PERFORMANCE

Term	Remarks	Staff Member's Signature
I		
II	good completed	

Scanned by TapScanner

Exam Seat No. _____

THAKUR COLLEGE OF SCIENCE & COMMERCE

NAAC Accredited with Grade "A" (3rd Cycle)

THAKUR TRUSTS

ISO 9001 : 2015 Certified

Degree College

Computer Journal

CERTIFICATE

SEMESTER II UID No. _____

Class FYBSC-CS Roll No. 1757 Year 2019-20

This is to certify that the work entered in this journal is the work of Mst. / Ms. Priyanka Kumari Sao

who has worked for the year 2019-20 in the Computer Laboratory.

Teacher In-Charge

Head of Department

Date : _____

Examiner

Scanned by TapScanner

☆☆ INDEX ☆☆

No.	Title	Page No.	Date	Staff Member's Signature
1.	Demonstrate the use of the different file accessing mode different attribute of the file object and differentiate among the different read Method	17-20		Not Submitted
2.	Demonstrate the use of iter() and next() method along with a class to display the odd numbers upto 100.	20-21		
3.	A:- Write a program using the exception block related to environment error.	24-25	16/12/19	
	B:- Write a program for demonstrative then use of value error in the given program statement.	25-26		

☆☆ INDEX ☆☆

No.	Title	Page No.	Date	Staff Member's Signature
4.	Regular expression	26-30	6/01/20	
5.	Graphical user Interface	31-35	12/01/20	
	1> Parent Window			
	2> Radio Button			
	3> Scroll bar			
	4> message box			
	5> Traversing			
	6> Spin box			
	7> Paned Window			
	8> Canvas Widget			
6.	(a) Demonstrate the use of database connectivity.	42-43		
	(b) Creating the table and inserting value.	43-45		

Scanned by TapScanner

PRACTICAL-01

AIM : Demonstrate the use of the different file accessing mode different attribute of the file object and differentiate among the different read Method.

Step1: Create a file object by using open Method and use the write access Mode followed by writing some contains of the file and closing the file.

Step2: open the file on read Mode and use the read Method or read line Method or read lines Method and store the output on the variable and finally display the content of that variable.

Step3:- Use the file obj for finding the name of the file, the mode in which the file will be open, whether the file is still open or closed and finally the output of the softspace attribute.

Scanned by TapScanner

```
Fileobj = open("abc.txt", "w") # file open (write mode)
file obj.write("I am Priyanka"+"/n")
file obj.write("my rollno is 1757/n now I am doing
                fybsc cs/n it's my first practical/n")#file
                                                write
file obj.close() # file close
fileobj= open("abc.txt","r") # read mode
# read()
str1 = fileobj.read()
Print("the output of read method is:", str1)
fileobj.close()
```

>>> the output of read method is: I am Priyanka/n my rollno is 1757 /n now i am doing fyBSC cs/n it's my first Practical/n

```
#readline()
fileobj = open("abc.txt","r")
str2 = file obj.readline()
Print("the output of read method is:", str2)
fileobj.close()
```

>>> The output of read method is: I am Priyanka/n my rollno is 1757/n Now i am doing fybsc cs/n it's my first Practical/n

```
# readlines()
fileobj = open("abc.txt","r")
str3 = fileobj.readlines()
Print("the output of read Method is:", str3)
file_obj.close()
```

>>> the output of read Method is: ["I am Priyanka/n my rollno is 1757 /n now i am doing fybsc cs/n it's my first Practical/n"]

```
a=fileobj.name
Print(" the output of name is :",a)
fileobj.close ( )
b= fileobj.closed
Point(" the output of closed is :",b)
fileobj.close( )
c= fileobj.mode
Print(" the output of mode is :",c)
fileobj.close( )
d= fileobj.softspace
Print(" the output of softspace is :",d)
fileobj.close( )
```

```
>>>> the output of name is :abc.txt
>>> the output of closed is : True
>>> the output of mode is : r
>>> the output of softspace is : 0
```

```
# w+ mode
fileobj = open("abc.txt","w+")
fileobj.write ("Saurabh")
fileobj.close ( )
```

```
# r+ mode
fileobj =open("abc.txt","r+")
str1 = fileobj.read(6)
Print(" output of r+",str1)
fileobj.close ( )
```

```
>>> ('output of r+', 'saurabh')
```

```
# write mode
fileobj = open("abc.txt",
"w")
fileobj.write("DBMS")
fileobj.close()
```

```
# read mode
fileobj=open ("abc.txt",
"r")
str 2 =fileobj.read()
Print(" output of
read mode ! ",str2)
```

```
>>>('output of read
mode!', 'saurabh')
```

Scanned by TapScanner

Step 04: Now open the fileobject in write mode write some another content close subsequently then again open the fileobject in 'wt' mode that is the update mode and write contents.

Step 05: open fileobject in read mode, display the update written contents and close open again in 'rt' mode with parameter passed and display the output subsequently.

Step 06: Now open fileobject in append mode open write mode write content close the fileobject again open the fileobject in read mode and display the append output.

Scanned by TapScanner

Step 07 : open the fileobject in read Mode, declare a Variable and Perform fileobject dot tell method and store the output consequently in Variable.

Step 08 : Use the seek Method with the arguments with opening the fileobject in read Mode and Using subsequently.

Step 09 : Open fileobject with readmode also use the readlines Method and store the output subsequently in and print the same for counting the length use the for condition statement and display the Length.

Scanned by TapScanner

```
# append mode
fileobj =open("abc.txt", "a")
fileobj. write("data structure")
fileobj.close( )
fileobj = open("abc.txt", "r")
str3 = fileobj.read ( )
Print(" output of append mode :", str3)
fileobj.close( )
>>>('Output of append mode :', 'Saurabh', 'datastructure')
# tell( )
fileobj =open("abc.txt", "r")
Pos = fileobj. tell( )
Print("tell( ) :", Pos)
fileobj.close
>>>('tell( ) :', Pos)
# seek( )
fileobj =open("abc.txt", "r")
str4 = fileobj.seek(0,0)
str8 = fileobj.read(10)
Print("The beginning of the file :", str8)
# Finding length of different lines exist within lines
file obj = open("abc.txt", "r")
str 9 = file obj. read lines( )
Print(" output: ", str9)
For line in str9:
        Print(len(line))
fileobj.close( )
>>>('output :', ["college database'])
```

Nafl
9/12/19

Practical-02

(1) Aim: Demonstrate the use of iter() and next() method along with a class to display the odd numbers upto 100.

(1) Algorithm:

Step 1: Define a class and within that define the iter Method which will initialize the first element within the container object.

Step 2: for extracting the next element from the container with an argument and compare the number of elements require in a ~~computer~~ container by using the conditional statement.

Step 3:
Now create an object from the given class and pass this object as argument to the iter() method.

Step 4: Now using the conditional statement display all the values from the given container.

Scanned by TapScanner

```
class odd:
    def __iter__(self):
        self.num = 1
        return self
    def __next__(self):
        if self.num <= 100:
            num = self.num
            self.num += 2
            return num
        else:
            raise StopIteration
myobj = odd()
myiter = Iter(myobj)
For num in myiter:
    Print(num)
```

<u>Output</u>:

```
1
3
5
7
9
11
.
.
.
97
99
```

(B) Algorithm :

Algorithm :

(2) Program to display the elements of list as even and odd

Step 1 :

Define a function named as evenodd with an argument which would be an integer.

Step 2 : Define the logic for determining the numbers as even or odd appropriately

Step 3 :

Declare a list variable and define all the numbers that you want in the list.

Step 4 : Use map() method with the above function and the list variable as Arguments.

