

PRACTICAL NO. 01

Objective :- Demonstrate the uses of different file accessing modes, different attributes, read method.

Step 1 :- Create a file object using open method and use the write access mode followed by writing some contents onto the file and then closing the file.

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

Step 3 :- Now use fileobject for finding the name of the file, the mode in which it is opened whether the file is still open or close and finally the output of the soft space attribute.

is

4) write mode

Step 4 : Now open the file object in write mode write some other contents close subsequently then again open file object in 'w+' mode that is the update mode and write contents

Step 5 :- Open fileobject in read mode . display the update written contents and close open again in 'r+' mode with parameter passed and display the output subsequently .

Step 6 :- Now open fileobject in append mode open write method to write content & close the fileobject again open the fileobject in read mode and display the append output .

Step 7 : Open the fileobject in read mode, declare a variable and perform fileobject dot tell method and store it, the output consequently in variable pos

Step 8 : use the seek method with the arguments with opening the fileobject in read mode and closing subsequently.

Step 9 : open fileobject with read mode also use the readlines methods and store the output consequently in and print the some for counting the length use the for condition statement and display the or

SS

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

fileobj = open ("abc.txt", "r")
str1 = fileobj.read ()
print ("The output of read method ", str1)
>> ('The output of read method : ', 'Computer Science subjects in DBMS
in python in DS\n')

# read line()
fileobj = open ("abc.txt", "r")
str2 = fileobj.readline ()
print ("The output of readline method ", str2)
>>> ('The output of readline method : ', 'Computer science subjects\n')

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

>>> ('The output of readline method : ', 'Computer Science
Subject \n', 'DBMS\n', 'python\n', 'DS\n')
```

file attributes

a = fileobj.name

print("name of file (name attribute)", a)

>> ('name of file (name attribute)', 'abc.txt')

b = fileobj.closed

print("(close) attribute", b)

>> ('(close) attribute', True)

c = fileobj.mode

print("file mode", c)

>> ('file mode', 'r')

d = fileobj.softspace

print("softspace", d)

>> ('softspace', 0)

w+ mode

fileobj = open("abc.txt", "w+")

fileobj.write("chandresh")

fileobj.close()

r+ mode

fileobj = open("abc.txt", "r+")

str1 = fileobj.read()

print("output of r+", str1)

fileobj.close()

>> ('output of r+', 'chandresh')

append mode

```
fileobj = open ("abc.txt", "a")
```

```
fileobj.write ("data structure")
```

```
fileobj.close()
```

```
fileobj = open ("abc.txt", "r")
```

```
str2 = fileobj.read()
```

```
print ("Output of append mode", str2)
```

```
fileobj.close()
```

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

tell

```
fileobj = open ("abc.txt", "r")
```

```
pos = fileobj.tell()
```

```
print ("tell()", pos)
```

```
fileobj.close
```

```
>> (tell(), pos)
```

Seek

```
fileobj = open ("abc.txt", "r")
```

```
str3 = fileobj.seek (0, 0)
```

```
str8 = fileobj.read (10)
```

```
print ("The beginning of file =", str8)
```

finding length of different line exist within file

```
fileobj = open ("abc.txt", "r")
```

```
str9 = fileobj.readlines()
```

```
print ("Output", str9)
```

```
for line in str9
```

```
print (len(line))
```

```
fileobj.close()
```

```
>> (Output:- ['college database'])
```

PRACTICAL NO: 2

Topic : Iteration and Iterable

Step 1 : Create a tuple with all elements that we need

Step 2 : Then uses iter and next method for iterating element and getting next element in the tuple.

Step 3 : After pointing to the next element, iterate and get output

Aim : iter method using Conditional Statement

Step 1 : The obtained Statement by using similar output for can be Conditional

Step 2 : An iterable variable is to be declared in for loop which will iterate.

Mytuple1 = ("chandresh", "deepak", "Aniket")

My iter1 = iter (Mytuple1)

print (next (myiter1))

print (next (myiter1))

print (next (myiter1))

>>> chandresh

>>> deepak

>>> Aniket

④

My tuple = ("chandresh", "Gupta")

for x in mytuple

print (x)

>>> chandresh

Gupta.

Aim :- Write a program using iterable object to display set of (i) first 20 numbers

Step 1 :- Define a iter method with the argument and initialise to the first value

Step 2 : For extracting next element from the Container use the next method with the argument and compare the no of element required in a Container by using the Conditional Statement

Step 3 : Now Create an objects from the given class and pass the objects as argument to the iter method.

Step 4 : Now use in the conditional statement display all the values from the given Container.

35

```
class MyClass:  
    def __iter__(self):  
        self.a = 1  
        return self  
    def __next__(self):  
        if self.a <= 20:  
            x = self.a  
            self.a += 1  
            return x  
        else:  
            raise StopIteration
```

