☆☆ INDEX ☆☆

No.	Title	Page No.	Date	Staff Member's Signature
1.	Demonstrate the use of different file accessing modes	22	28/11/19 5/12/19	5/12/19
2.	Iteration	26	9/12/19	19/12/19
3.	Program to demonstrate Exception Handling	28	19/12/19	16/1
4.	Demonstrate the use of regular expression	31	8/1/20	16/1
5.	A) GUI components	35	16/1/20	23/1
	B) " GUI components "	37	16/1/20	
	C) GUI components (multiple window)	40	23/1/20	
	D) GUI (Displaying the image)	43	6/2/20	
	E) GUI (paned windows & canvas widget)	45	13/2/20	
6.	Database Connectivity	47	20/2/20	27/2

```
fileobj = open ("sample.txt", "w")
fileobj.write ("Computer Science subjects" + "\n")
fileobj.write ("DBMS \n PYTHON \n FOSS \n")
fileobj.close()
# read()
fileobj = open ("sample.txt", "r")
str1 = fileobj.read()
print ("The output of read method :", str1)
fileobj.close()
>>> ('The output of read method :', 'Computer Science Subjects
    \n DBMS \n PYTHON \n FOSS \n')

# readline()
fileobj = open ("sample.txt", "r")
str2 = fileobj.readline()
print ("The output of readline method :", str2)
>>> ('The output of readline method :', 'Computer Science Subjects')

# readlines()
fileobj = open ("sample.txt","r")
str3 = fileobj.readlines()
print ("The output of readlines method :", str3)
>>> ('The output of readlines method :', ['Computer Science Subjects\n',
'DBMS\n', 'PYTHON\n', 'FOSS\n'])

# file attributes()
a = fileobj.name
print ("Name of file (name attribute) :", a)
>>> ('name of file (name attributes)', 'sample.txt')
b = fileobj.closed
print ("(close) attribute :", b)
                    >>> ('(close) attribute :', 'True')
```

PRACTICAL – 1

Objective :- Demonstrate the use of different file accessing modes different attributes, read methods.

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

Step 2 :- Now open the file in read mode & than use read() readline() & readlines() & store the output in variable & finally display the contents of variable.

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

Step 4 :- Now open the fileobj in write methode write some another content close subsequently then again open the file obj in 'w+' mode that is the update mode & write contents.

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

Step 6 :- Now open fileobj in append mode open write methood write contents close the fileobj again open the fileobj in read mode & display the 'appending' output.

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

w+ mode

```
fileobj = open ("sample.txt ", "w+")
fileobj.write (" Loukik Sir ")
fileobj.close ()
```

r+ mode

```
fileobj = open ("sample.txt ", "r+")
g1 = fileobj.read (6)
print ("Output of r+ ", g1)
fileobj.close ()
>>> ('Output of r+','Loukik')
```

append mode

```
fileobj = open ("sample.txt", "a")
fileobj.write ("Data Analysis")
fileobj.close ()
fileobj = open ("sample.txt ", "r")
g3 = fileobj.read ()
print ("Output of append mode : ", g3)
fileobj.close ()
>>> ('Output of append mode : ', ' Loukik Sir
    Data Analysis')
```

write mode

```
fileobj = open ("sample.txt ", "w")
fileobj.write ("DBMS")
fileobj.close ()
```

21

read mode

```
fileobj = open ("sample.txt ", "r")
g = fileobj.read ()
print ("Output of read mode!", g)
>>> ('Output of read mode!',
    'Loukik Sir.')
```

tell()

```
fileobj = open("sample.txt","r")
pos = fileobj.tell()
print("tell() :",pos)
fileobj.close
>>> ('tell() : ',0L)
```

seek()

```
fileobj = open("sample.txt","r")
st = fileobj.seek(0,0)
print("seek(0,0) is :", st)
fileobj.close()
>>> ('seek(0,0) is : ', none)
fileobj = open("sample.txt", "r")
st1 = fileobj.seek(0,1)
print("seek(0,1) is :", st1)
fileobj.close()
>>>('seek(0,1) is : ', none)
fileobj = open("sample.txt", "r")
st2 = fileobj.seek(0,2)
print("seek(0,2) is :", st2)
fileobj.close()
>>> ('seek(0,2) is : ', none)
```

finding length of different lines exist within lines

```
fileobj = open("sample.txt", "r")
stat = fileobj.readlines()
print("Output = ", stat)
for line in stat:
    print(len(line))
fileobj.close()
>>>('Output : ',['Loukik Siv, Data Analysis'])
```

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

Step 8 :- Use the seek method with the arguments with opening the file obj in read mode & closing subsequently

Step 9 :- Open fileobj with read mode also use the readlines method & store the output consequently in & print the same for counting the length use the for conditional statement & display the length.

PRACTICAL - 2

Aim :- Iteration

Programs :-

I] Odd Numbers (Algorithm) :-

Step 1 :- Start

Step 2 :- Define iter method with an arguement & initialize its value as 1. Return the value.

Step 3 :- Increment the value by 2. Return the value.

Step 4 :- Create an object & pass it through iter method use while conditional statement to print.

Step 5 :- Stop.

II] Power (Algorithm) :-

Step 1 :- Start.

Step 2 :- Define iter method with 3 arguements. Initialize one arguement the value 1. Initialize another arguement the number whose power is to be searched. Initialise another arguement the maximum limit of power.

Step 3 :- Define the next method & use while loop to

Source Code (odd):-

```
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

y = iter(odd())
y.next()
```

Output:-

1
3
5
7
9
11
13
15
17
19

Source code (power):-

```
class myiter:
    def __iter__(pow):
        pow.n = 1
        pow.ni = int(input("Enter a number: "))
        pow.nii = int(input("Enter the power's limit"))
        return pow
    def __next__(pow):
        if pow.n <= pow.nii:
            num = pow.ni ** pow.n
            pow.n += 1
            return num
        else:
            raise StopIteration
y = iter(myiter())
while True:
    print(next(y))
```

Output:-

Enter a number: 2
Enter the power's limit: 4

2
4
8
16

Source Code (Range 1-40):

```
class my_range():
    def __iter__(self):
        self.a = 1
        return self
    def __next__(self):
        if self.a <= 40
            x = self.a
            self.a += 1
            return x
            else:
                raise StopIteration

myclass = my_range()
mydata = iter(myclass)
for x in mydata:
    print(x)
```

Output :-

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
```

use Step 4:- If the initialized value or input variable is bigger than 6 then raise stop iteration or else returns the multiplication of agreement by itself.

Step 5:- Create a object of given class & pass the iter method & use while loop to use next method.

Step 6:- Stop

III) Range (Algorithm) :-

Step 1:- Start.

Step 2:- Define an iter method with a argument & initialise the value & return it.

Step 3:- Define the next method with a argument & compare the upper limit by using a conditional statement.

Step 4:- Now create an object of given class & pass this object in the iter method.

Step 5:- Print x using for loop with range as the object.

Step 6:- Stop

14/11/22

Practical - 3

Aim:- Program to demonstrate exception handling.

1) Write a Program using the exception method of the nature Arithmetic error.

Step 1:- Use the try block & except the input using the raw input method and convert it into the integer datatype and subsequently terminate the block.

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

2) Write a program for accepting the file in a given mode and use the environment error as an exception for the given input.

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

Step 2:- Use the except block with IO error and display the message regarding missing of the file or incompatibility of the mode use the else block to display a message that the operation is carried out successfully.

Program :-