Step 4 : Print the result.

(3) Program for Printing the square and cube of given set of numbers using the map().

Step 1 : Define a Square function with as Argument and return the square of the number.

Step 2 : Define a function cube with an

code / output :

(2)
```
def evenodd(n):
    if n%2 == 0:
        return "Even"
    else:
        return "odd"

listnum = [0,4,5,7,9,11,13,5,20,9,25]
listI = list(map(evenodd, listnum))
Print(listI)
```

```
>>>
['Even', 'Even', 'odd', 'odd', 'odd', 'odd', 'odd',
 'odd', 'Even', 'odd', 'odd']
```

(3)
```
def square(x):
    return(x**2)
def cube(x):
    return(x**3)
FunCI = [square, cube]
For r in range(5):
    valueout = list(map(lambda x : x(r), FunCI))
    Print(valueout)
```

```
>>>
[1,1]
[4,8]
[9,27]
[16,64]
```

argument and return the cube of the number.

<u>step3</u>: Declare a list variable and call the functions square and cube in the list.

<u>step4</u>: Use for conditional statement and use the map() to find the square and cube given set of numbers.

1) Simple use of iter() method.

create a tuple object and assign it the value <u>or</u> element, then create a iterable object to iter through the tuple with iter() method. Now use to conditional statement to display all the elements in the iterable object.

16/12/17

(5) myfupleI = ("Banana", "Orange", "Apple")
myiterI = iter(myfuple I)
for i in myiterI:
Print(i)

>>> Banana
orange
Apple

Practical : 3

Aim : Write a program using the exception block related to environment error.

Algorithm :

Step I : Use the tryblock to defining the normal course of action.

e.g : define the file object and open the file in the write mode and write some containts of the file.

Step2 : Use the except block with the Io error as an environment error and convey the appropriate message to the user else display the message that operation is carried out successfully.

Scanned by TapScanner

try:

```
    FO=open("abc.txt", "w")
    FO.write("I am Priyanka")
except IoError:
        Print("Error occured under Environment
                error")
else:
    Print("operation successfull")
```

>>> operation successfull

Scanned by TapScanner

(B) Aim: Write a program for demonstrative the use of value error in the given program statement.

Alagorithm:

Step I: Execept the value from the user and if is the valid value display the entered value and terminate the condition by using the break statement.

Step 2: define the execept block with the value error as a Keyword and display the appropriate Message.

Step 3: We can define the multiple exception using the execept statement for finding the different catagrie of error.

Scanned by TapScanner

```
try:
  x=int(input("enter a no"))
  fo=open("abc.txt","w")
  fo.write("I am priyanka")
except valueError:
        Print("It is a language")
except IOError:
        Print("operation successfull")
else:
  Print("default action")
```

```
>>> enter a number45
  default action
```

```
while true:
    try:
        file obj = open("Priyanka","w")
        file obj=write("All indians any sisters")
        a=int(input("Enter a number"))
        Print(a)
        break
    except IOError:
            Print("There is an environmental error")
    exceptValueError:
            Print("The value is invalid")
```

<u>Output</u>

```
>> Enter a number = Priyanka
   The value is invalid
   Enter a number: Paiy
   The value is invalid
   Enter a number :56
```

Practical-4

Topic : Regular expression

Step1 : Import re module declare pattern and declare sequence use match method with declare arguments if arguments matched than print the same otherwise print Pattern NOT FOUND!

step2 : Import re module declare pattern with literal and meta character Declare string value. use the findall() with arguments and print the same.

step3 : Import re moduls declare pattern with meta character use the split() and print the output.

Scanned by TapScanner

~~# import ( )~~

match ( )

```
import re
Pattern = r"FYCS"
Sequence = "FYCS represents computer science stream"
if re.match(Pattern, sequence):
    Print("matched pattern found!")
else:
    Print("NOT Found!")
```

>>> matched pattern found!

numerical values (segreation)

```
import re
Pattern = r'\d+'
String = 'hello123, howdy789, 45howru'
Output = re.findall(Pattern, string)
Print(output)
```

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

Split ( )

```
import re
Pattern = r'\d+'
String = 'hello123, howdy789, 45howru'
Output = re.split(Pattern, string)
Print(output)
```

>>> ['hello', ', howdy', ', ', 'howru']

Scanned by TapScanner

Step4: import re module declare string and accordingly declare pattern replace the blank space with no-space. Use scale() with 3 arguments and print the string without spaces.

Step5: import re module declare a sequence use search Method for finding subsequently use the group() with dot operator as search() gives memory location using group() it will show up the matched string.

Step6: import re module declare list with numbers.
Use the conditional statement here ~~use~~ 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 numbers are equal to 10. if criteria matches print all number matches otherwise print failed.

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

# group()
import re
sequence = 'Python is an interesting language'
v = re.search('\APython', sequence)
print(v)
v1 = v.group()
print(v1)
>>> <-Sre.SRE-Match object at 0x0281DF00>
    Python

# verifying the given set of phone numbers.
import re
list1 = ['8004567891', '9145673210', '7865432981',
         '9876543201']
For value in list1 :
    if re.match(r'[8-9]{1}[0-9]{9}',
    value or len(value)==10) :
    print("criteria matched for cell number!")
else:
    print("criteria failed!")
```

Step7: import re module declare a string use the module with findall() for finding the vowels in the string and declare the same.

Step8: import re module declare the host and domain name declare pattern for separating the host & domain name. Use the findall() and print the output respectivelly.

Step9: import re module enter a string use pattern to display only two elements of the particular string. Use findall() declare two variables with initial value as zero use for condition and subsequently use the if condition check whether condition satisfy add up the are else increament value. And display the values subsequently.

```
>>> criteria matched for cell number
    criteria matched for cell number
    criteria failed!
    criteria matched for cell number
```

Vowels

```
import re
str1 = 'plant is life overall'
output = re.findall(r'\b[qeiouAEIOU]\w+', str1)
Print(output)
>>>['is', 'overall']
```

host & domain

```
import re
seq = 'abc.tcsc@edu.com, xyz@gmail.com'
Pattern = r'[\w\.-]+[\w\.-]'
output = re.findall(Pattern, seq)
Print(output)
>>>['abc.tcsc', 'edu.com', 'xyz', 'gmail.com']
```

Counting of first 2 Letters:

```
import re
s = 'mr.a, ms.b, ms.c, mr.f'
P = r'[\ms\mr\]+
o = re.findall(p,s)
Print(o)
m = 0
f = 0
for v in o:
```

```
if(v == 'ms'):
    f = f+1

else:
    m = m+1

print("No of males is :", m)
print("No of females is :",f)