```
myobj = MyClass()
```

```
myiter = iter(myobj)
```

```
for x in (myiter):  
    print(x)
```

output :-

```
>>> 1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20
```

Aim : To write a program using the iterable method for displaying set of odd numbers.

Step 1 : Define a class and with in that define the iter() method which will initialise the first element with in the container objects

Step 2 : Now use the next() method and define the logic for displaying the odd values.

Step 3 : about code the will throw exception if the object are not able to define the element from the given set with

Step 4 : Now create an object for the given class and pass the object as per argument to the iter() method.

Step 5 : Now using the conditional objects display all value from the given container

class Odd :

def __iter__(self):

self.num = 1

return self

def __next__(self):

num = self.num

self.num += 2

return num

myobj = Odd()

myiter = iter(myobj)

x = int(input("Enter a number"))

for i in myiter

if i < x:

print(i)

Output :-

>> Enter a number : 8

1

3

5

7

25

Aim : Write a program check the whether weather odd or even number is given

Step 1 : Declare a listnum variable and declare some elements in it
like 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Step 2 : Now use map method with help of lambda function

Step 3 : Given two argument display the output.

Step 4 : Define even function with one parameter then check whether the number is even and odd and return respectively.

30

```
listnum = [0, 4, 5, 7, 11, 13, 15, 17, 19]
```

```
listnum = list(map(lambda x: x%2, listnum))
```

```
print(listnum)
```

```
def even(x):
```

```
    if x%2 == 0:
```

```
        return "even"
```

```
    else:
```

```
        return "odd"
```

```
list(map(even, listnum))
```

Output :

```
[0, 4, 0, 1, 2, 3]
```

Aim : To find square and cube of element using map function

Step 1: Define a function name Square with a parameter of which will obtain output of square value of given number

Step 2: In similar function declare function Cube of which get the return value raised to 3 and return the same

Step 3: Call the declared function using map function

Ans:- without using map function

Step 1: Define a list variable with a given set of numbers

Step 2: Define an empty list which will contain output
use the for conditional statement and subsequently
use append () method for finding the square
value and finally print statement for the output

Q8

```
[P1, P2, P3, P4, P5, P6] = [None, None, None, None, None, None]
def square(x): # x is float. If you want integer, then int(x)
    return (x**2)
def cube(x):
    return (x**3)
func1 = [square, cube]
for n in range(4):
    value = list(map(lambda x: x(n), func1))
    print (list(value))
```

Output :-

```
[16, 64]
```

without using map()

```
list1 = [1, 2, 3, 4, 5]
```

```
empty = []
```

```
for i in list1:
```

```
    empty.append(i**2)
```

```
print(empty)
```

Output :-

```
[1]
```

```
[4]
```

```
[9]
```

```
[16]
```

```
[1, 4, 9, 16]
```

```
[1, 4, 9, 16, 25]
```

PRACTICAL 3.

- i) Aim :- Using the exception block related to the environment error.

Step 1 :- Use other try block to define the normal course of action.

Step 2 :- Use the exception block with the I/O environment error and convey the appropriate message to the user. else display the message that the operation is carried out successfully.

try :

```
fo = open ("abc.txt", "w")
```

```
fo.write ("chandresh")
```

except IOError:

```
print ("It is working")
```

else:

```
print ("operation success")
```

Output :-

operation success

Aim :- A program for demonstrating the use of value error in the given program statement.

Step 1 :- Accept the value from the user & if it is a valid value display the entered value & terminate the condition by using break statement.

Step 2 : Define the except block with value error as a keyword and display the appropriate message.

Step 3 : we can the multiple exception using the except statement for finding the different category of error.

Jyoti

26

try :-

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

```
except ValueError:
```

```
    print("This is value error")
```

```
except IOError:
```

```
    print("This is environmental error")
```

```
else:-
```

```
    print("The operation success")
```

Output :-

» Enter a number: q

This is value error

» Enter a number: 12

This is operation success

PRACTICAL NO.4

topic : Regular expression

Step 1 : Import re module declare pattern and declare sequence use match method . declare arguments if arguments matched then print the same otherwise print .

Step 2 : Import re module declare pattern with literal and meta character values use the findall() with argument and print the same .

Step 3 : Import re module declare pattern with meta character use the split() and print the output .

Step 4 : import re module declare string and according declare pattern replace the blank space with no-space, use sub with 3 argument and print the string without Space.

Step 5 : Import re module declare a Sequence use Search method for finding Subsequently use the group () with do operator as Search () given memory exception using groups () if gives will show up the matched string

Step 6 : Import re module declare list with numbers, use the conditional statements here we have used up the for condition Statement. Use if condition for checking first number is either 8 or 9 and next number are in range of 0 to 9 and check whether the entered number are equal to 10. if criteria matched print all number matched otherwise print failed.

Step 7 : Import re module declare a String use module with finding all() for finding the value if the string and declare the same.

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

Step 9 : Import re module enter a String use pattern to display only two element of the particular string use findall () declare two variable with initial value as zero use for condition and Subsequently use the if Condition check whether condition satisfy add up the value on else increment value and display the subsequently.

1) # match()

```
import re
pattern = re.compile("FyCS")
```

Sequence = "FyCS represent Computer science string"

If re.match(pattern, Sequence):

```
    print("matched pattern found")
```

else:

```
    print("not found")
```

>> matched pattern found

2) # numerical value (Segmentation)

```
import re
```

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

String = ' hello123, howdy789, 45howu '

Output = re.findall(pattern, String)

```
print(output)
```

>> ['123', '789', '45']

3) # split()

```
import re
```

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

String = " hello123, howdy789, 45howu "

Output = re.split(pattern, String)

```
print(output)
```

>> ['hello', 'howdy', 'howu']

```
a) # no-space  
import re  
string = 'abc def ghi'  
pattern = re'\s+'  
replace = ''  
v1 = re.sub(pattern, replace, string)  
print(v1)  
'''>>> abcdefghi
```