```
while True
    try
        x = int (input ("Enter class"))
        break
    except Value Error
        print ("Enter Numeric Value")
```

Output :- Enter class 46

Enter class abc
Value Error
Enter Numeric Value.

Program :-

```
try :
    f = open ("sample.txt", "w")
    f.write ("Python")
except IOError
    print ("Error writing on the file")
else :
    print ("Operation carried out successfully")
    f.close ()
```

Output :-

Operation carried out successfully.

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 except with ValueError and print the message not a valid range.

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

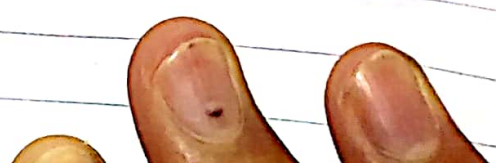
Step 1:- Define a function which accepts an arguements and check using the assert statement whether the given list is empty list & accordingly when print the message.

Step 2:- Create the function & in the body of the program and define certain elements in the list & 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 error exception otherwise return the valid age.

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

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



Program :-

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

Output :- list is empty.

Program :-

```
def acceptage():
    age = int(input("Enter age: "))
    if age >= 40 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: 5
Not a valid age
Enter age: 18
```

Code :-

```
import re
String = "hello 1234 abc4579"
result = re.findall("\d", string)
result1 = re.findall ("\", string)
print (result)
print (result1)
```

Output :-

```
>>> ['1234', '4579']
>>> ['hello', 'abc']
```

Practical - 4

Aim :- Demonstrate the use of regular expression.

Theory :- Regular expression represents the sequence of characters which is mainly used for finding & replacing the given patterns in a string & for this use import remodule & common usage of regular expression involves following functionalities.

* Searching a given string
* Finding a string
* Breaking a string into smaller substring
* Replacing part of string

Q.1. Write a regular expression segregating numeric & alphabetic values from a given string.

Algorithm :-

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

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

Q.2. Write a regular expression for finding the match string at the beginning of given sequence.

Algorithm :-

Step 1 :- Import, remodule & apply a string.

Step 2 :- Use search() with "\A Python" & string as three parameters.

Step 3 :- Now display the output.

Step 4 :- Now use if conditional statement for user the known whether the match is found or not.

Q.3. Write a regular expression to check whether the given mobile number starts with 8 or 9 & the total length 10.

Algorithm-

Step 1 :- Import re module & ally a string of mobile no. &.

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

code 2

```
import re
string = "Python is a important language"
result = re.match ("(A Python", string)
print (result)
if result:
    print ("In Match found")
else:
    Print ("In Match not found").
```

Output :-

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

code 3

```
import re
li = ["9876394512", "8755932913", "7977899055", "6543211987"]
for element in li:
    result = re.match ("[8-9]-[1?] [0-9]{9}", element)
if result:
    print ("correct mobile no")
    Print (result.group (1))
else:
    print ("Incorrect mobile no")
```

Output :-

```
>>> correct mobile no : 9876394512.
    Correct mobile no : 8755932913
    Incorrect mobile no : 7977899055
    Incorrect mobile no : 6543211987
```

code 4

```
import re
string = "Python is important"
result1 = re.findall ("\w*.", string)
result2 = re.findall ("\w+", string).
print (result 1)
print (result 2)
```

Output :-

```
>>> ['Python', 'is', 'important']
['Python', 'is', 'important']
```

code 5

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

Output :-

```
>>> ['Python']
>>> ['important']
```

P.4. Write a regular expression for extracting a word from given string along with space characters in b/w the word & subsequently extract its words without space characters.

Algorithm :-

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

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

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

Step 4 :- Now display the output.

P.5. Write a regular expression for extracting first & last word from a string.

Algorithm :-

Step 1 :- Import remodule & apply a string.

Step 2 :- Use findall() is which use '^\w+' as one parameter find first word of string then. are "\w+$" as parameter to find last word of string.

Step 3 :- Display the output.

Q.6. Write a regular expression for substituting the data in format dd-mm-yyyy, by using the findall() where the string has following format. AMIT, 201, 24-12-2019

Algorithm -

Step 1:- Import re module & apply string.

Step 2:- Use findall() method & use "\d{2}-\d{2}-\d{4}" as an parameter.

Step 3:- Now display the output.

Q.7. Write a re for extracting the (i) Username & email-id (ii) Hostname from email-id (iii) Both username & host-name from email-id.

Algorithm :-

Step 1:- Import re module & apply a string

Step 2:- Use findall() as find username, hostname & user as email-id.

Step 3:- Use "\w+" for username use "\w+@" for hostname & use "[\w\.-]+" for both as parameter in findall()

Step 4 - Display the output.

code 6

```
import re
string = "Amit 201 24-12-2019"
result = re.findall ("\d{2}-\d{2}-\d{4}", string)
print (result)
```

output :-

```
>>> ['24-12-2019']
```

code 7

```
import re
string = "abc @ tcsc.edu"
result 1 = re.findall ("\w+", string")
result 2 = re.findall ("+\w+.\w+$", string")
result 3 = re.findall ("[\w/.-] + ", string)
print (result 1)
print (result 2)
print (result 3)
```

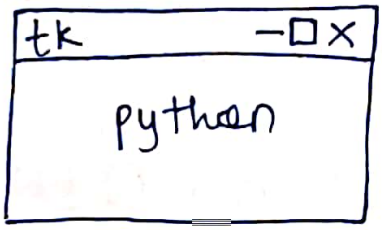
Output :-

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

creation of parent window

```
& from Tkinter import *
  root = Tk()
  l = Label (root, text = "python")
  l.pack()
  root.mainloop()
```

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", fg = "blac
           font = "20")
l2.pack (side = LEFT, pady = 30)
l3 = Label (root, text = "CS1", bg = "yellow", fg = "blac
     font = "10")
l3.pack (side = TOP, ipadx = 40)
```

Practical - 5 (A)

Topic :- GUI components.

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

Step 2 :- Create an object using the TK()

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

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

#2 :-

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

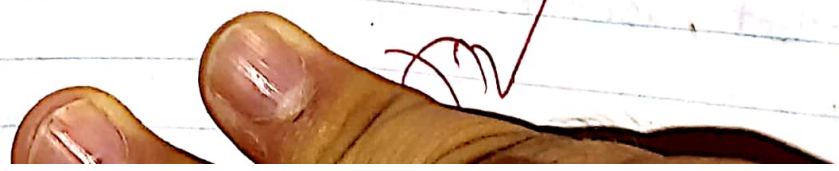
Step 2 :- Create a variable from the text method & position it on the parent window.

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

1) side = LEFT, padx = 20
2) side = LEFT, pady = 30
3) side = TOP, ipadx = 40
4) side = TOP, ipady = 50

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

Step 5:- Now repeat above steps with the label() which takes the following arguements.
i) Name of the parent window.
ii) Text attribute which defines the string.
iii) The background color (bg)
iv) The foreground fg & than use the pack() within a relevant padding attributes.



Radio button

```
from Tkinter import *
root = Tk()
root.geometry("500 x 500")
def select():
    Selection = "You just selected" + Str(var.get())
    t1 = Label(Text = Selection, bg = "white", fg = "grey")
    t1.pack(side = TOP)
var = StringVar()
l1 = Listbox()
l1.insert(1, "List 1")
l2.insert(2, "List 2")
l1.pack(anchor = N)
r1 = RadioButton(root, text = "option 1", variable = var,
  value = "option 1", command = select)
r1.pack(anchor = N)
r2 = RadioButton(root, text = "option 2", variable =
    var, value = "option 2", command = select)
r2.pack(anchor = N)
root.mainloop()
```

Practical - 5(B)

TOPIC :- GUI components.

