★★ INDEX ★★

No.	Title	Page No.	Date	Staff Member's Signature
1.	Demonstrate the code of different file accessing modes different attributes read method	22	28-11-19	
2.	Iterators	27	5-12-19	
3.	Program to demonstrate Exception Handling	31	19/12/19	
4.	Regular Expression	35	9/1/20	
5.	GUI components (A)	38	16/1/20	
(B)	GUI components (B) - message Box - Radio Button - Scroll bar - using Frame widgets	43	23/01/20	
•	GUI components (C) - Transposing - Displaying the image	48	6/2/20	

★★ INDEX ★★

No.	Title	Page No.	Date	Staff Member's Signature
•	GUI Components (1)	53	~~25~~/02/20	
	- Spinbox			
	- Panedwindow			
	- Canvas widgets			
6.	Data connectivity	56	20/02/20	
7.	Project (Spanish Learning course)	59		

Scanned with CamScanner

```
fileobj = open("abc.txt", "w") #file open(write mode)
fileobj.write("computer science subjects" + "\n")
fileobj.write("DBMS in PYTHON \n") # file write
fileobj.close() # file close

fileobj = open("abc.txt", "r") # read mode
# read()
st#1 = fileobj.read()
print(" the output of read method : ",st#1)
fileobj.close()
>>> ('the output of read method : ', 'computer science subjects \n DBMS \
Python \n')
# readline()
fileobj = open("abc.txt",'r')
st#2 = fileobj.readline()
print("the output of readline method : ",st#2)
fileobj.close()
>>> ('the output of readline method :', 'computer science subject \n')
# readlines()
fileobj = open("abc.txt", "r")
st#3 = fileobj.readlines()
print("the output of readlines method :",st#3)
fileobj.close()
>>> ('the output of read lines method :', ['computer science subjects \n
DBMS \n', 'PYTHON \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)
```


PRACTICAL NO. 1

OBJECTIVE : Demonstrate the use of different file accessing modes different attributes read methods.

Step 1 : Create a file objectfile using open methods 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 readlines() and store the output in variables and finally display the content 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 and finally the output of the softspace attribute

Scanned with CamScanner

Step 4 : Now open the fileobj in write mode write some another content close subsequently then again open the fileobj in ('w+') mode that is the update mode and write contents.

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

Step 36: Now open fileobj in append mode open write method write contents close the file object again open the fileobj in read mode and display the 'appending' output.

```
c=fileobj.mode
print("file mode",c)
>>>file mode, w
d=fileobj.seek(0)
print("seek space",d)
>>>('seek space:', )

# w+ mode
fileobj=open("abc.txt","w+")
file_obj.write("looking gfg!!")
fileobj.close()

# r+ mode
fileobj=open("abc.txt","r+")
  s1=fileobj.read(6)
print("Output of r+",s1)
fileobj.close()
>>>('Output of r+', 'looking')

# append mode
fileobj=open("abc.txt","a")
fileobj.write("data structure")
fileobj.close()
fileobj=open("abc.txt","r")
  s3=fileobj.read()
print("output of append mode:",s3)
  >>fileobj.close()
>>>('output of append mode:', 'looking gfg! data structure')

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

#read mode
fileobj=open("abc.txt","r")
s=fileobj.read()
print("Output of read mode!",s)
>>>('Output of read mode!',
'looking gfg!')
```


Scanned with CamScanner

```
#tell
fileobj=open("abc.txt","r")
pos=fileobj.tell()
print("tell() :",pos)
fileobj.close
>>> ('tell() : ', 0 2 )

# seek()
fileobj = open("abc.txt","r")
st = fileobj.seek(0,0)
print("seek(0,0) fg : ",st)
fileobj.close
>>>('seek(0,0) fg : ', none)
fileobj = open("abc.txt","r")
st1 = fileobj.seek(0,1)
print("seek(0,1) fg : ",st1)
fileobj.close()
>>> ('seek(0,1) fg : ', none)

fileobj =open("abc.txt","r")
st2 = fileobj.seek(0,2)
print("seek(0,2) fg : ",st2)
file obj.close()
>>> ('seek(0,2) fg : ', none)

# Finding length of different lines exist within lines

fileobj =open("abc.txt","r")
stat = fileobj.readlines()
print("output : " stat)
for line in stat :
        print(len(line))
fileobj.close()

>>> ('output :', ['100kik sir data structure'])
26
```


Step 7 : Open the file in read mode. declare a variable and perform fileobject tell method and store the output consequently in variable.