```
b) # group()  
import re  
sequence = "python is an interesting language"  
v = re.search("python", sequence)  
print(v)  
v1 = v.group()  
print(v1)  
'''>>> <_Sre_SRE-Match object at 0x0281DFOO>  
python
```

```
c) # Verifying the given set of phone number  
import re  
list1 = ['800US6789', '9143673210', '7865432481', '9876543201']  
for value in list1:  
    if re.match(r'[8-9]\{1\}[0-9]\{9\}', value and  
              len(value) == 10):  
        print("Criteria match for cell number!")  
    else:  
        print("Criteria failed")
```

>>> Criteria matched for all numbers
 Criteria matched for all numbers
 Criteria failed!
 Criteria matched for all numbers

value

import re

start = 'plant is life overall'

output = re.findall(r'\b[aeiou.]+\b', start)

print(output)

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

host of domain

import re

seq = ['abc.tesc@edu.com', 'xyz@gmail.com']

pattern = r'\b[\w\.-]+\b[\w\.-]\b'

output = re.findall(pattern, seq)

print(output)

>>> ['abc.tesc', 'edu.com', 'xyz', 'gmail.com']

Counting of first 2 letters :

import re

s = 'msa.a, ms.b, ms.c, ms.t'

p = r'\b[m s / m a l]\b+

o = re.findall(p, s).

print(o)

m = 0

f = 0

for v in o:

28

and write a program to find the number
of males and females in a group
according to the following condition:
if ($v == 'ms'$)
 $f = f + 1$ // count the no. of females
else:
 $m = m + 1$
print ("No. of males is : ", m)
print ("No. of females is : ", f)
>> ['m1', 'ms', 'ms', 'm1']
(No. of males is : 2)
(No. of females is : 2)

PRACTICAL - 5

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

a) Algorithm :-

Step 1 :- Use the Tkinter for importing the feature of text widget.

Step 2 :- Create a variable from text & position it also onto the parent window.

Step 3 :- use the pack() along with the object created from text method & use the parameter.

i) Side = Top, ipadx=20, ipady=50

Step 4 :- use the mainloop() method for triggering Corresponding event.

Step 5 :- Now repeat above step with a label method which takes the following argument.

i) Name of parent window.

ii) Text attribute which defines the string

iii) The background colour (bg)

iv) The foreground colour (fg)

Now use pack() with relevant attributes.

PROGRAM :-

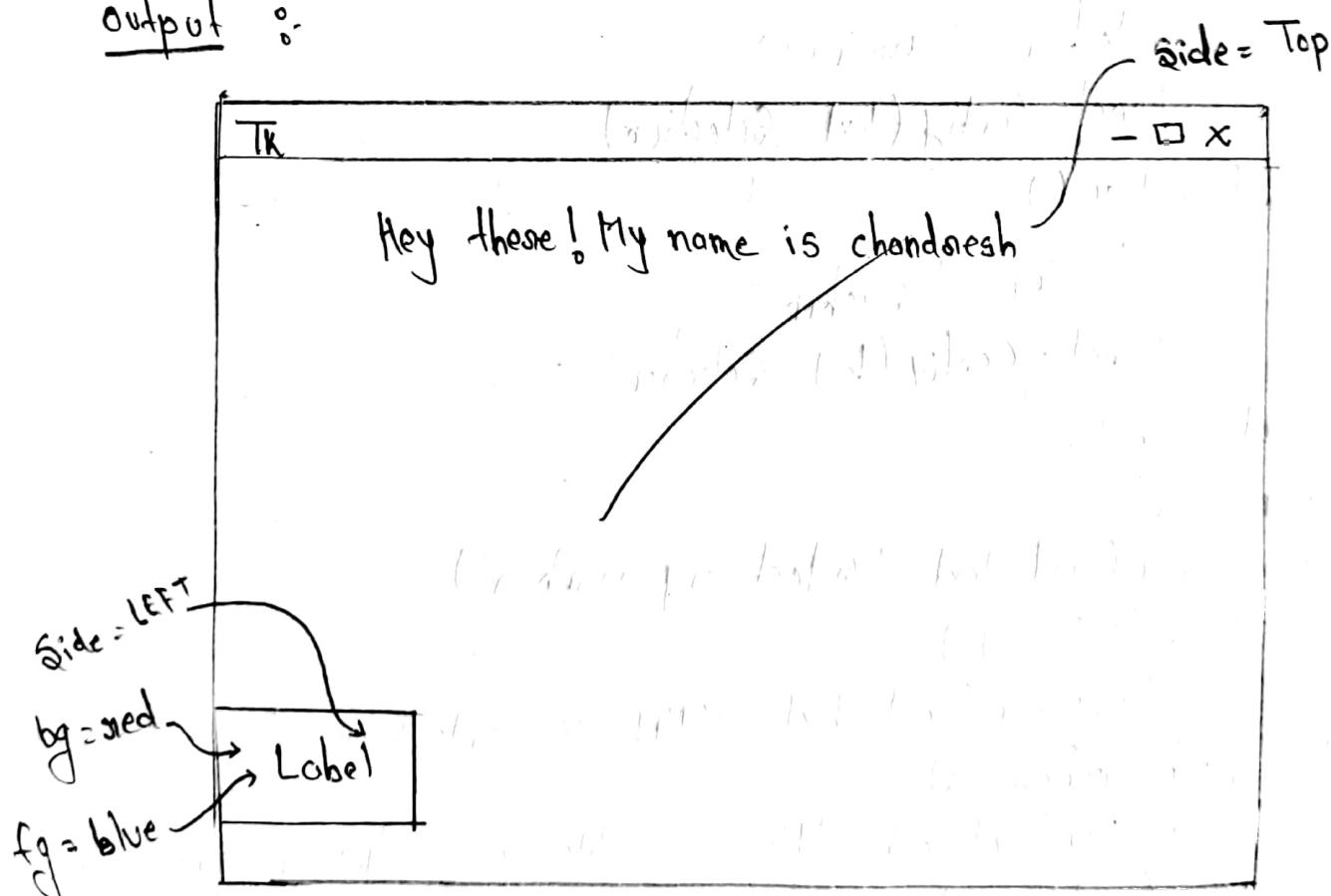
a)