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

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

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

Step 4 :- Now define the parent window and define the option with control variable.

Step 5 :- Use the list box() & insert options on the parent window along with the pack() with specifying anchor attribute.

Step 6 :- Create an object from radio button which will take following arguements & parent window object, text variable which will take the values option no 1, 2, 3, ... variable arguement, corresponding value.

Step 7 :- Trigger the function declared.

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

#2 :-

Step 1 :- Import relevant methods from the tkinter library

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

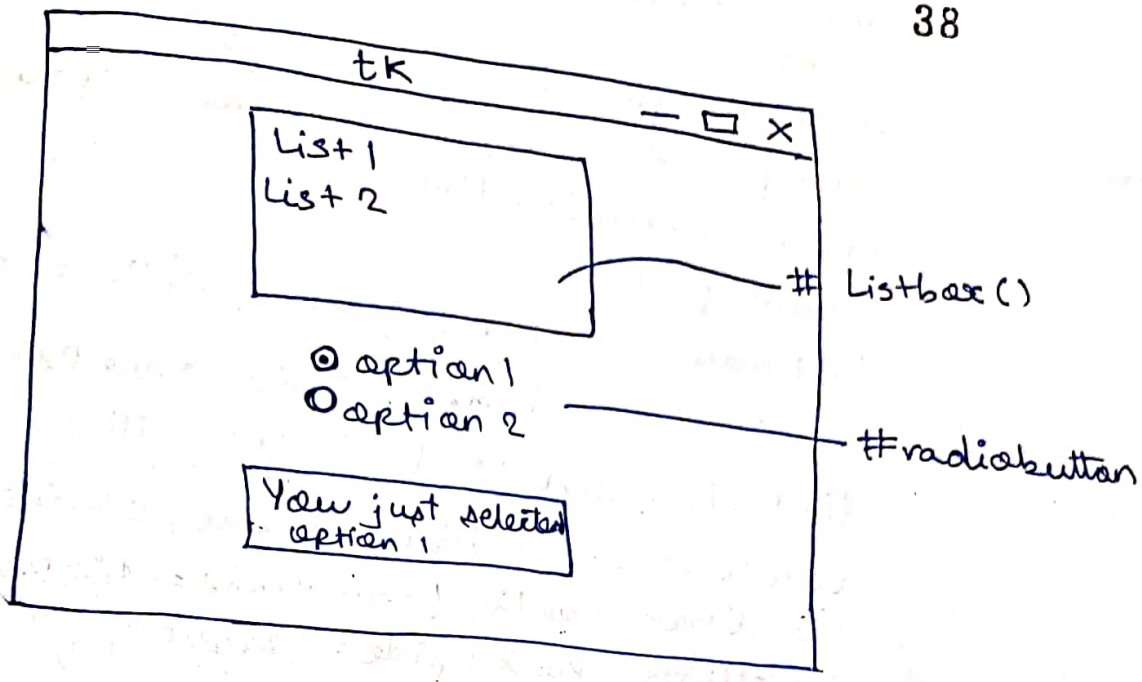
Step 3 :- Use the geometry () for laying of the window.

Step 4 :- Create an object & use the scrollbar ()

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

Step 6 :- Use the mainloop with the parent object.

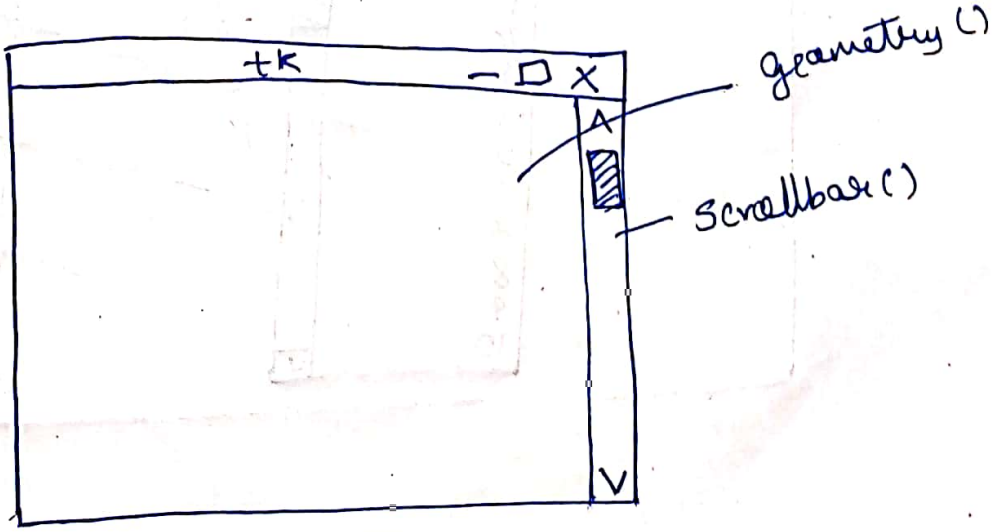
Output :-



2 :-

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

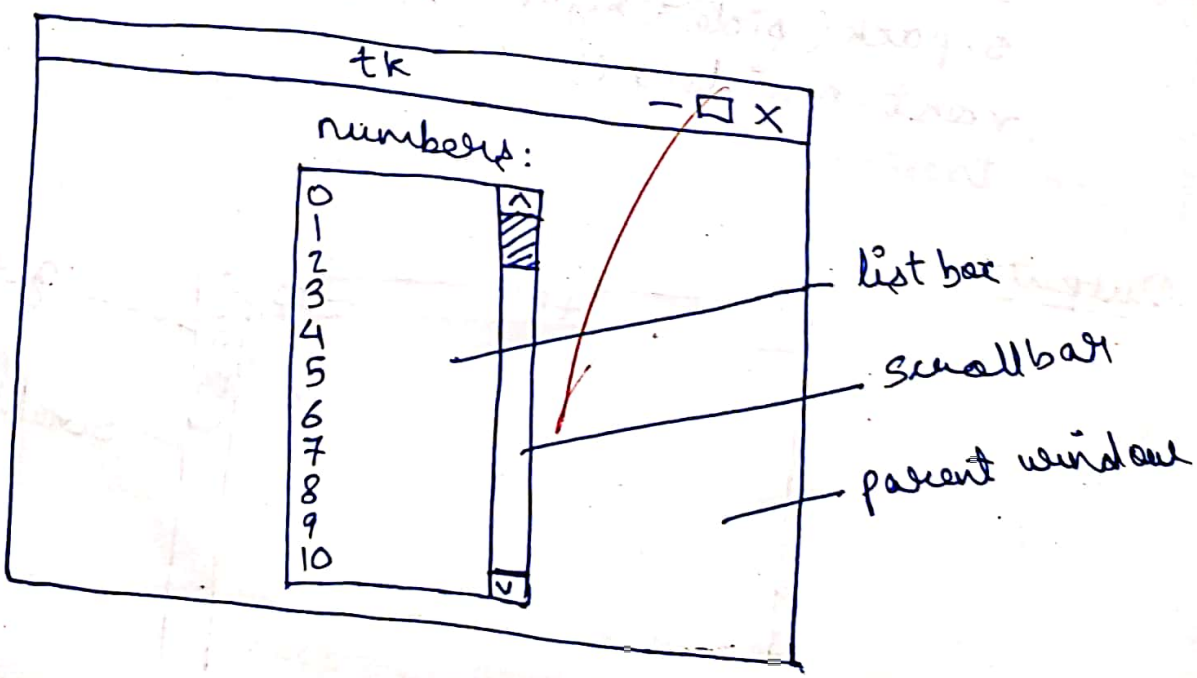
Output :-



#3 :-

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

Output :-



#3 Step 1:- Import the relevant libraries from the tkinter method.

