the aggregate function by using quit method.

Dr. M

~~top~~ from tkinter import *

top = Tk()

top.title("mustafaaaa")

top.config(bg="pink")

top.minsize(200, 200)

frame = Frame(top)

leftframe = frame(width=100, height=100,
bg="green").grid(row=0, column=0,
padx=10, pady=15)

rightframe = frame(width=75, height=75,
bg="yellow").grid(row=0, column=1,
padx=15, pady=20).

Label(leftframe, text="mustu", bg="yellow")
grid(row=0, column=0)

Label(rightframe, text="mustuu", bg="black")
grid(row=0, column=1)

image = PhotoImage(file="Desert.gif")

ani-image = image.subsample(1, 2)

Write a program to insert an image in the frame other widgets using the

Step 1 : Create the parent window object and use the method title, config and minsize with this object.

Step 2 : Create an object from the frame window and place it onto the parent object with width, height and bg colour and use the grid method() along with row and column attribute as (0,0) with some external padding.

Step 3 : Similarly create the rightframe object from the frame method with row and column attribute making the values (0,1).

Step 4 :- use the label () & the parent window object corresponds to left frame with text and relief attribute and use the grid method with row and column value as (0,0)

Step 5 :- similarly create the label
and the right frame and use
the row, column value as $(0,1)$.

Step 6 :- use the ~~photo~~ photo() with
attribute specified and subsequently
subsample() for specifying the image
object.

Step 7 :- Now use the label() using
leftframe and the image attribute
and the row, column value specified
in guid().

Step 8 :- similarly create the label() using
rightframe obj with the image attribute
with some bg colour $(0,0)$.

Step 9 :- Now define a function using
statement which shall be called on
clicking the window.

Step 10 :- Now to terminate the mainloop we
use the given program.

Step 10 :- Create the label
the toolbox with method
some position it
title.

052

Insert Image

- X

image 1

image 2

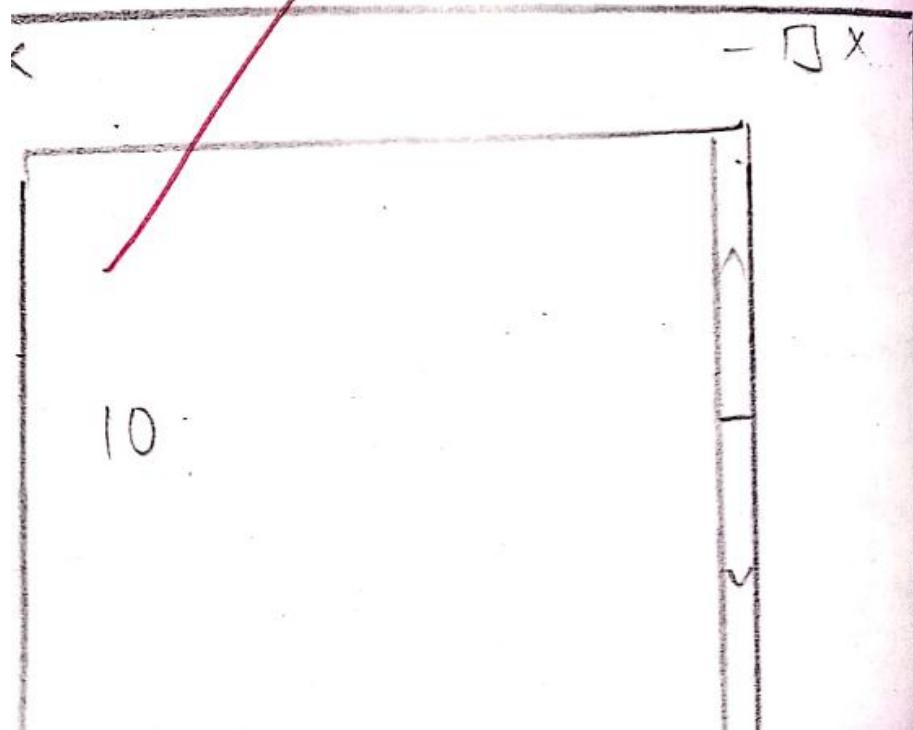
~~from, from - = 0, by~~ to)

loop()

input *

master, from - = 0 , to = 10, by
ck ^ , font = " . . . ")

)()



Write a program to make use of Spinbox Widget.