```
from tkinter import * . pack method
root = Tk()
T1 = Text(root)
T1.insert(END, "Hey There! My name is chondaesh:")
T1.pack(side=TOP, padx=20, pady=30, ipadx=40, ipady=50)
L1 = Label(root, text="Label", bg="red", fg="blue")
L1.pack(side=LEFT, padx=10, ipadx=20, ipady=30)
root.mainloop()
```

38

38

Output :-

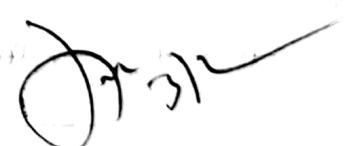


- b) Aim :- To make use of RadioButton widget for Selection of one of the option.
- Algorithm :-
- Step 1 : use the `tkinter` method to import the `radiobutton` method
- Step 2 : Define a function which tells user about given Selection mode from multiple option available
- Step 3 : use the `config` method along with label method & call the variable as an argument with in method.
- Step 4 : Now define the parent window & define option using control variable.
- Step 5 : Now Create of `RadioButton` which will take following argument
- i) position on parent window
 - ii) Text variable
 - iii) Define variable argument
 - iv) Corresponding value function. the trigger the given

Step 6 :- Now call the pack method for Corresponding Radio object to Create and specify argument as anchor attribute.

Step 7 :- Now define a Label object and place it onto parent window using pack method & finally use main loop method.

After defining label object and placing it onto parent window using pack method, now run the program and display will be like this.



Jr. S

b) PROGRAM :

Radio Button

```
from tkinter import *  
def Sel1():  
    Selection = "chandresh"  
    Label.config(text=Selection)  
def Sel2():  
    Selection = "Raj"  
    Label.config(text=Selection)  
def Sel3():  
    Selection = "pranay"  
    Label.config(text=Selection)  
def Sel4():  
    Selection = "Sachin"  
    Label.config(text=Selection)  
root = Tk()  
var = IntVar()  
L1 = Label(root, text="Select any number")  
L1.pack(side=TOP)  
R1 = Radiobutton(root, text="1747", variable=var, value=0, command=Sel1)  
R1.pack(anchor=N)  
R2 = Radiobutton(root, text="1746", variable=var, value=1, command=Sel2)  
R2.pack(anchor=N)  
R3 = Radiobutton(root, text="1745", 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)

root . mainloop ()

(root . label . grid
row = 0, column = 0)

(root . label . grid
row = 1, column = 0)

(root . label . grid
row = 2, column = 0)

(root . label . grid
row = 3, column = 0)

(root . label . grid
row = 4, column = 0)

(root . label . grid
row = 5, column = 0)

(root . label . grid
row = 6, column = 0)

(root . label . grid
row = 7, column = 0)

(root . label . grid
row = 8, column = 0)

(root . label . grid
row = 9, column = 0)

(root . label . grid
row = 10, column = 0)

(root . label . grid
row = 11, column = 0)

(root . label . grid
row = 12, column = 0)

(root . label . grid
row = 13, column = 0)

(root . label . grid
row = 14, column = 0)

(root . label . grid
row = 15, column = 0)

(root . label . grid
row = 16, column = 0)

(root . label . grid
row = 17, column = 0)

(root . label . grid
row = 18, column = 0)

(root . label . grid
row = 19, column = 0)

(root . label . grid
row = 20, column = 0)

(root . label . grid
row = 21, column = 0)

(root . label . grid
row = 22, column = 0)

(root . label . grid
row = 23, column = 0)

(root . label . grid
row = 24, column = 0)

(root . label . grid
row = 25, column = 0)

(root . label . grid
row = 26, column = 0)

(root . label . grid
row = 27, column = 0)

(root . label . grid
row = 28, column = 0)

(root . label . grid
row = 29, column = 0)

(root . label . grid
row = 30, column = 0)

Output :-

TK

Select any roll number from below

(radio buttons)

1747

1746

1745

1744

Raj

c) 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 objects corresponding to the parent window & Create an object from Scrollbar & place on the parent window so Create

Step 3 :- Create an object of label method to provide heading and place it on parent window

Step 4 : use pack method of list box and place it on to parent window with attribute yscroll command along with side & fill.

Step 5 :- Create an object listbox method and placed it on to parent widow with attribute yscroll command.

Step 6 : use for loop to insert values in the object of list Box by using insert method.

Step 7 : use config method along with scroll bar object if use Command attribute.