>>> ['mr', 'ms', 'ms', 'mr']
('No. of males is :'2)
('No. of females is :', 2)
```

Scanned by TapScanner

Practical-05

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

Algorithm:

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

Step 2: Create a Variable from a text Variable & Position it onto the Parent Window.

Step 3: Use the Pack() along with the object created from text Method & use the Parameter.
1> Side = TOP, Padx = 20, ipadx = 40, ipady = 50

Step 4: Use the mainloop Method for triggering corresponding event.

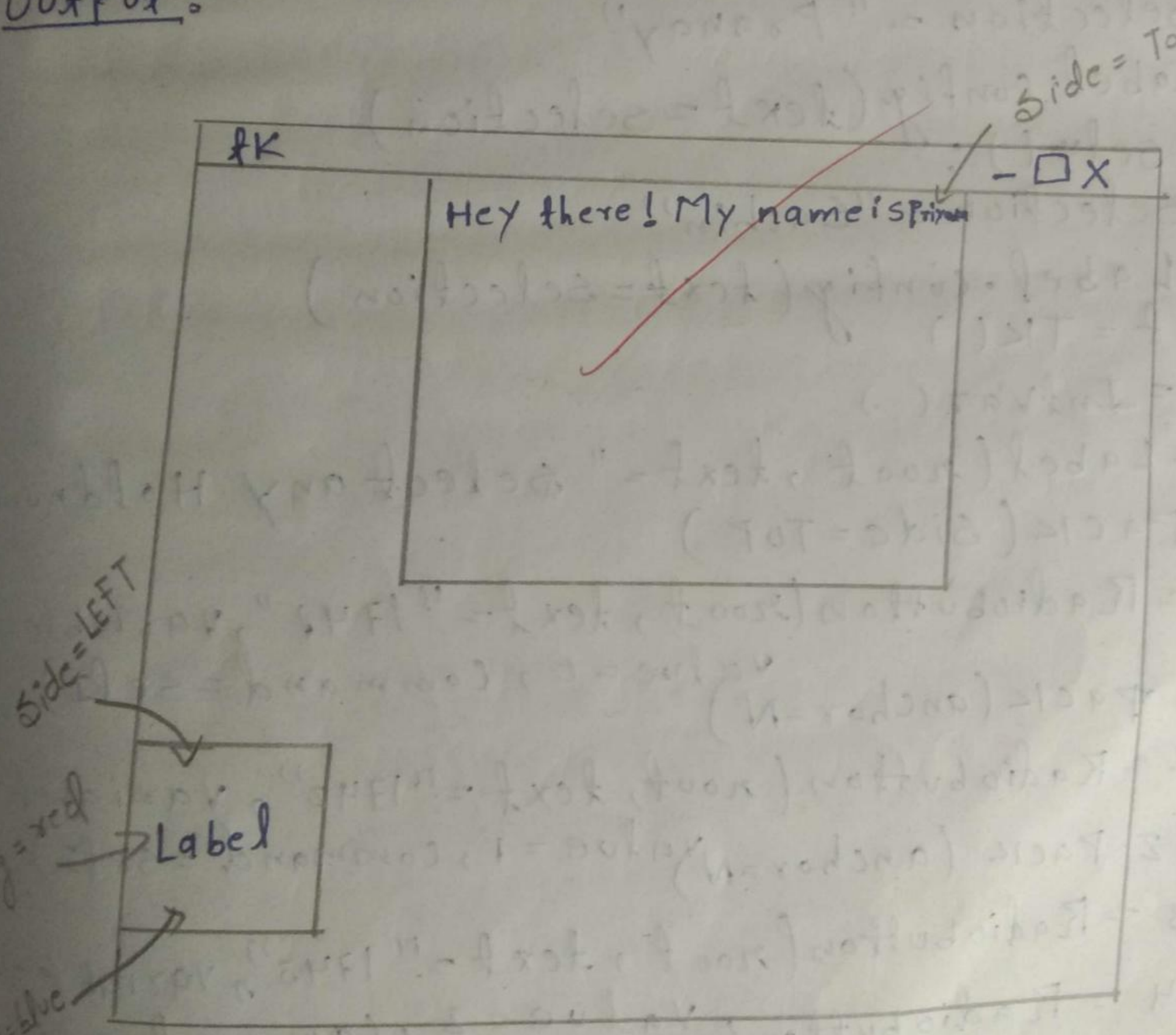
Step 5: Now repeat above step with a Label Method which takes the following argument.
1> Name of Parent Window
2> The text attribute which defins the string.
3> The background colour (bg).

iv> Now : Use the Pack() with the foreground colour (fg) relevant attributes.

Program

```
from tkinter import *
root = TK( )
T1 = Text (root)
T1.insert(END, "Hey There! My Name is Priyanka.")
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, pady=10, ipadx=20, ipady=30)
root.mainloop( )
```


Output:



Program :