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

Step 3:- Use the geometry manager with pixel size (680x500) 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 attributes created and use the pack method

Step 6:- Use the listbox method along with the attributes like width, height font. Do create a listbox methods object use pack() for the same.

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

Step 8:- Trigger the events using mainloop.

PRACTICAL - 5(C)

Aim :- GUI components

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

Step 2 :- Import tkMessagebox

Step 3 :- Define a parent window object along with the parent window.

Step 4 :- Define a function which will use tkMessagebox with showinfo method along with info window attribute.

Step 5 :- Declare a button with parent window object along with the command attribute.

Step 6 :- Place the button widget onto the parent window & finally call mainloop() for triggering of the events called above.

```
# Message Box
from tkinter import *
import tkMessageBox
root = Tk()
def func:
    tkMessageBox.showinfo("info window","Python 2.7")
    b1 = Button(root, text = "Python version info.", command
                = func)

    b1.pack()
    root.mainloop()
```

#= Multiple Windows
#= Different button (Relief())

```
from tkinter import *
root = Tk()
root.minsize(300,300)
def main():
    top = Tk()
    top.config(bg = "black")
    top.title("HOME")
    top.minsize(300,300)
    L= Label (top, text = "SAN FRANCISCO In Places of Interest
        In Golden Gate Bridge In Lombard Street In Chinatown
        In Coit Tower")
    L.pack()
    b1 = Button(top, text = "next", command = second)
    b1.pack(side = RIGHT)
    b2 = Button(top, text = "exit", command = terminate)
    b2.pack(side = LEFT)
    top.mainloop()
```

Step 1:- Import the relevant methods from the tkinter library along with parent windows object declared.

Step 2:- Use parentwindow object along with minsize function for windows size.

Step 3:- Define a function main() ; declare parent windows object & use config(), title(), minsize(), Label() as well as Button() & use pack() & mainloop() simultaneously.

Step 4:- Similarly define the function second & use the attributes accordingly.

Step 5:- Declare another function button along with parent object & declare button with attributes like FLAT, RIDGE, GROOVE, RAISED, SUNKEN along with the relief widget.

Step 6:- Finally call the mainloop() for event driven programming.

```
bB = Button (top3, text = "ridge button", relief = RIDGE)
top3.mainloop()
def terminate():
    quit()
bB = Button (root, text = "TOUR DETAILS", command = main)
bB.pack()
b6 = Button (root, text = "BUTTON DETAILS", command =
                                                button)
b6.pack()
root.mainloop()
```

```
def second():
    top2 = Tk()
    top2.config(bg = "orange")
    top2.title("About US !")
    top2.minsize(300,300)
    L=Label(top2, text = "Created by : Shubham Uniyal \n
    For more details contact the our official
    account")
    L.pack()
    b3 = Button(top2, text = "prev", command = main)
    b3.pack(side=LEFT)
    b2 = Button(top2, text = "exit", command = terminate)
    b2.pack(side = RIGHT)
    top2.mainloop()

def button():
    top3 = Tk()
    top3.geometry("300x300")
    b1 = Button(top3, text = "flat button", relief = FLAT)
    b1.pack()
    b2 = Button(top3, text = "groove button", relief = GROOVE)
    b2.pack()
    b3 = Button(top3, text = "sunken button", relief = SUNKEN)
    b3.pack()
    b4 = Button(top3, text = "raised button", relief = RAISED)
    b4.pack()
```

```
from tkinter import *
root = Tk()
root.title("Python GUI")
root.maxsize(900,800)
root.config(bg="black")
leftframe = Frame(root, bg="yellow", height=
                  width="150")
leftframe.grid(row=0, column=0)
rightframe = Frame(root, bg="blue", height=
                   width="150")
rightframe.grid(row=0, column=0)
Label(leftframe, text="Photo", height=2, width=
      .grid(row=0, column=0)
image1 = PhotoImage(file="bear.gif")
image1.subsample(1,2)
image2 = PhotoImage(file="bear.gif")
image2.subsample(3,4)
Label(leftframe, image=image1).grid(row=0, colu
Label(rightframe, image=image2).grid(row=0, co
                  padx=20, pady=10)
                  padx=10, pady=10)
toolbar = Frame(leftframe, width=200, height=40
                bg="white").grid(row=2, colu
Label(toolbar, text="Personal Info.", height=2, wi
      relief=RAISED).grid(row=0, column=0, padx=20,
                                  pady=10)
def name():
    print("Name : Sruthi")
```

PRACTICAL - 5(D)

* Displaying the image

Algorithm :-

Step 1: Create an object corresponding to the parent window & use the following 3 methods. Title - Maxsize - Config.

Step 2: Create a leftframe object from the frame method & place it onto the parent window with the height, width and the bg specified. Subsequently use the grid method with the row, column, padx, pady specified.

Step 3: Now create a rightframe object from the frame method with the width, height specified and the rows & the column value should be specified.

Step 4: Create a label object from the label method & place it onto the leftframe with text attribute denoting the original image with relief attribute used as RAISED value and subsequently use grid method with row, column values specified as (0,0) with some external padding values.

Step 5 :- Now use the photo image method with the file attribute specified.

Step 6:- Use the sub sample method with the object of the image & give the x, y co-ordinate values.

Step 7:- Use the label method & position it onto the left frame and placing the image after the sampling and use the grid method for the positioning in the first row.

Step 8:- Create another label object & positioning it onto the rightframe. and specifying the image and background attribute with row and column attribute specify it as (0,0).

Step 9:- Now create an toolbar object from the frame method and position it onto the leftframe with the height & width specified and position it onto the second row.