Step 8 : Finally call the mainloop .

c)

PROGRAM :-

```
from tkinter import * Scrollbar
root = Tk()
root.geometry('450x400')
l = Label(root, text = "B Batch Roll Number : ", bg = "black",
          fg = "red")
l.pack()
scroll = Scrollbar(root)
scroll.pack(side = RIGHT, fill = Y)
```

```
mylist = Listbox(root, textvariable=scrollcommand=scroll.set,  
                 bg="light blue")
```

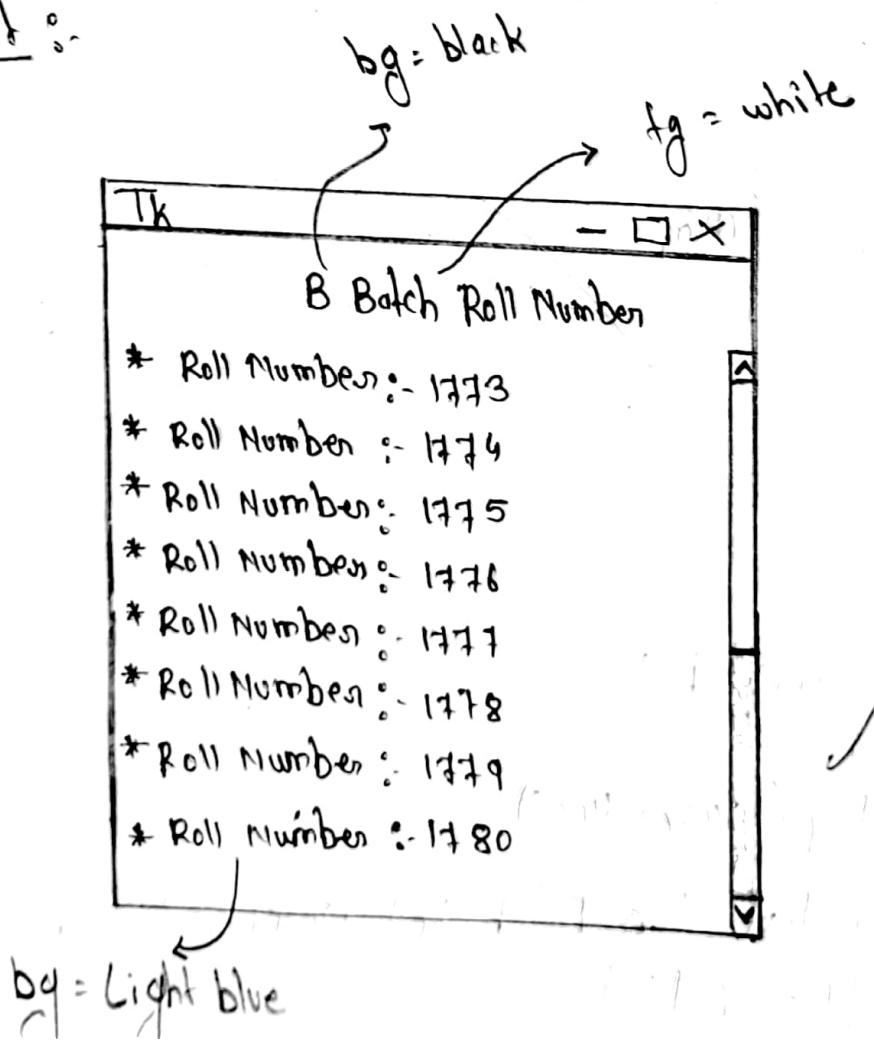
for name in string(41, 81):

```
    mylist.insert(END, "* Roll Number: " + str(name))
```

```
mylist.pack(side=LEFT, fill=BOTH)
```

```
scroll.config(command=mylist.yview)  
root.mainloop()
```

Output :-



D) Aim :- To make use of message Box method of the application
programm :-

Step 1 : Import relevant method from tkinter library

Step 2 : Define a function and use messagebox along with different method available which contains are on more argument

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

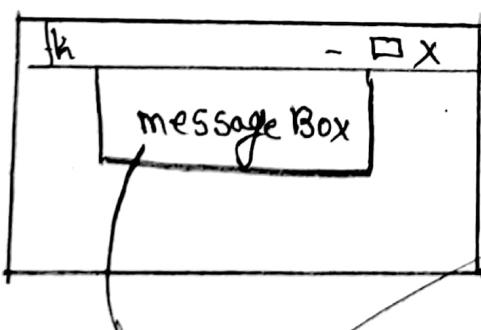
Step 4 :- use pack method and finally use the mainloop method.

~~QUESTION :-~~

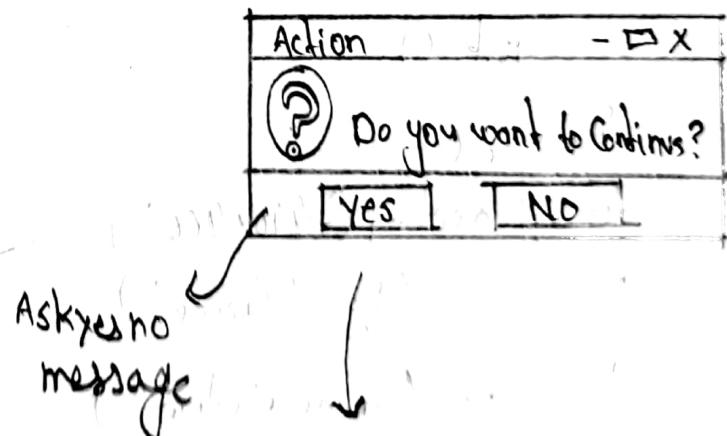
D) program :-

```
from tkinter import *
from tkinter import messagebox
def msgbox():
    messagebox.askyesno("Action", "Do you want to Continue?")
    messagebox.showerror("Error", "Can't use the process")
root = Tk()
root.config(bg="Green")
B1 = Button(root, text="messageBox"), bg="Red", command=msgbox)
B1.pack()
root.mainloop()
```