```
import
from tkinter import *
def sel1 ():
    selection = "NeeraJ"
    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 Holdnumber")
L1.pack(side=TOP)
R1 = Radiobutton(root, text = "1741", variable=var,
                value = 0, command = sel1)
R1.pack(anchor=N)
R2 = Radiobutton(root, text = "1742", variabel=var,
R2.pack(anchor=N) value = 1, command = sel2)
R3 = Radiobutton(root, text = "1743", variable = var,
R4 = Radiobutton(value = 2, command = sel3)
                root, text = "1744", variable = var
R4.pack(anchor = N) value = 3, command = sel4)
Label = Label(root)
Label.pack(side = BOTTOM)
```

Aim: To make use of RadioButton widget for selection of one on the multiple option.

Algorithm:

Step 1: Use the tkinter Method to import the relevant 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 of Call the variable as an argument within Method.

Step 4: Now define the Parent window of define option using control variable.

Step 5: Now Create object of Radio Button which will take following arguments:

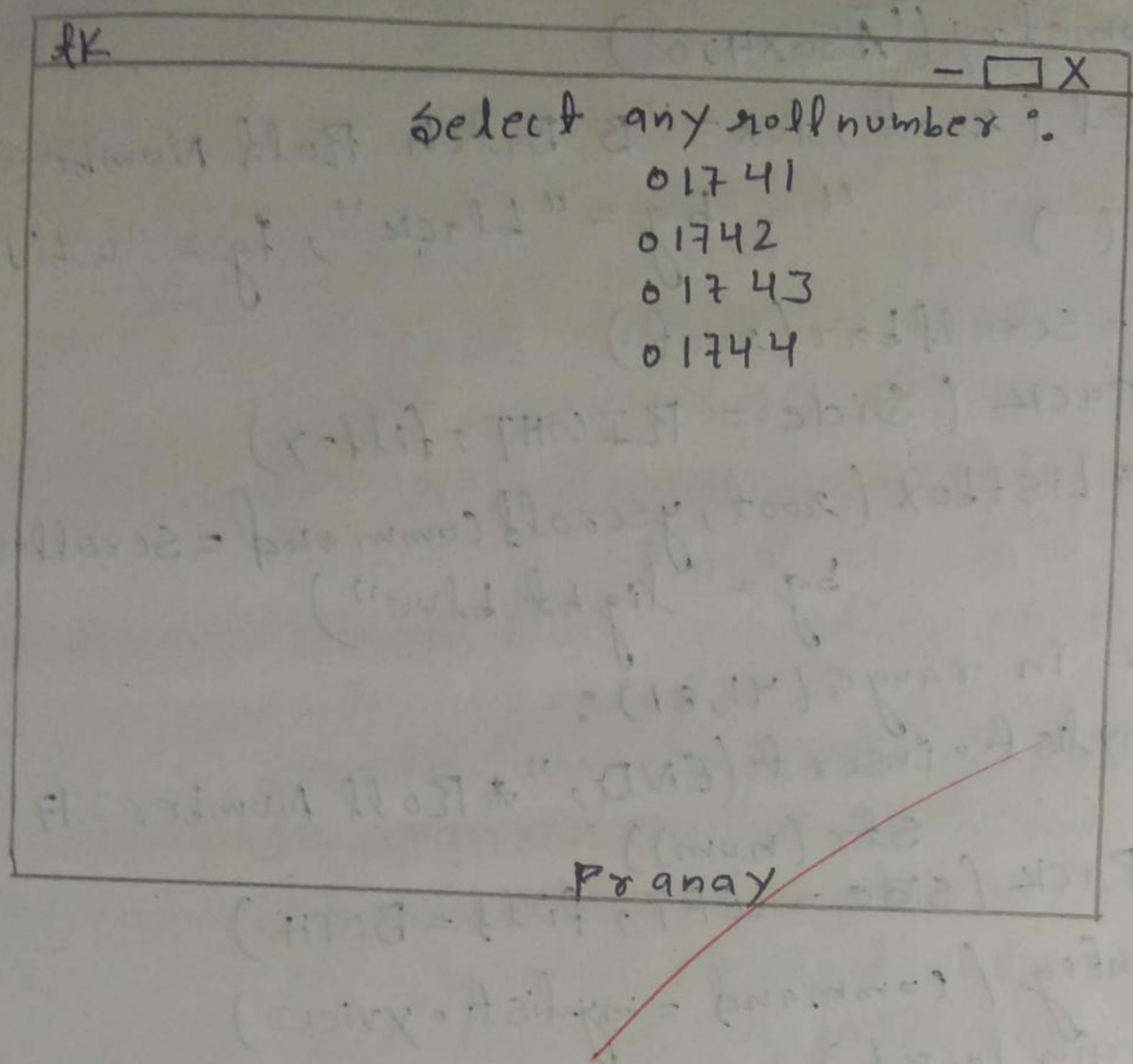
1) Positioning on Parent window
2) Text variable
3) Define variable argument
4) Corresponding value and trigger the given function.

Step 6: Now call the Pack Method for corresponding Radio object to created and specify argument as an anchor attribute.

Step 7: Now define a Label object and place in onto parent window using Pack Method & finally use mainloop Method.

root.mainloop( )

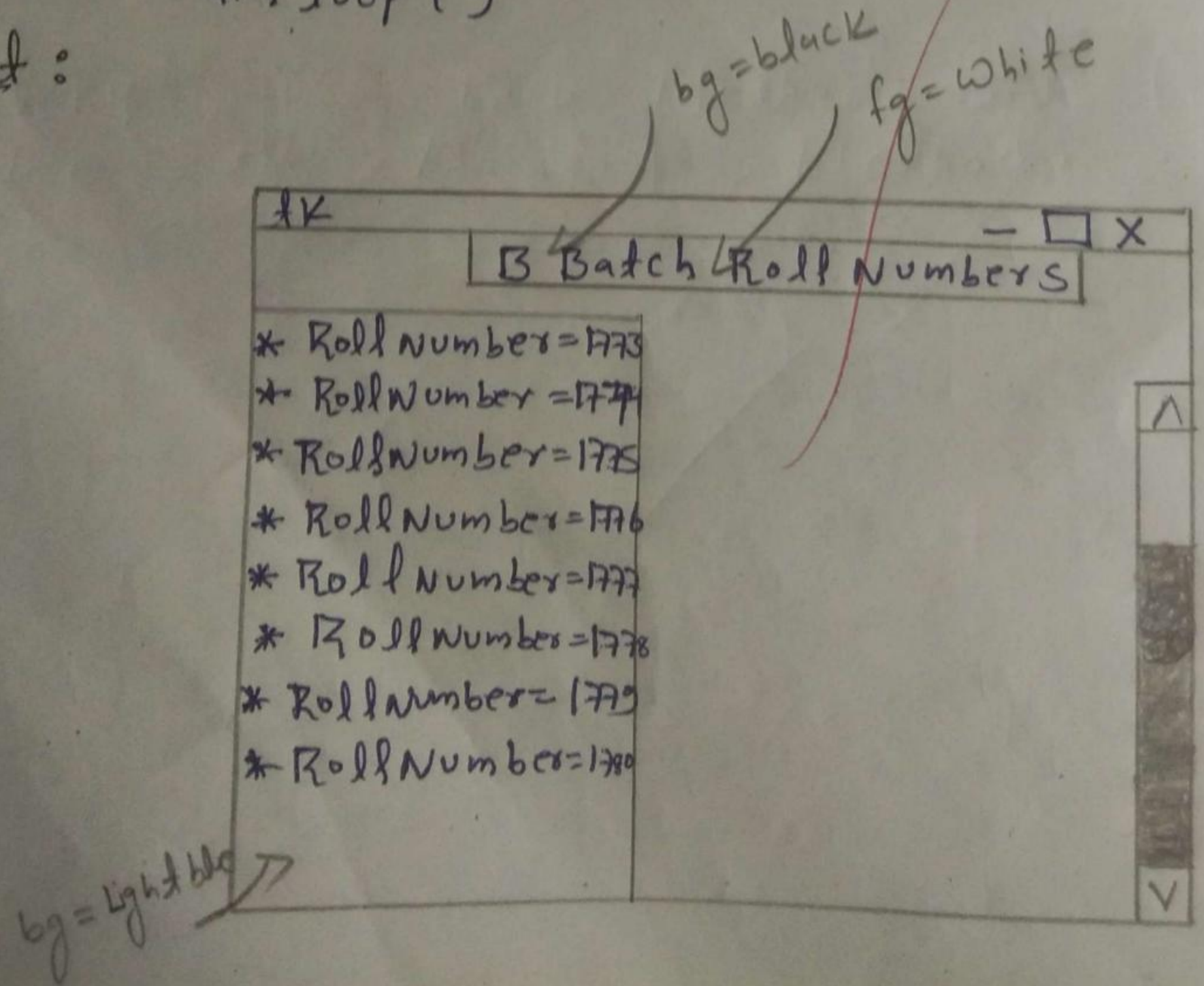
output:



Program:

```
from tkinter import*
root = TK()
root.geometry('450x400')
L = Label (root, text = "B Batch Roll Numbers:
                    ", bg = "black", fg = "white")
L.pack()
scroll = scrollbar(root)
scroll.pack (side = RIGHT, fill=y)
mylist = Listbox (root, yscrollcommand = scroll.set,
                  bg = "light blue")
for num in range(41,81):
        mylist.insert(END, "* Roll Number : 17",
                  str(num))
mylist.pack (side = LEFT, fill = BOTH)
scroll.config (command = mylist.yview)
root.mainloop()
```

Output:



Aim: To make use of Scroll Bar widget of the GUI application.

Algorithm:

Step1: Import tkinter Library to use Scroll Bar widget.

Step2: Create an object corresponding to scroll parent window & create an object from scroll bar & place it on the parent window so created.

Step3: Create an object of Label method to provide a heading and place it on parent window.

Step4: Use pack Method along with object of Scrollbar Method & use argument side & fill.

Step5: Create an object of ListBox Method and place it on to parent window with attribute yscroll command.

Step6: Use for loop to insert values in the object of List Box by using insert method.

Step7: Use Config Method along with scroll bar objects & use command attribute.

Step8 = Finally call the mainloop Method

8.

Aim: To make use of messagebox Method of the GUI application.

Algorithm:

Step1: Import relevant Method from tkinter Library.

Step2: Define a function and use Message box along with different Methods available which contains one or More arguments.

Step3: Create an object from button Method and place it onto the parent window with text and command attribute specified.

Step4: Use Pack Method and finally use the Mainloop Method.

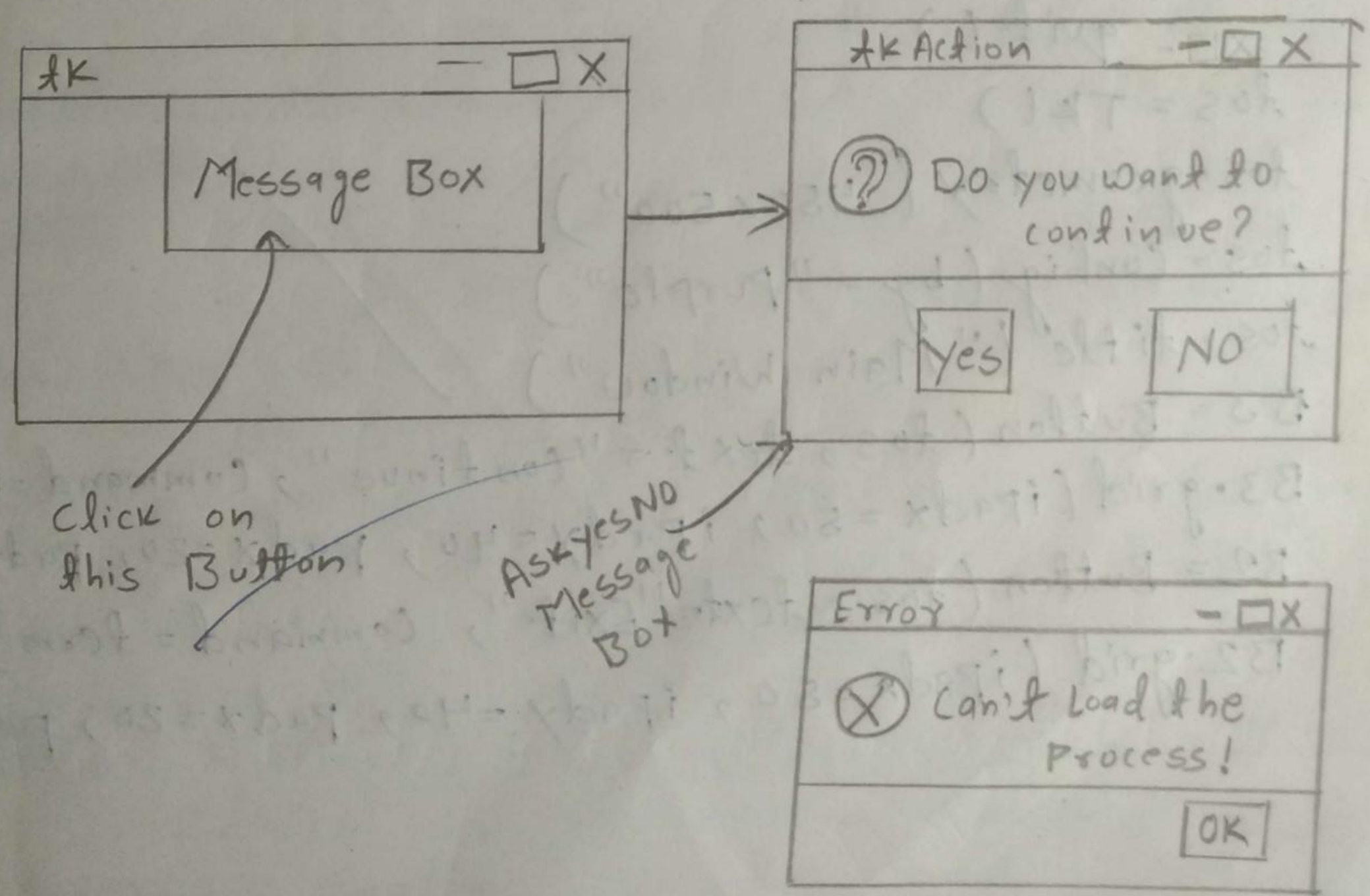
Scanned by TapScanner

Program:

```
From tkinter import*
from tkinter import Messagebox
def msgbox():
    Messagebox.showaskyesno("Action", "Do you want to continue")
    messagebox.showerror("Error", "Can't Load the Process!")

root=Tk()
root.config(bg="grey")
B1 = Button(root, text="Message Box", bg="Light blue", command=msgbox)
B1.Pack()
root.mainloop()
```

Output:



Aim: Program to traverse various windows using the button widget.

Algorithm:

Step1: Import the relevant Method from tkinter Library.

Step2: Define a function and create a object of given window by using the three Methods namely config, tittle, Minsize.

Step3: Define a Button object which will be placed on the current window to traverse and define another button which will be used to exit from the window and place it onto current window.

Step4: Define another function which will use the quit Method to terminate the program.

Step5: Now create an object of Main window and use various Methods like Config, tittle, geometry etc.

Step6: Define two buttons which will be placed on the main window; onto traverse another window and the other to terminate the Program.

Scanned by TapScanner

Program:

```
From tkinter import *
def main ():
    root = TK ( )
    root.geometry ("450 X 500")
    root.Config (bg = "Light green")
    root.title ("Window I")
    B1 = Button (root, text = "Next", command =
                main I)
    B1.grid (ipadx=50, ipady = 40, padx=20,
                pady=30)
    B2 = Button (root, text = "Exit", command=term)
    B2.grid (ipadx = 50, ipady=40, padx=20, pady=30)
def term():
    quit ( )
tos = TK( )
tos.geometry ("450 x 500")
tos.Config (bg = "purple")
tos.title ("Main Window")
B3 = Button (tos, text = "Continue", command = Main)
B3.grid (ipadx = 50, ipady = 40, padx = 20, pady=30)
B2 = Button (tos, text="Exit", command = term)
B2.grid (ipadx = 50, ipady = 40, padx = 20, pady=30)
```

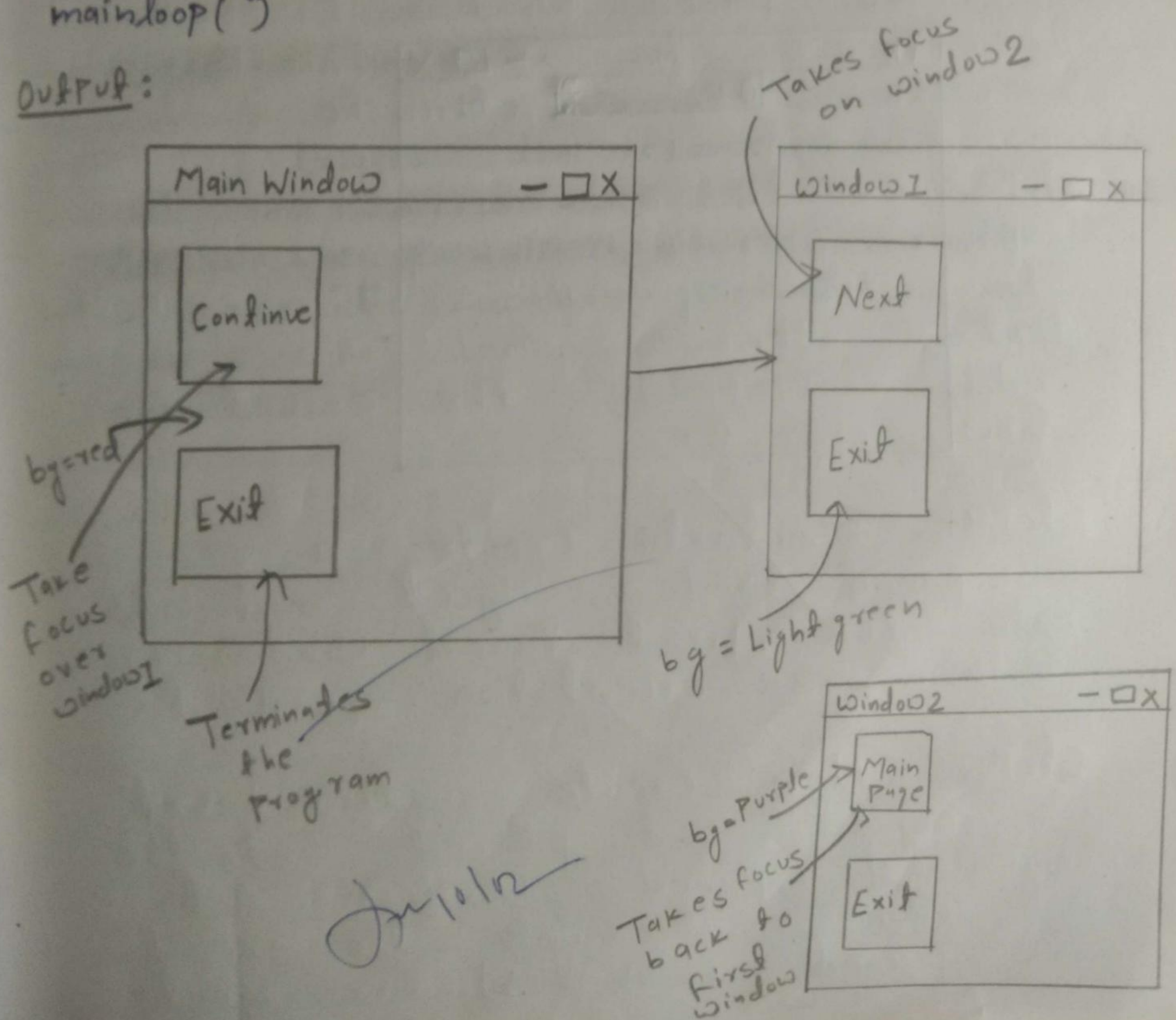
Step7: Define another function which will carry various button placed on third window. Define two buttons respectively and use the grid Method along with the two buttons.

Step8: Finally call the Mainloop Method.

Scanned by TapScanner