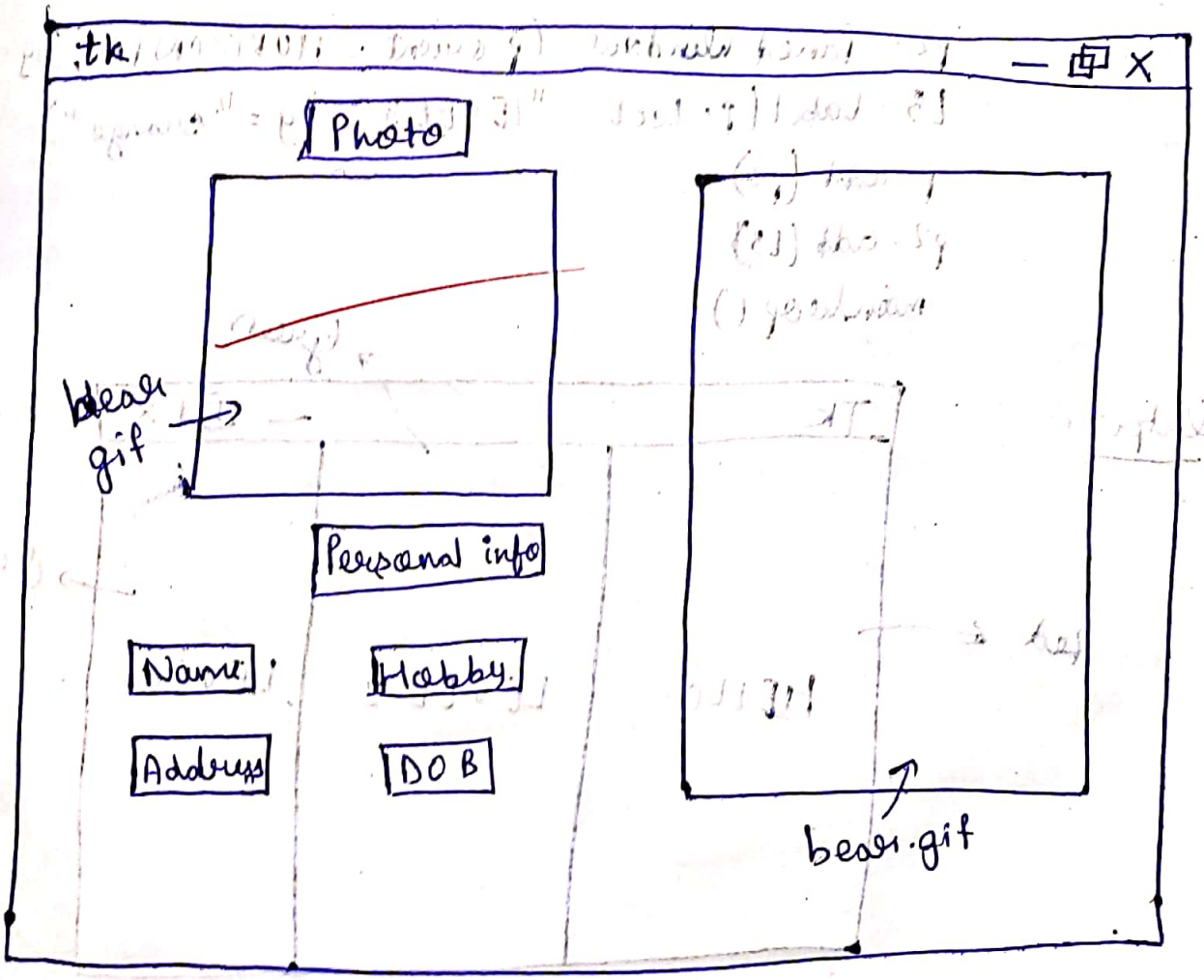
Step 10:- Now define the various function for different tool bar options provided in the leftframe.

Step 11:- From the label method position the text on the toolbar, use the relief attribute and corresponding grid value & incorporate the internal padding as well.

Step 12:- Create the label method position it on the toolbar with the next title as personal information and position it on same row but next column.

Step 13:- Now make use of mainloop method.

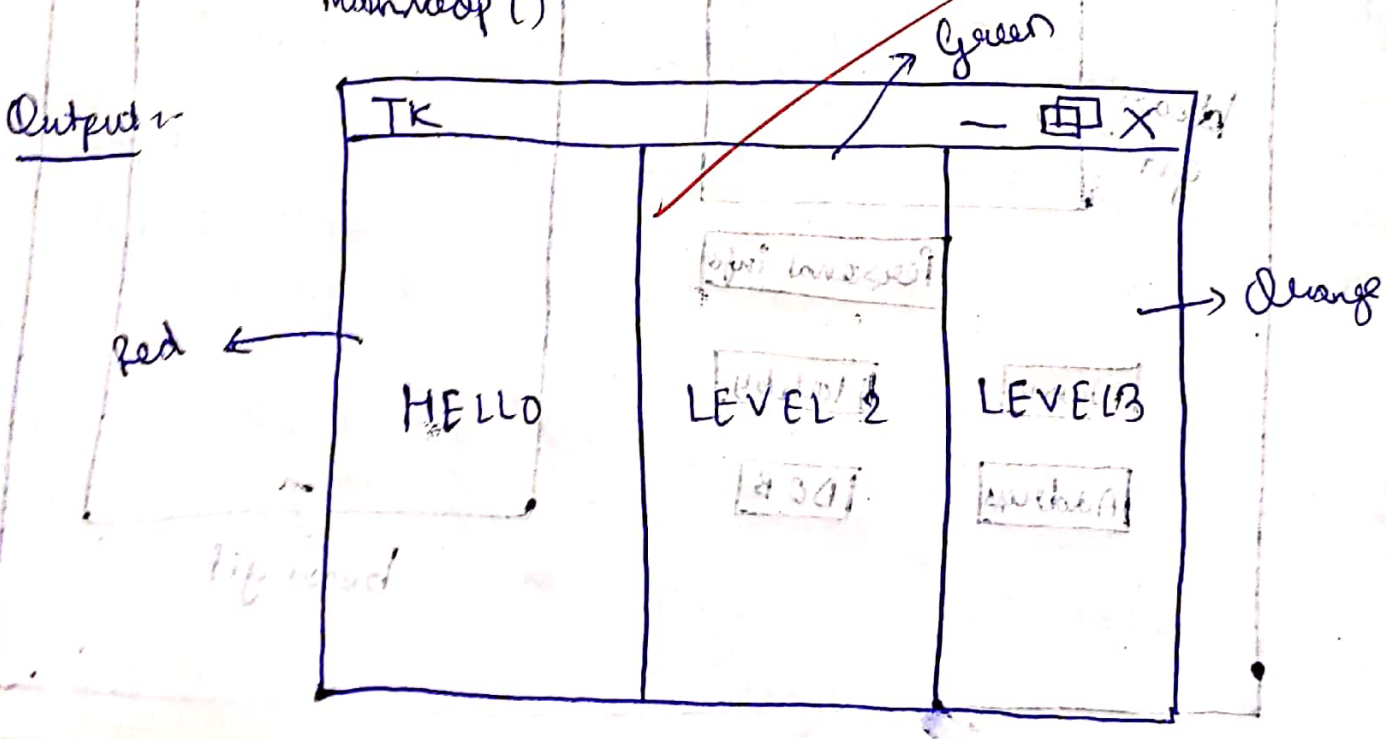
```
def hob():
    print ("Hobby : coding ")
def add():
    print ("Address : Mumbai ")
def dob():
    print ("DOB : 21/10/2001")
Button (toolbar, text = "Name", height = 1, width = 16,
        command = name).grid (row = 1, column = 0)
Button (toolbar, text = "Hobby", height = 1, width = 16,
        command = hob).grid (row = 1, column = 1)
Button (toolbar, text = "Address", height = 1, width = 16,
        command = add).grid (row = 2, column = 0)
Button (toolbar, text = "DOB", height = 1, width = 16,
        command = dob).grid (row = 2, column = 2)
root.mainloop()
```



Code :-

```
from tkinter import *
root = Tk()
p = PanedWindow(bg = "pink")
p.pack(fill = BOTH, expand = 10)
L1 = Label(p, text = "HELLO", bg = "RED")
p.add(L1)
~~L2 = Label(p, text = "")~~
p1 = PanedWindow(p, orient = VERTICAL, bg = "yellow")
p.add(p1)
L2 = Label(p1, text = "LEVEL2", bg = "green")
p1.add(L2)
p2 = PanedWindow(p, orient = HORIZONTAL, bg = "blue")
L3 = Label(p, text = "LEVEL3", bg = "orange")
p.add(p2)
p2.add(L3)
mainloop()
```

Output :-



PRACTICAL - 5(E)

* Paned Window :-

Step 1: Create an object from the paned window method and use the pack method to make this object visible.