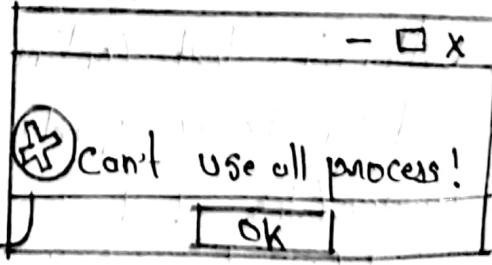
Output :-



click on the
button



Askyesno
message



show
error

A) Aim :- Program to traverse various windows using the button button widget. and add and remove window to the listbox.

Algorithm :- After defining function and method for traversing and adding removing window.

Step 1 : Import the element method from tkinter library.

Step 2 : Define a function and Create objects of given window by using the three method normally Config, title, minsize

Step 3 : Define a Button objects which will be placed on the current window to traverse and define another button which will be used to exit from the window and place to current window.

Step 4 : Define function which will be the quit method to terminate the program.

Step 5 : Now Create an objects of main window and use the various method like, Config, title, geometry, etc.

Step 6 : Define the two button objects which will be placed on to traverse on other window and the other to terminate the program.

step 7: Define another function on which will carry various button placed on third window. Define to a buttons respectively and use the grid method along with the two button.

Step 8 finally call the mainloop method.

E) program :

```

from tkinter import *
def main():
    root = Tk()
    root.geometry("440x460")
    root.config(bg="Yellow")
    root.title("window1")
    B1 = Button(root, text="Next", command=main1)
    B1.grid(ipadx=60, ipady=40, padx=10, pady=20)
    B2 = Button(root, text="Exit", command=term)
    B2.grid(ipadx=60, ipady=40, padx=10, pady=20)
def term():
    quit()
    tos = Tk()
    tos.geometry("440x460")
    tos.config(bg="Green")
    tos.title("mainwindow")
    B3 = Button(tos, text="Continue", command=main)
    B3.grid(ipadx=60, ipady=40, padx=10, pady=20)
    B4 = Button(tos, text="Exit", command=term)
    B4.grid(ipadx=60, ipady=40, padx=10, pady=20)

```

def main1():

44

top = Tk()

top.config(bg="yellow")

top.geometry("440x460")

~~mainloop()~~

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

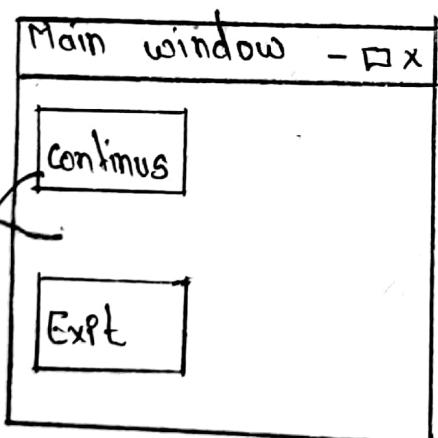
B1.grid(ipadx=260, ipady=40, padx=10, pady=20)

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

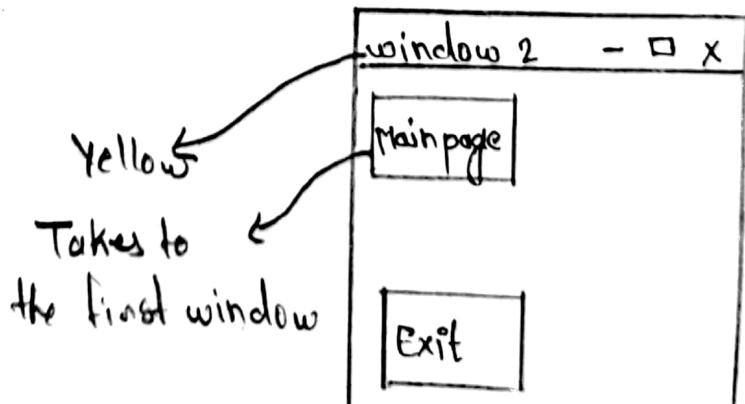
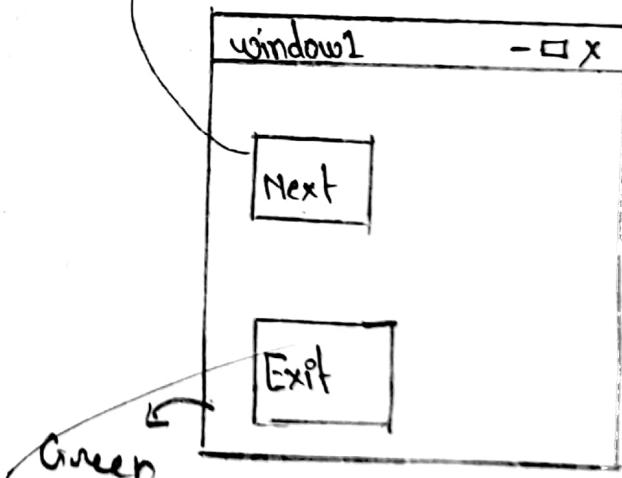
B2.grid(ipadx=60, ipady=40, padx=10, pady=20)