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

Step 9 : Open file object with readmode also use the readlines() method and store the output consequently in and print the same for counting the length use the for conditional statement and display the length.

Scanned with CamScanner

iter() and next()

```
mytuple = ("banana", "orange", "apple")
myiter1 = iter(mytuple)
print(next(myiter1))
myiter2 = iter(mytuple)
print(next(myiter2))
myiter2 = iter(mytuple)
print(next(myiter1))
```

Output

```
>>> banana
    orange
    apple
```

For loop

```
mytuple1 = ("kevin", "steve", "jack")
for x in mytuple1:
    print(x)
```

```
>>> kevin
    steve
    jack
```

Square and cube

```
def square(x):
    y = x * x
    return y
def cube(x):
    z = x * x * x
    return z

unit = [square, cube]
```


Practical No: 7

Objective : Demonstrate Iterators

Step 1 : Create a tuple with elements that we need to iterate using the iter and next method the number of time we use the iter and next method use will get the next iterating element in the tuple display the same.

Step 2 : The similar output can be obtained by using by using for conditional statement An iterable variable is to be declared is for loop will iterate

Step 3 : Define a function name square with a parameter which will obtain with a square value of the given number. In similar fashion declare cube to get the value raised 3 and return the same.

Step 4 : Call the declare function using function call

Scanned with CamScanner

Step 5 : Using for conditional statement specifying range use the list tuple containing with map method declare a 'lambda' i.e anonymous function and print the same

Step 6 : Declare a listnum variable and declare same elements then use the map method with help of lambda function give two argument display the output

Step 7 : Define a function even with a parameter then using conditional statement do check whether the number is even and odd and return respectively.

Step 8 : Define a class and within that define the iter() method which will initialize the first element within the container object.

Step 9 : Now use the next() and define the logic for display odd value

```
myobj = odd()
myiter = iter(myobj)
x = int(input("enter a number"))
for i in myiter:
    if(i<x):
        print(i)

for s in range(5):
    value = list(map(lambda x : x(s), funct1))
    print(value
```

Output

```
>>> [0,0]
    [1,1]
    [4,8]
    [9,27]
    [16,64]
```

map()

```
listnum = [0,4,5,7,9,11,13,20,19,25]
listnum = list(map(lambda x = x%.5, listnum))
print(listnum)
def even(x):
    if (x%.2 == 0):
        return "even"
    else :
        return "ODD"
list(map(even, listnum))
```

odd numbers :

```
class odd :
    def __iter__(self) :
        self.num = 1
        return self
```


Scanned with CamScanner

```
def __next__(self):
    num = self.num
    num = self.num + 2
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 : 15
1
3
5
7
9
11
15
```


Step 10 : Define an object of a class.

Step 11 : Accept an number from the user till which we want to display the odd numbers.

Scanned with CamScanner

Factorial :

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

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

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

#Factorial

```
class fact :
    def __iter__(self):
        self.f = 1
        return self
    def next(self):
        if self.f <= 10:
            num = self.f
            self.f += 1
            fac = 1
            for i in range(1, num+1):
                fac = fac * i
            print(self.f-1, "! = ", fac)
        else:
            raise StopIteration
```

Output

```
>>> f = fact()
>>> x = iter(f)
>>> x = next()
    1! = 1
>>> x.next()
    2! = 2
>>> x.next()
    3! = 6
```

19/12/19

Program:-

```
while True
    try
        x = int(input("Enter class"))
        break
    Except ValueError
        print("Enter numeric value")
```

Output

```
Enter class 467
```

Program:-

```
try:
    fO = open("abc.txt", "w")
    fO.write("Raghutosh royal")
except IOError
    print("error writing on the file")
else:
    print("operation carried out successfully")
    fO.close()
```

Output:

```
Operation carried out successfully
```


PRACTICAL NO. 3

Aim : Program to demonstrate exception handling

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

Step 1 : Use the try block and 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 and 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.