Step 1 :- Use the tkinter library to import the relevant methods.

Step 2 :- Create the parent window object.

Step 3 :- Create an object by using spinbox Method & place it on to the parent window with specified option.

Step 4 :- Now use the pack method () to make object visible on the parent window and than use the mainloop method to terminate the program.

Dr. M

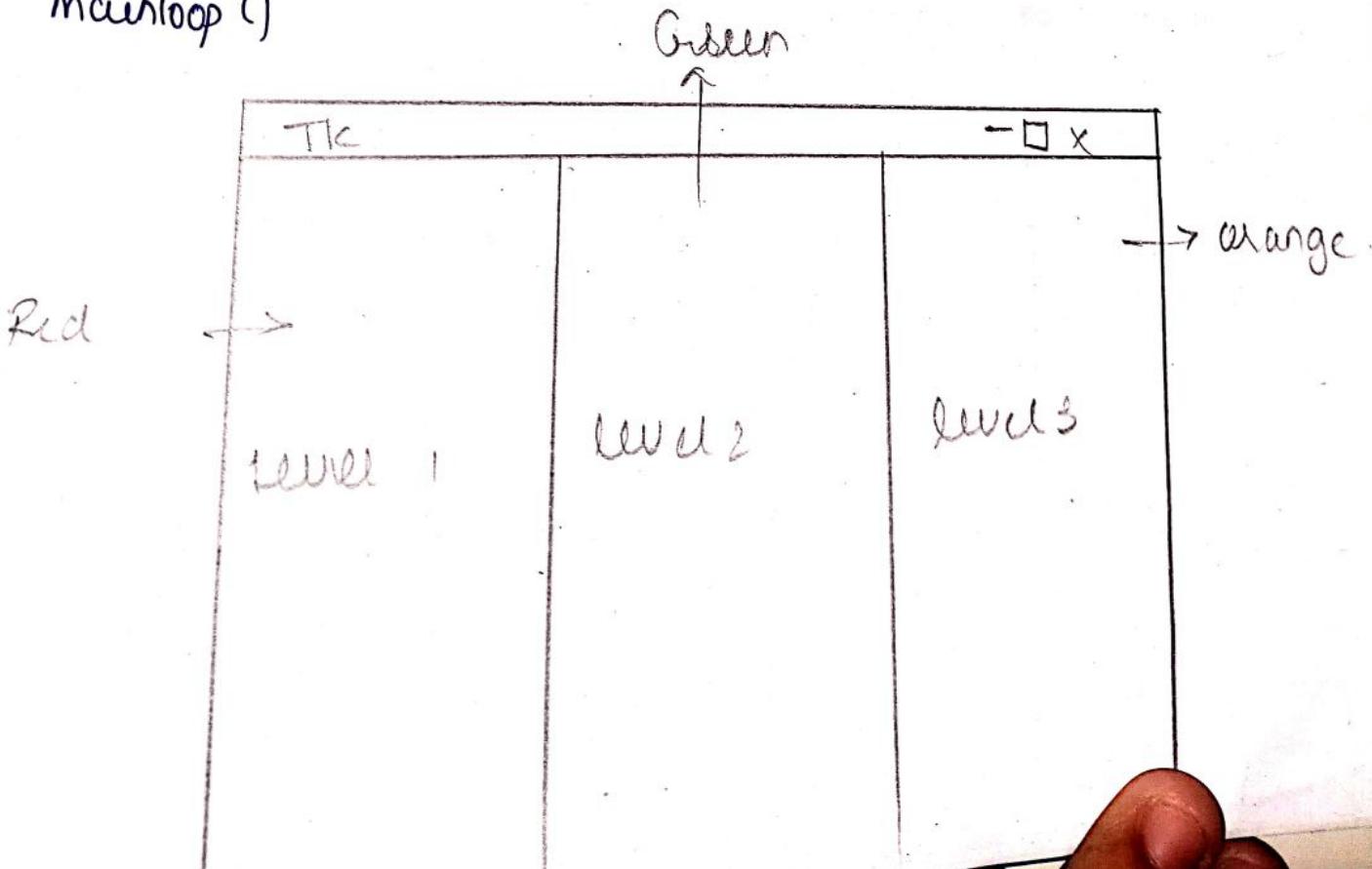
*. Paned window

- i) Create an object from the method and use the pack method to make the object visible.
- ii) Now create an object from the widget and place it onto the and use the add method. Similarly create object from a paned window.
- iii) Create an object & place it onto the and use the add method accordingly.
- iv) Create a button widget with the paned window define a function and place it.
- v) Use the pack() method & for the corresponding event to trigger. Now mainloop