~~mainloop()~~

output:



Takes window 2



F) Aim = To write a Spin Box program briefly

Step 1: Create objects from the ~~tkinter~~ method and subsequently create an objects spin box method

Step 2: Make the objects created or to the parent window trigger the corresponding event

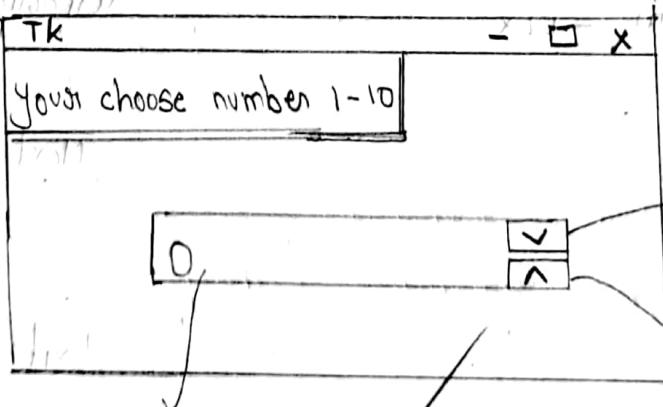
Step 3: Finally call the mainloop method.

```

22. Python GUI
      Tkinter
      from tkinter import *
      master = Tk()
      l1 = Label(master, text="Your choose number 1-10", bg="orange", fg="blue")
      l1.pack(side="top")
      s1 = Spinbox(master, from_=0, to=10, font=20, bg="pink", fg="Red")
      s1.pack(side="top")
      mainloop()

```

Output :-



Enter the number.

click and
left number
click and right
number

Q) * Paned window with orientation specified

Step 1: Create an objects from paned window and use the pack method with the attribute fill and expand by this code right side.

Step 2: Create an objects from the label method and put it onto the paned window with the text attribute and use the add method to embed the new objects.

Step 3: Similarly Create a second paned window object and add it onto the 1st paned window with orientation specified.

Step 4: Now Create an other label objects and place onto the 2nd paned window objects and add the onto the 2nd pane.

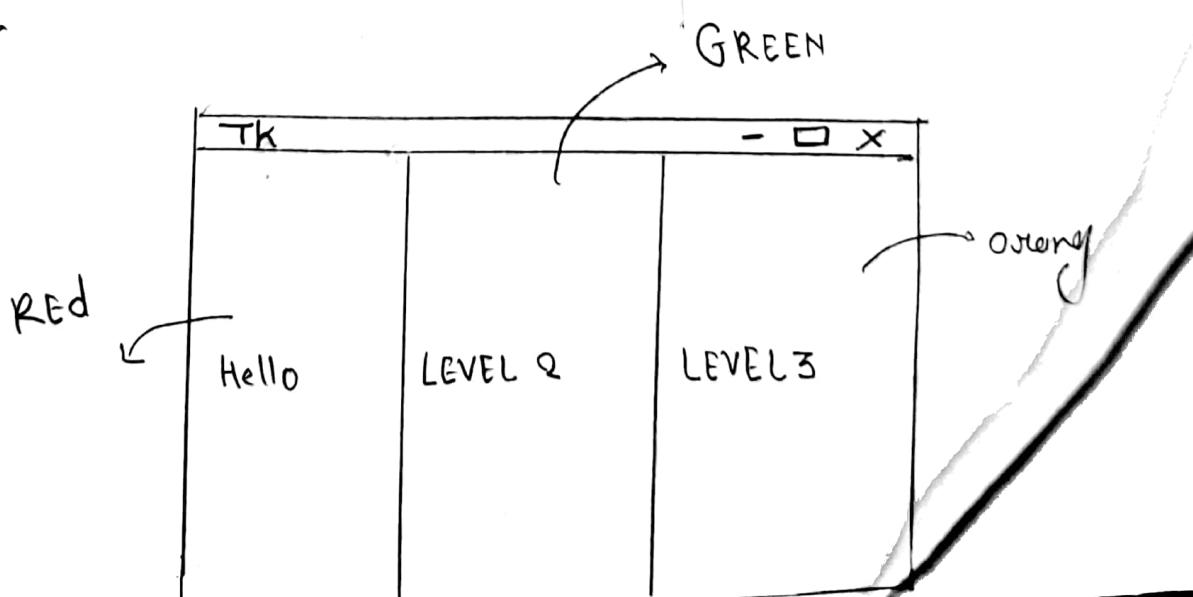
Steps: Now use the mainloop method to terminal onto.

Program :-

```

from tkinter import *
root = Tk()
p = PanedWindow(bg = "pink")
p.pack(fill=BOTH, expand=1)
l1 = p.Label(p, text = "HELLO", bg = "RED")
p.add(l1)
l2 = Label(p, text = "LEVEL 2", bg = "GREEN")
p.add(l2)
p1 = PanedWindow(p, orient=VERTICAL, bg = "yellow")
p.add(p1)
l3 = Label(p1, text = "LEVEL 3", bg = "orange")
p1.add(l3)
mainloop()
    
```

Output :-



* Canvas widget

Step 1: use the tkinter method and Create an objects from the canvas method and use the attribute height , weight , by colour and the parent window objects

Step 2: use the method create oval , create line and arc along with canvas objects so Create and use the co-ordinate value . Also use fill attribute to assign various colour.