Step 2: Now create an object from the entry widget and placed it onto the paned window and use the add method. Similarly, create an object of a panend window.

Step 3: Create an object from the scale widget & place it onto the preceding paned window and use the add method accordingly.

Step 4: Create a button widget and place it onto the paned window define a functionality along with the button widget.

Step 5: Use the pack method & mainloop method for the corresponding event to trigger.

* Canvas Widget

Step 1: Create an object from the canvas widget by using the attribute height width bg color & the parent window object.

Step 2: Use the corresponding method for drawing simple geometrical shape like arc, oval & line and specify the co-ordinate values.

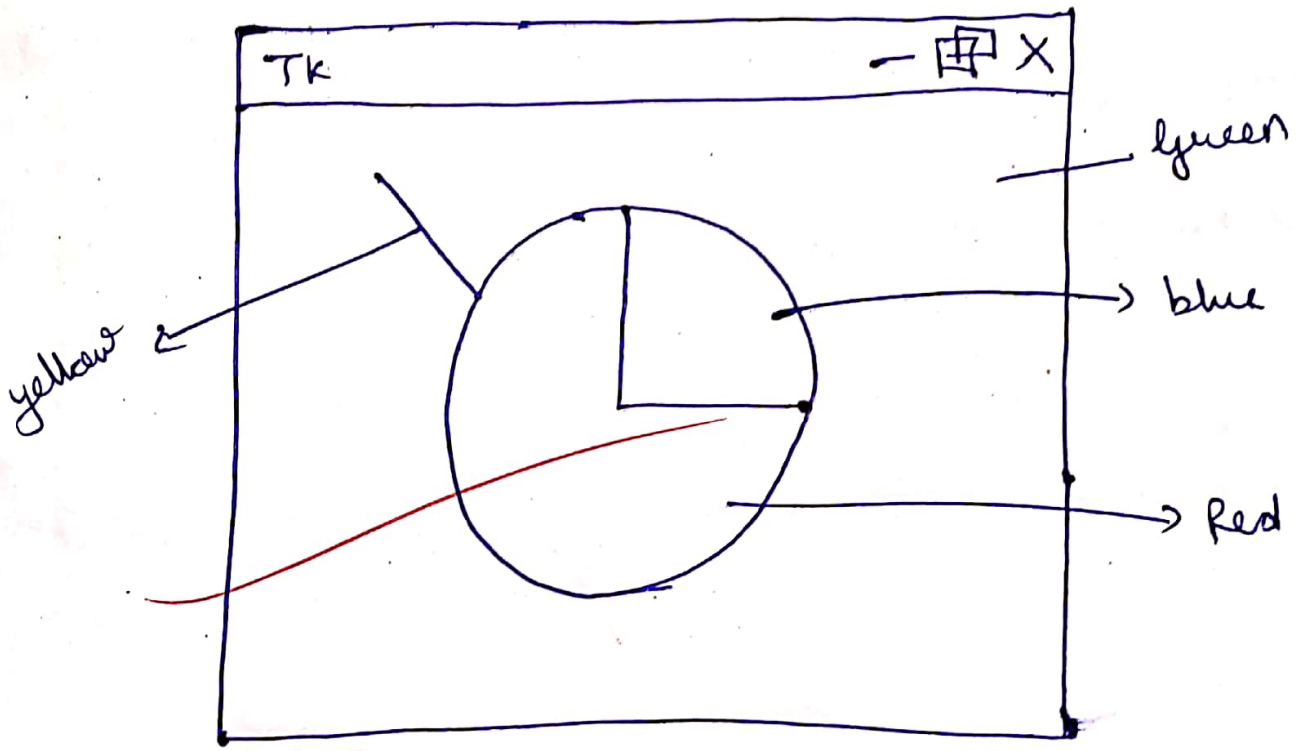
Step 3: Similarly use the create line & create oval meth along with the co-ordinate values & the fill attribut for specifying the colour.

Step 4: Finally use the pack & mainloop method.

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(300,18,18,400, fill = "blue")
c1.pack(side = TOP)
root.mainloop()
```

Output :-



I]