```
def main():
    top=TK()
    ~~top=TK()~~
    top.geometry("450x500")
    top.config(bg="purple")
    top.title("Window2")
    B4 = Button(top, text="Main Page", command=Main)
    B4.grid(ipadx=50, ipady=40, padx=20, pady=30)
    B2 = Button(top, text="Exit", command=term)
    B2.grid(ipadx=50, ipady=40, padx=20, pady=30)
mainloop()
```

<u>Output</u>:



10/12

Aim: To make the use of spinbox method of the GUI application.

Step1: Import relevant method from tkinter Library.

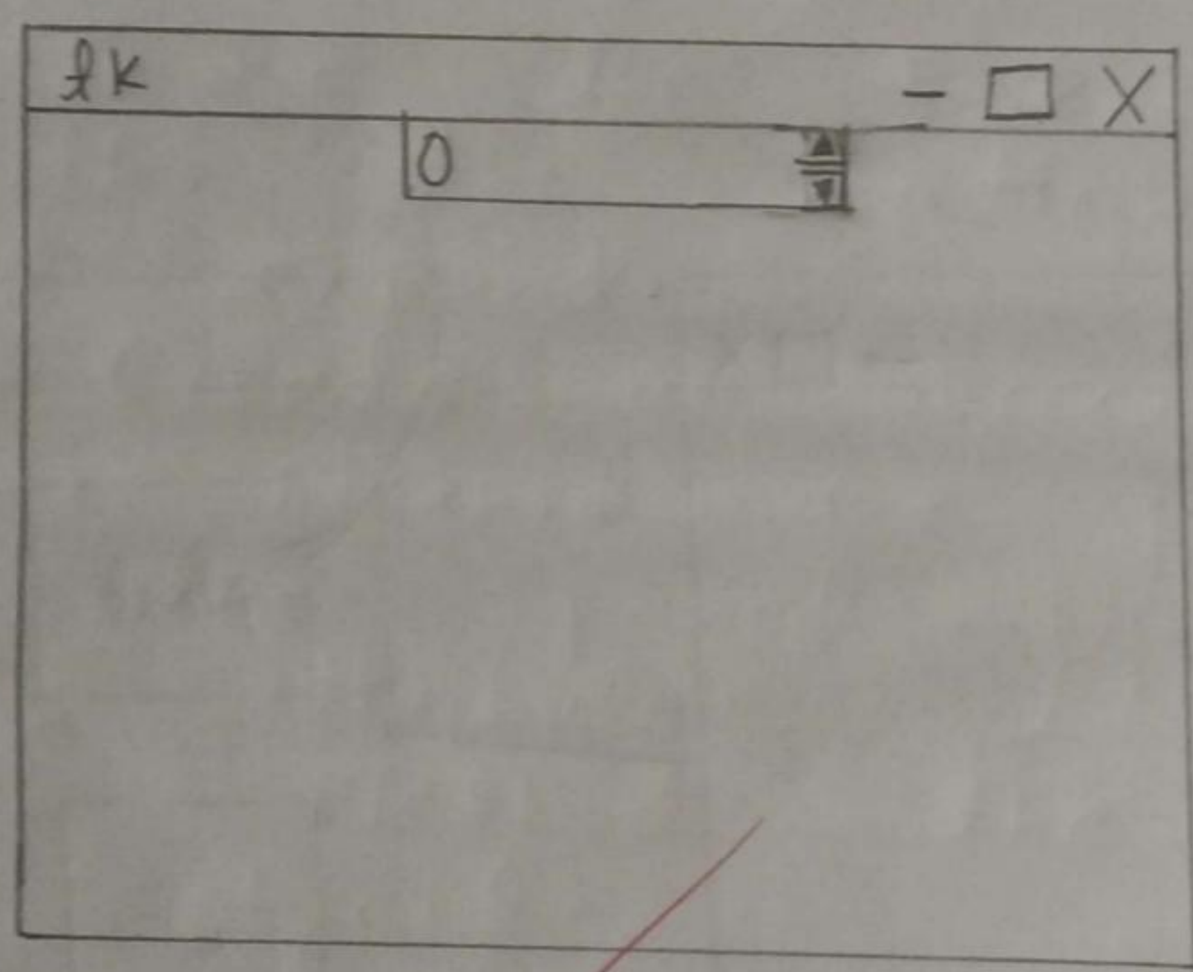
Step2: Create an object from the tk Method and subsequently create an object from the String spin box Method.

Step3: Make the object so created on the Parent window and trigger the corresponding event.

Step4: Use Pack Method and finally use the mainloop Method.

```
From tkinter import*
master = Tk( )
S1 = Spinbox(master, from_ = 0, to=10)
S1.Pack( )
mainloop( )
```

Output:



Aim: To make the Use of Paned Window Method of the GUI application.

Step1: Import relevant method from tkinter Library.

Step2: Create an object from the Paned Window Method and use the Pack Method with the attribute fill and expand.

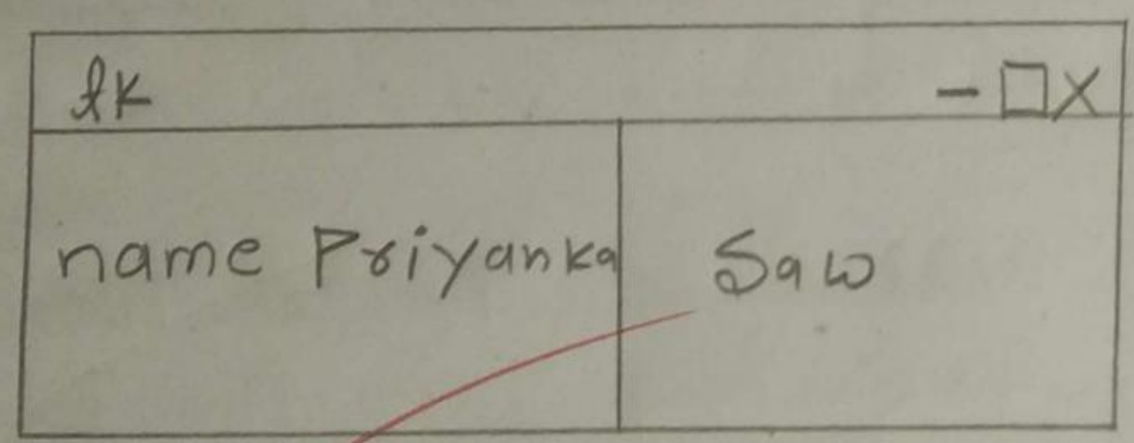
Step3: Create an object from the Label Method and put it on to the paned Window with the text attribute and use the add Method the embed the new object.

Step4: Similarly create a second paned Window object and add onto the first Paned Window object with ~~spec~~ orientation specified.

Step5: Now create another Label object and placed onto the second Paned Window and add it onto the second paned ~~Method~~ Window object.