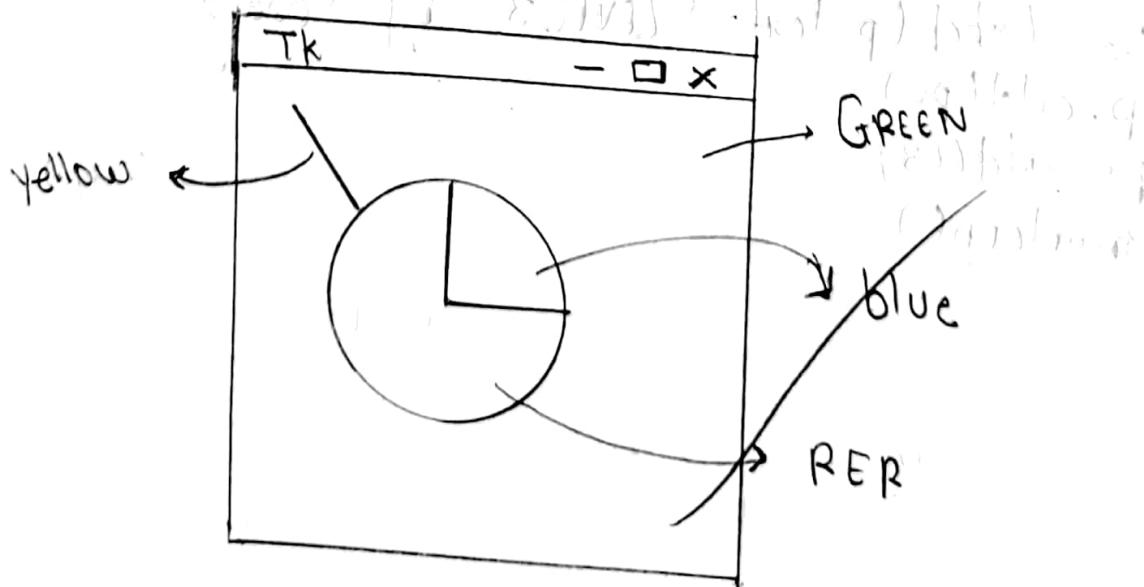
Step 3: Now call pack and mainloop method .

Jn 17/02

program :-

```
from tkinter import *  
root = Tk()  
c1 = Canvas(root, height=5000, width=5000, bg="GREEN")  
oval = c1.create_oval(300, 12, 12, 400, fill="RED")  
line = c1.create_line(30, 20, 70, 60, fill="yellow")  
arc = c1.create_arc(300, 12, 12, 400, fill="blue")  
c1.pack(side=TOP)  
mainloop()
```

output :-



PRACTICAL - 6

Aim : To make use a varieties of Library

Step 1 :- import db library and use the open method for creating the database by specifying name of the database along with the corresponding flag.

Step 2 :- use the objects for accessing to given web size and the corresponding regulation for the web size

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

```
import dbm  
db = dbm.open("data base", flag = "c")  
if db["WWW"] != None:  
    print("good")  
else:  
    print("Not good")  
db.close()
```

Output:

good



- 7) a) Aim : To make use of the various types of database statements.
- Step 1: Import the Corresponding Library taking of database connection.
- Step 2: Now Create Connection objects using SQLite library and connecting method for Create the new database.
- Step 3: Now Create the Cursor objects using Cursor method from the Connection objects created in steps.
- Step 4: Now use the executing method for Creating the table by with the column name and respective data type.
- Step 5: Now with the Cursor objects use insert statements for entering the values Co-ordinating the different field considering the data types.
- Step 6: use the Commit method to Complete the transaction using the Connection objects.

Step 7: Use the execute statement along with the Cursor objects for accessing the value the database using selecting from when clause.

Step 8: Finally use the fetchall method for display the value for the table using the Cursor objects

Step 9: Use the execute method and the drop table Syntax for terminating the database finally use the close method.

Log object for result handle will be created

and the number of rows affected is zero

program :-

```

import sqlite3
connection = sqlite3.connect("student.db")
c1 = connection.cursor()
c1.execute('Create table student(Name, RNO, DOB)')
c1.execute('insert into student values("Rakesh",
                                         1840, 23-06-2002)')
c1.execute('insert into student values("Sachin", 1841,
                                         25-04-1996)')
c1.execute('insert into student values("Manoj", 1842,
                                         02-02-2000)')
connection.commit()

c1.execute('Select * from student')
for c1.fetchone()
c1.execute('Drop table student')

```

32

Output :-

[('Rakesh', 1840, 23-06-2002), ('Sachin', 1841, 25-04-1996)]

[('Manoj', 1842, 02-02-2000)]

(('Rakesh', 1840, 23-06-2002), ('Sachin', 1841, 25-04-1996))

((('Manoj', 1842, 02-02-2000), ('Rakesh', 1840, 23-06-2002)), ('Sachin', 1841, 25-04-1996))

((('Rakesh', 1840, 23-06-2002), ('Sachin', 1841, 25-04-1996)), ('Manoj', 1842, 02-02-2000)))