```
>>> anydbm
>>> db = dbm.open("database", flag='c', mode = 438)
>>> db["https://www.google.com"] = ("employee:db")
>>> if db["https://www.google.com"] != None:
        print ("database not empty")
    else:
        print ("database empty")
database not empty.
>>> db.close
```

PRACTICAL – 6

Aim :- Database connectivity.

Algorithm :-

Step 1 :- Import the 'dbm' library & use the open() for creating the database by specifying the name of the database along with the corresponding flag.

Step 2 : Use the object so created for accessing the given website & corresponding regular name for the website.

Step 3 : Check whether the given url address matches with the regular name of the page is equal to none then display the message found / match or else not found / mismatch.

Step 4 : Use the close() to terminate database library

II) Step 1: Import os & sqlite-3 library to make database connection.

Step 2: Now create the connection object using sqlite3 library and connect() for creating new database.

Step 3: Now create cursor object using the cursor() from the connection object created.

Step 4: Now use the execute() for creating the table with column name and representative datatype.

Step 5: Now with cursor object use the insert statement for entering the values corresponding to different fields corresponding the datatype.

Step 6: Use the commit() to complete the transaction using the connection object.

Step 7: Use the execute statement along with cursor objects for accessing the values from the database using the select from where clause.

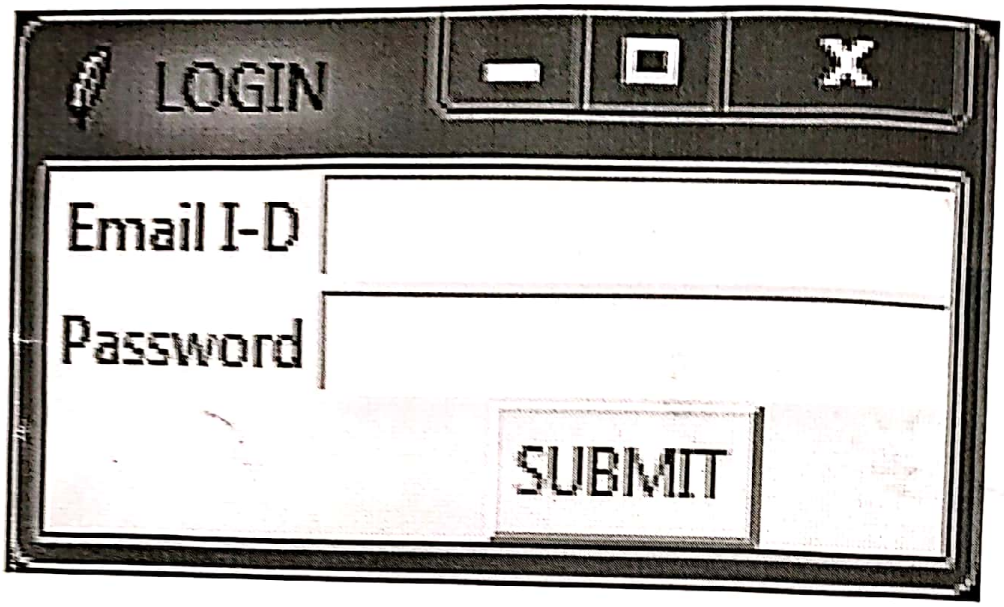
Step 8: Finally use the fetch() or fetchall() for displaying the value from the table using cursor-object.

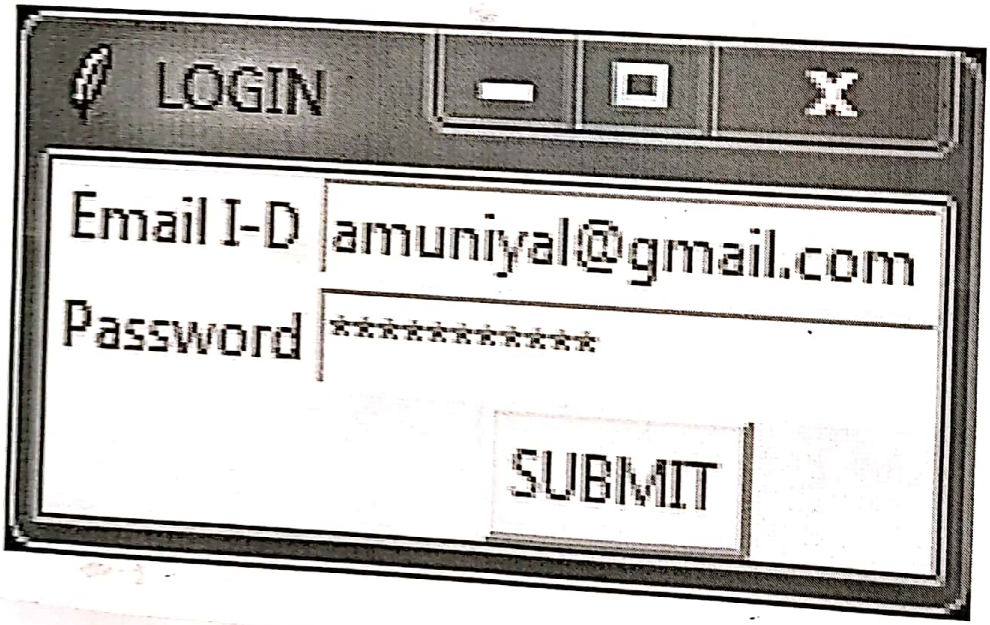
Step 9: Execute() and drop table syntax for terminating the database and finally use the close()

II] import os, sqlite3
connection = sqlite3.connect("student.db")
cur = connection.cursor()
cur.execute('Create table abc(Name varchar(20), Rno int(25,
Address varchar(20), Course varchar(30))')
cur.execute('insert into abc values("Shubham", 1786, "Mira Road",
"Science")')
cur.execute('insert into abc values("Azriel", 1799, "Malad", "Science")')
cur.execute('insert into abc values("Raclit", 1789, "Kandivali", "Arts")')
connection.commit()

Output :- >>> cur.execute('select Name from abc')
<sqlite3.Cursor object at 0x001C07A0>
>>> cur.fetchall()
[('Shubham',), ('Azriel',), ('Raclit',)]

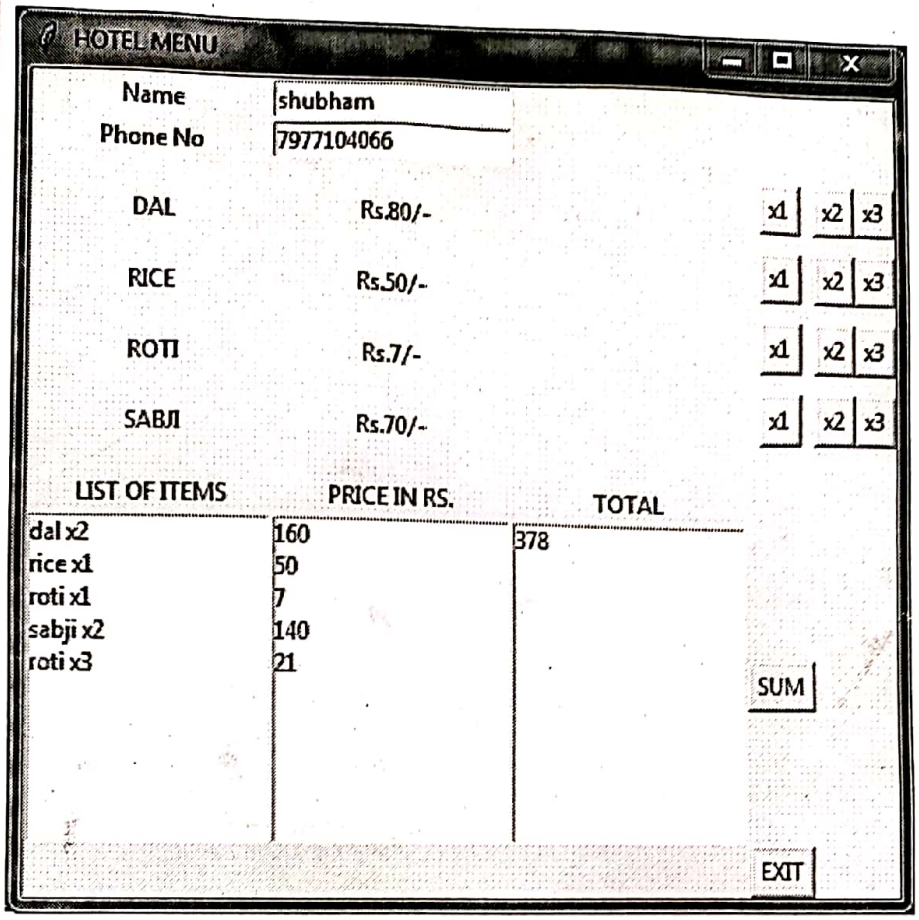
27/9

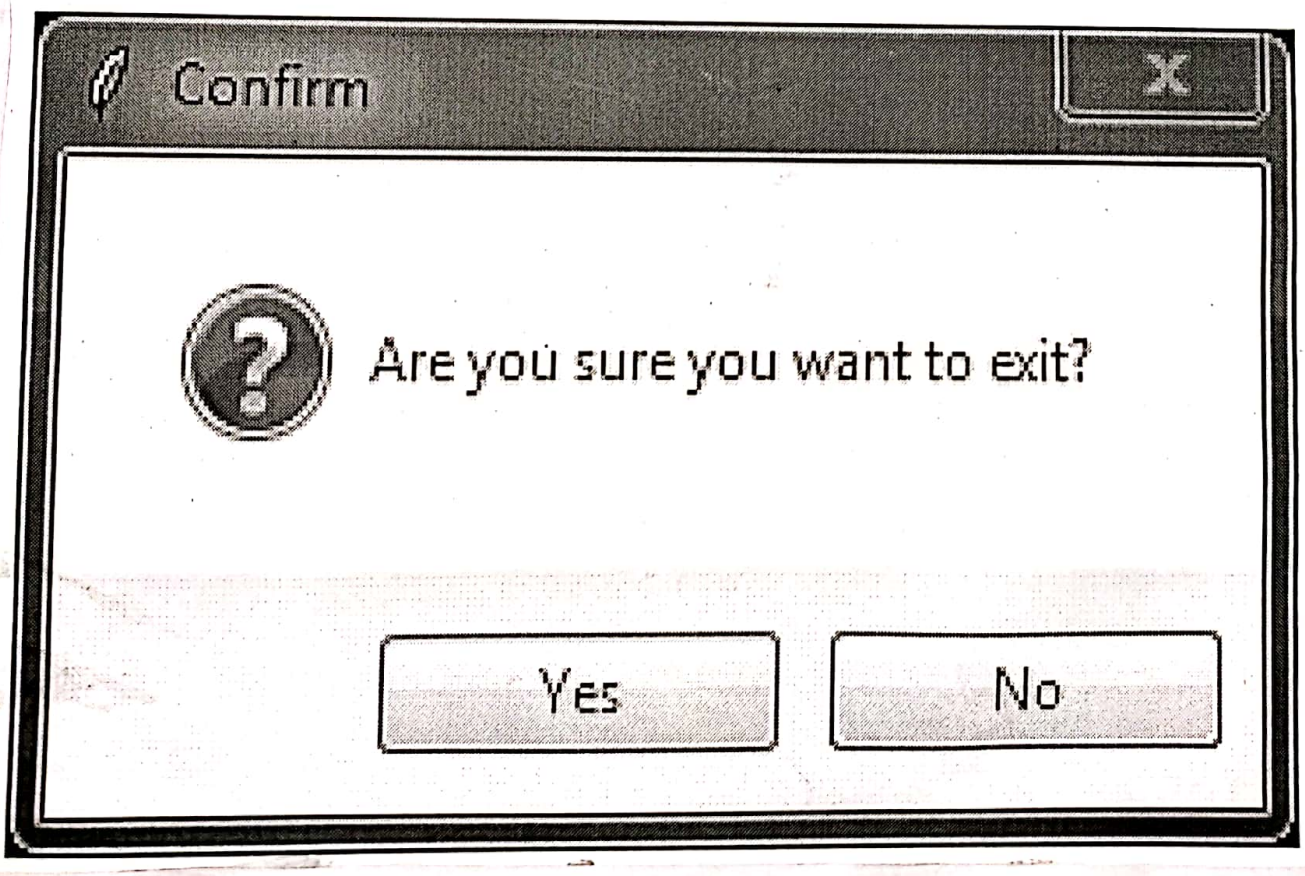