```
From tkinter import*
P= PanedWindow( )
P.pack(fill= BOTH, expand=10)
L= Label( P, text= "name")
P.add(L)
P1= PanedWindow(P, orient= VERTICAL)
P.add(P1)
L1= Label(P, text= " Priyanka")
P.add(L1)
L2 = Button(P, text = "Saw")
P.add(L2)
mainloop( )
```

Output :



Step6: Now create another button object and placed onto the third paned window and add it onto the third paned window object.

Step7: Trigger the Mainloop Method.

Aim: Canvas Widget

Step1: Use the tkinter method and create an object from the canvas method and use the attribute height, weight, bgcolour and the parent window object.

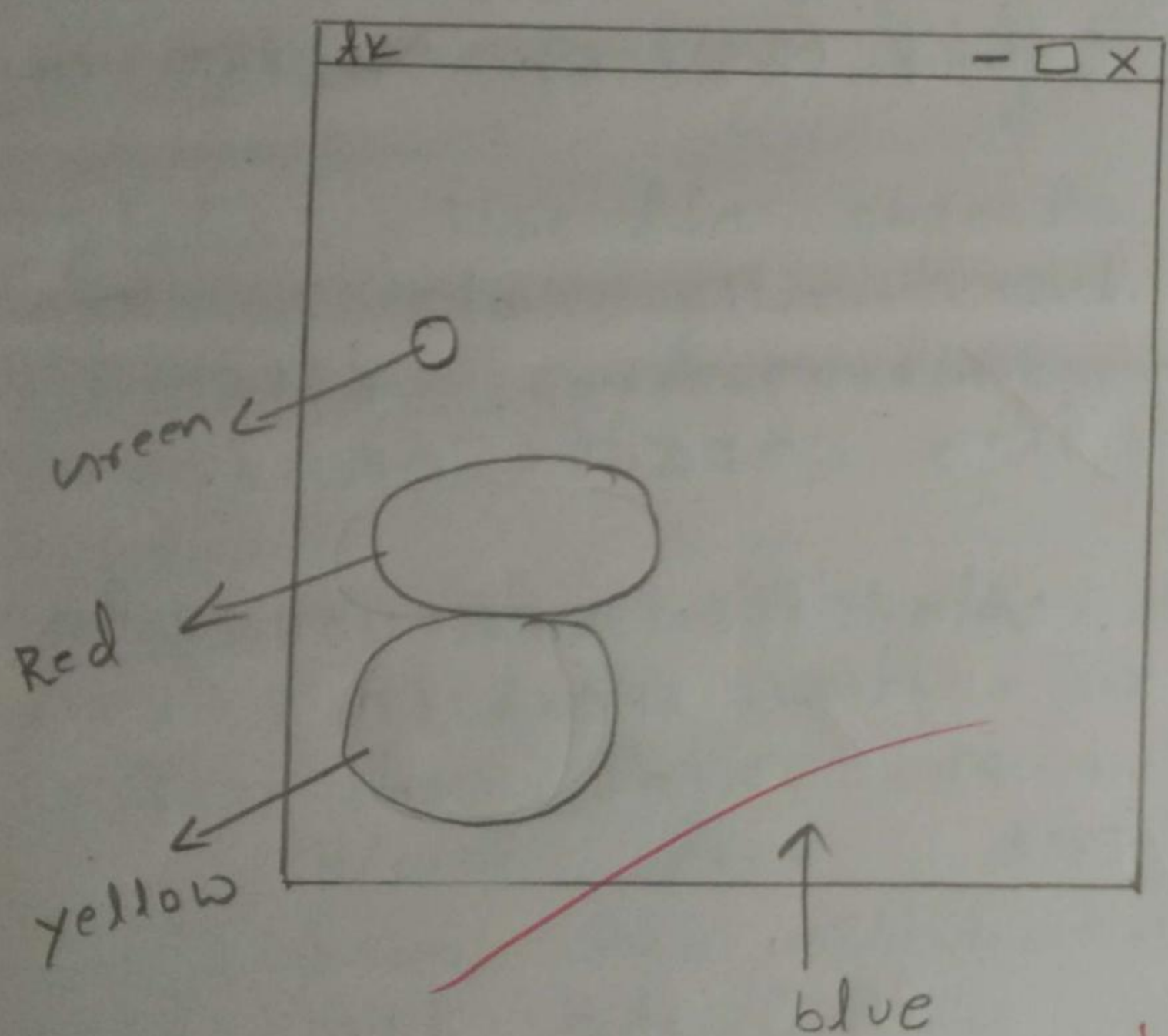
Step2: Use the method create oval, create line and create arc along with the canvas object so created and use the coordinate value. Also use the fill attribute to assign various colours.

Step3: Now call the pack method and mainloop method.

Scanned by TapScanner

```
from tkinter import*
root = TK()
C1 = Canvas(root, height = 400, width = 400, bg = "blue")
oval = C1.create_oval(20, 140, 150, 250, fill = "yellow")
line = C1.create_oval(30, 40, 50, 60, fill = "green")
arc = C1.create_oval(20, 140, 150, 60, fill = "red")
C1.pack()
root.mainloop()
```

Output:



Jan 17/12

Practical-6

(a) Aim: Demonstrate the Use of connectivity

Step1: import db library and use db open method for creating the data base by specify name of the data base along with the corresponding flag.

Step2: Use the objects for accessing to given web size and the corresponding regular for the web size.

Step3: check whether the given URL address with the regular of the Pages is not equal to None then display the message from the URL address else NOT FOUND.

Scanned by TapScanner

```
import dbm
db = dbm.open("database", flag="c")
if db["WWW"] != None:
    Print("yess")
else:
    Print("NO")
```

<u>output</u>:

yes

Scanned by TapScanner

(b) Creating the table and inserting Value

Step1: import the corresponding Library taking of data base connection.

Step2: Now create connection objects using sqlite3 Library and connecting method for create the new database.

Step3: Now create the cursor objects using cursor method forom the connection objects create in steps.

Step4: Now use the executing method for creating the table with the column name and respective data type.

Step5: Now with the cursor objects use insert statements for entering the values co-ordinating into the different field considering the data type.