Code :-

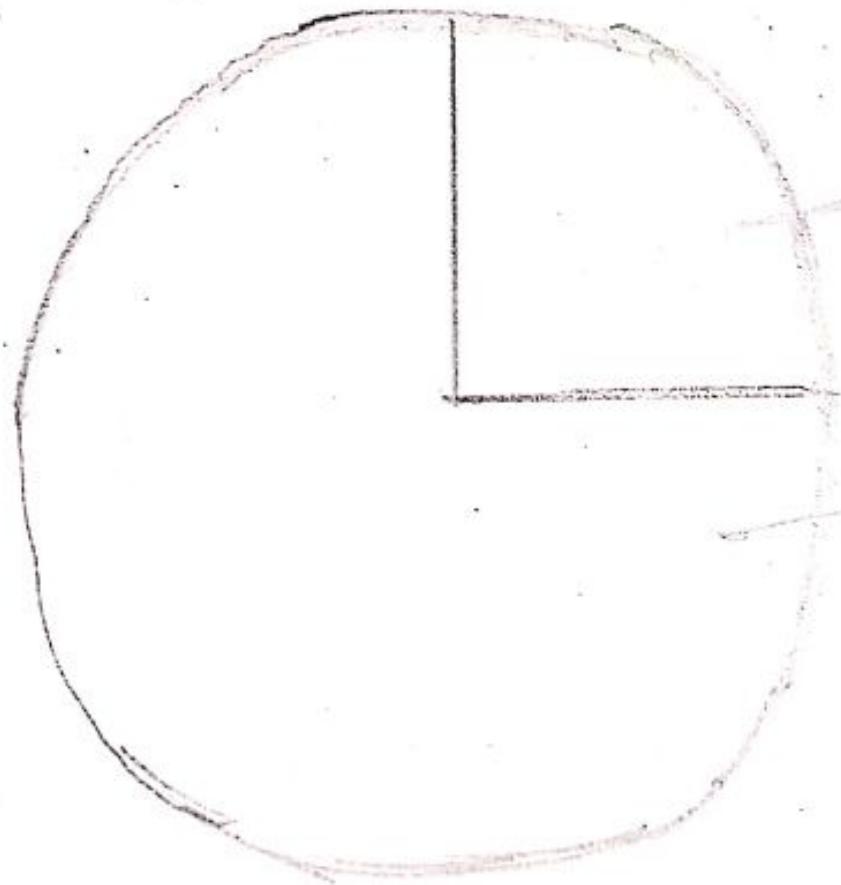
054

```
from tkinter import *
root = Tk()
P = Panedwindow(bg = "pink")
P.pack(fill = BOTH, expand = 1)
L = Label(P, text = "LEVEL 1", bg = "red")
P.add(L)
P1 = Panedwindow(orient = VERTICAL, bg = "yellow")
P.add(P1)
L2 = Label(P1, text = "LEVEL 2", bg = "green")
P1.add(L2)
P2 = Panedwindow(orient = HORIZONTAL, bg = "blue")
L3 = Label(P2, text = "LEVEL 3", bg = "orange")
P.add(P2)
P2.add(L3)
mainloop()
```



TK

- □ X



* Canvas Widget

- i) Create an object from the canvas widget by using the attribute height, width, bg colour & parent window object.
- ii) Use the corresponding method for drawing the simple geometrical shape like arc, oval & line and specify the coordinate values.
- iii) Similarly use the create line & create oval method along with the co-ordinate value & fill attribute for specifying the colour.