```
from tkinter import *                    project
from tkinter import messagebox
import math

b=Tk()
b.title("LOGIN")
l1=Label(b,text="Email I-D").grid(row=0,column=0)
e1=Entry(b).grid(row=0,column=1)
l2=Label(b,text="Password").grid(row=1,column=0)
e2=Entry(b,show="*").grid(row=1,column=1)
def submit():
      b.destroy()
Button(b,text="SUBMIT",command=submit).grid(row=2,column=1)
b.mainloop()

a=Tk()
a.title("HOTEL MENU")
l1=Label(a,text="Name").grid(row=0,column=0)
e1=Entry(a).grid(row=0,column=1)
l2=Label(a,text="Phone No").grid(row=1,column=0)
e2=Entry(a).grid(row=1,column=1)
t=Label(a,text="\n DAL")
t.grid(row=2,column=0)
t=Label(a,text="\n RICE ")
t.grid(row=3,column=0)
t=Label(a,text="\n ROTI")
t.grid(row=4,column=0)
t=Label(a,text="\n SABJI")
t.grid(row=5,column=0)
t=Label(a,text="\n Rs.80/-")
t.grid(row=2,column=1)
t=Label(a,text="\n Rs.50/- ")
t.grid(row=3,column=1)
t=Label(a,text="\n Rs.7/- ")
t.grid(row=4,column=1)
t=Label(a,text="\n Rs.70/- ")
t.grid(row=5,column=1)
t=Label(a,text="\n LIST OF ITEMS")
t.grid(row=6,column=0)
t=Label(a,text="\n PRICE IN RS. ")
t.grid(row=6,column=1)
t=Label(a,text="\n TOTAL ")
t.grid(row=6,column=2)


def dal1x():
      pr="dal x1"
      rs=80
      L1.insert(END,rs)
      L.insert(END,pr)
      list1.append(rs)
b1=Button(a,text="x1",command=dal1x).grid(row=2,column=3)
def dal2x():
      pr="dal x2"
      rs=160
      L1.insert(END,rs)
      L.insert(END,pr)
      list1.append(rs)
b2=Button(a,text="x2",command=dal2x).grid(row=2,column=4)

def dal3x():
      pr="dal x3"
      rs=240
      L1.insert(END,rs)
      L.insert(END,pr)
      list1.append(rs)
b3=Button(a,text="x3",command=dal3x).grid(row=2,column=5)
```

```
    def rice1x():
          pr="rice x1"
          rs=50
          L1.insert(END,rs)
          L.insert(END,pr)
          list1.append(rs)
    b4=Button(a,text="x1",command=rice1x).grid(row=3,column=3)

    def rice2x():
          pr="rice x2"
          rs=100
          L1.insert(END,rs)
          L.insert(END,pr)
          list1.append(rs)
    b5=Button(a,text="x2",command=rice2x).grid(row=3,column=4)

    def rice3x():
          pr="rice x3"
          rs=150
          L1.insert(END,rs)
          L.insert(END,pr)
          list1.append(rs)
    b6=Button(a,text="x3",command=rice3x).grid(row=3,column=5)

    def roti1x():
          pr="roti x1"
          rs=7
          L1.insert(END,rs)
          L.insert(END,pr)
          list1.append(rs)
    b7=Button(a,text="x1",command=roti1x).grid(row=4,column=3)

    def roti2x():
          pr="roti x2"
          rs=14
          L1.insert(END,rs)
          L.insert(END,pr)
          list1.append(rs)
    b8=Button(a,text="x2",command=roti2x).grid(row=4,column=4)

    def roti3x():
          pr="roti x3"
          rs=21
          L1.insert(END,rs)
          L.insert(END,pr)
          list1.append(rs)
    b9=Button(a,text="x3",command=roti3x).grid(row=4,column=5)

    def sabji1x():
          pr="sabji x1"
          rs=70
          L1.insert(END,rs)
          L.insert(END,pr)
          list1.append(rs)
    b10=Button(a,text="x1",command=sabji1x).grid(row=5,column=3)

    def sabji2x():
          pr="sabji x2"
          rs=140
          L1.insert(END,rs)
          L.insert(END,pr)
          list1.append(rs)
    b11=Button(a,text="x2",command=sabji2x).grid(row=5,column=4)

    def sabji3x():
          pr="sabji x3"
          rs=210
```





project

```
        L1.insert(END,rs)
        L.insert(END,pr)
        list1.append(rs)
  b12=Button(a,text="x3",command=sabji3x).grid(row=5,column=5)
  list1=[0,0,0,0,0,0,0,0,0,0,0,0,0]
  def sum1():
        sum1=sum(list1)
        L2.insert(END,sum1)
  b13=Button(a,text="SUM",command=sum1).grid(row=7,column=3)

  def bill():
       msg=messagebox.askquestion("Confirm","Are you sure you want to exit?")
       if msg == 'no':
        top=Tk()
        #Label(top,text="THANK YOU FOR VISITING SEE YOU
SOON!!").grid(row=0,column=0,rowspan=2,columnspan=3)
        #B=Button(top,text="GO BACK",command=t).grid(row=2,column=1)
        top.destroy()
        top.mainloop()
      else:
            a.destroy()
  b14=Button(a,text="EXIT",command=bill).grid(row=8,column=3)


 L=Listbox(a)
 L.grid(row=7,column=0)
 L1=Listbox(a)
 L1.grid(row=7,column=1)
 L2=Listbox(a)
 L2.grid(row=7,column=2)
 a.mainloop()
```