```
import os, 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')
c1.fetchall()
c1.execute("Drop table student")
```

<u>output:</u>

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

Scanned by TapScanner

step6: Use the commit Method to complete the transection use the connection objects.

step7: Use the execute statement along with the ~~value for~~ ~~the table~~ ~~using~~ Cursor objects for ~~accse~~ ~~acss~~ ~~acess~~ accessing the value the data base using selecting from wherecclause.

step8: finally use the ~~fetatt~~ fetchall method for display the value for the table using the cursor objects.

<u>step9</u>: Use the execute Method and the drop table for terminating the database finally use the close Method.

Project

Aim: To create a Travel Management using

```
from tkinter import*

from tkinter import ttk

import random

import time;

import datetime

import tkinter.messagebox

class Travel:

        def __init__(self,root):

                self.root=root

                self.root.title("Travel Management Systems")

                self.root.geometry("1350x750+0+0")

                self.root.configure(background='black')


                DateofOrder=StringVar()

                DateofOrder.set(time.strftime("%d/%m/%y"))

                Receipt_Ref=StringVar()

                PaidTax=StringVar()

                SubTotal=StringVar()

                TotalCost=StringVar()


                Var1=IntVar()

                Var2=IntVar()

                Var1=IntVar()

                Var3=IntVar()

                Var4=IntVar()
```

Scanned by TapScanner

```
Var5=IntVar()

Var6=IntVar()

Var7=IntVar()

Var8=IntVar()

Var9=IntVar()

Var10=IntVar()

Var11=StringVar()

Var12=StringVar()

Var13=StringVar()


Firstname=StringVar()

Surname=StringVar()

PostCode=StringVar()

Address=StringVar()

Telephone=StringVar()

Mobile=StringVar()

Email=StringVar()



AirportTax=StringVar()

Mile=StringVar()

Travel_Ins=StringVar()

Luggage=StringVar()


Standard=StringVar()
```

Scanned by TapScanner

```
Economy=StringVar()

FirstClass=StringVar()


AirportTax.set("0")

Mile.set("0")

Travel_Ins.set("0")

Luggage.set("0")


Standard.set("0")

Economy.set("0")

FirstClass.set("0")


#==================================================================


MainFrame=Frame(self.root)

MainFrame.grid()


Tops=Frame(MainFrame,bd=20,width=1350,relief=RIDGE)

Tops.pack(side=TOP)


self.lblTitle=Label(Tops, font=('arial',70,'bold'),text="Travel Management Systems")

self.lblTitle.grid()


#==================================================================


CustomerDetailsFrame=Frame(MainFrame,width=1350,height=500,bd=5,pady=5,relief=RIDGE)
```

Scanned by TapScanner

```
CustomerDetailsFrame.pack(side=BOTTOM)

FrameDetails=Frame(CustomerDetailsFrame,width=880,height=400,bd=10,relief=RIDGE)
FrameDetails.pack(side=LEFT)

CustomerName=LabelFrame( FrameDetails,width=150,height=250,bd=5,
font=('arial',12,'bold'),text='Customer
Name',relief=RIDGE)
CustomerName.grid(row=0,column=0)

TravelFrame=LabelFrame( FrameDetails,width=300,height=250,bd=10,
font=('arial',12,'bold'),text='TravelDetails',relief=FLAT)
TravelFrame .grid(row=0,column=1)

Ticket_Frame=LabelFrame( FrameDetails,width=300,bd=10,height=200,
font=('arial',12,'bold'),text='Ticket_Frame',relief=RIDGE)
Ticket_Frame.grid(row=1,column=0)

Cost_Frame=LabelFrame( FrameDetails,width=150,bd=10,height=150,
font=('arial',12,'bold'),text='Cost_Frame',relief=FLAT)
```

Scanned by TapScanner

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


#==========================================================================

Receipt_ButtonFrame=Frame(CustomerDetailsFrame,width=450,bd=10,height=400,relief=RIDGE)

Receipt_ButtonFrame.pack(side=RIGHT)


ReceiptFrame=LabelFrame(Receipt_ButtonFrame,width=350,bd=20,height=300,
font=('arial',12,'bold'),text='Receipt',relief=RIDGE)

ReceiptFrame.grid(row=0,column=0)


ButtonFrame=LabelFrame(Receipt_ButtonFrame,width=350,bd=20,height=300,
font=('arial',12,'bold'),text='Buttons',relief=RIDGE)

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



#==========================CustomerName==================================


self.lblFirstname=Label(CustomerName,
font=('arial',14,'bold'),text="Firstname",bd=7)

self.lblFirstname.grid(row=0,column=0,sticky=W)


self.txtFirstname=Entry(CustomerName,
font=('arial',14,'bold'),text="Firstname",textvariable=Firstname,bd=7,
```

Scanned by TapScanner

```
                                                insertwidth=2,justify=RIGHT)

                    self.txtFirstname.grid(row=0,column=1)


                    self.lblSurname=Label(CustomerName, font=('arial',14,'bold'),text="Surname",bd=7)

                    self.lblSurname.grid(row=1,column=0,sticky=W)



                    self.txtSurname=Entry(CustomerName,
font=('arial',14,'bold'),text="Surname",textvariable=Surname,bd=7,

                                                insertwidth=2,justify=RIGHT)

                    self.txtSurname.grid(row=1,column=1)


                    self.lblAddress=Label(CustomerName, font=('arial',14,'bold'),text="Address",bd=7)

                    self.lblAddress.grid(row=2,column=0,sticky=W)



                    self.txtAddress=Entry(CustomerName,
font=('arial',14,'bold'),text="Address",textvariable=Address,bd=7,

                                                insertwidth=2,justify=RIGHT)

                    self.txtAddress.grid(row=2,column=1)


                    self.lblPostCode=Label(CustomerName, font=('arial',14,'bold'),text="PostCode",bd=7)

                    self.lblPostCode.grid(row=3,column=0,sticky=W)
```

Scanned by TapScanner

```
self.txtPostCode=Entry(CustomerName,
font=('arial',14,'bold'),text="PostCode",textvariable=PostCode,bd=7,
insertwidth=2,justify=RIGHT)

self.txtPostCode.grid(row=3,column=1)


self.lblTelephone=Label(CustomerName,
font=('arial',14,'bold'),text="Telephone",bd=7)

self.lblTelephone.grid(row=4,column=0,sticky=W)



self.txtTelephone=Entry(CustomerName,
font=('arial',14,'bold'),text="Telephone",textvariable=Telephone,bd=7,
insertwidth=2,justify=RIGHT)

self.txtTelephone.grid(row=4,column=1)



self.lblEmail=Label(CustomerName, font=('arial',14,'bold'),text="Email",bd=7)

self.lblEmail.grid(row=5,column=0,sticky=W)


self.txtEmail=Entry(CustomerName,
font=('arial',14,'bold'),text="Email",textvariable=Email,bd=7,
insertwidth=2,justify=RIGHT)

self.txtEmail.grid(row=5,column=1)
```

Scanned by TapScanner

```
self.lblMobile=Label(CustomerName, font=('arial',14,'bold'),text="Mobile",bd=7)

self.lblMobile.grid(row=6,column=0,sticky=W)


self.txtMobile=Entry(CustomerName, font=('arial',14,'bold'),text="Mobile",bd=7,
                                    insertwidth=2,justify=RIGHT)

self.txtMobile.grid(row=6,column=1)



#=====================Flight
Information=================================

self.lblDeparture=Label(TravelFrame,font=('arial',14,'bold'),text="Departure",bd=7)

self.lblDeparture.grid(row=0,column=0,sticky=W)



self.cboDeparture=ttk.Combobox(TravelFrame,textvariable=Var11,state='readonly',font=('arial',20,'bold'
),
                                    width=14)


self.cboDeparture['value']=(' ','Healthrow','Lagos MM','Dew','Lutton')

self.cboDeparture.current(0)

self.cboDeparture.grid(row=0,column=1)
```

==========

```
if __name__ =='__main__':
        root=Tk()
        application=Travel(root)
        root.mainloop()
```

Scanned by TapScanner

Travel Management System

Travel Management Systems

Customer Name

Firstname	Puja
Surname	Saw
Address	Borivali
Postcode	9678
Telephone	9948673400
Email	abc@gmail.com
Mobile	7853237800

Travel Details

Departure: Delw

Receipt

Buttons

Ticket_Frame

Cost_Frame

Scanned by TapScanner