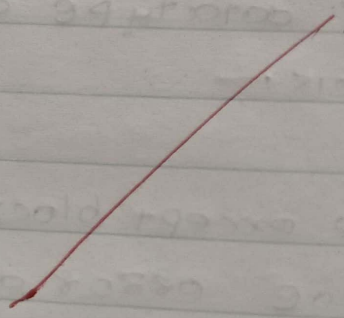
Step 1 : Within the try block open the file using the write mode and write some content on the file

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

Scanned with CamScanner

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 value error and print the message not a valid range



Scanned with CamScanner

Program:-

```
def assert_cn():
    assert (len(cn) == 0)
    print("list is empty")
var I = [ ]
print(assert_(var I))
```

Output:

List is empty

Program:-

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

Output

Enter age : 4
Not a valid age
Enter age : 18

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

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

Step 2 : close the function and in the body of program and define certain element in list and take appropriate

4] write a program to check the range of age of the students in given class and if the age do not fall in given range else the valueerror exception otherwise return the valid number.

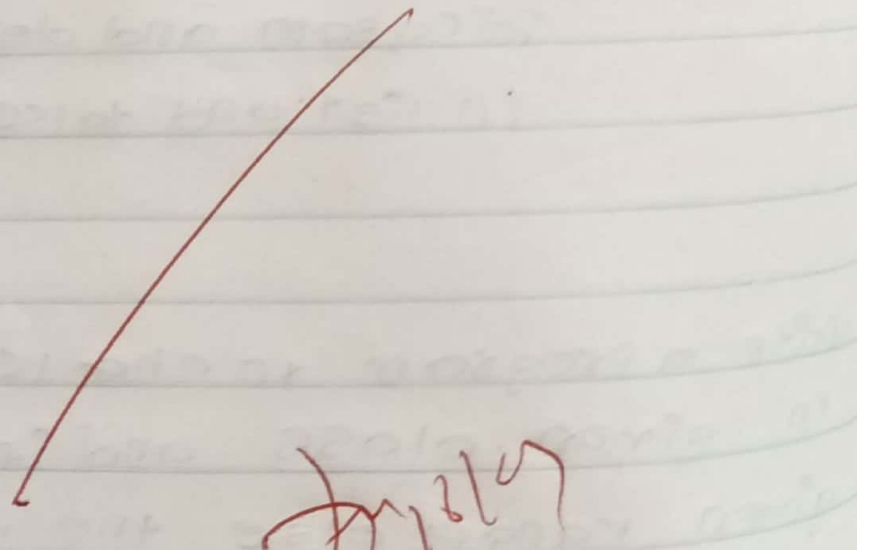
Step 1 : Define a function which will accept the age of the student from the standard input.

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

Scanned with CamScanner

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 value Error and print the message not a valid age.



Scanned with CamScanner

match()

```
import re
pattern = r"EYCS"
sequence = "EYCS represents computer science structure."
if re.match(pattern, sequence):
        print("matched patter found !!")
else:
        print("Not Found")
```

>>> matched pattern found !

Nomerical values (segreation)

```
import re
pattern = r'\d+'
string = 'hello123, kaddy789, 7891 hoo?'
output = re.findall(pattern, string)
print(output)
```

>>> ['123', '789', '7891']

split()

```
import re
pattern = r'\d+'
string = 'hello179, Rash789, Goyal234'
output = re.split(pattern, string)
print(output)
```

>>> ['Hello', 'aashu', 'goyal']

PRACTICAL NO. 4

TOPIC : Regular Expression

step 1 : import re module declare pattern and declare sequence use match method with declare arguments matched then print the same otherwise print Pattern NOT FOUND

step 2 : import re module declare pattern with nomeral and meta character. Declare string value. use the findall() with arguments and print the same.

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

Scanned with CamScanner

Step 4 : import re module declare string and accordingly declare pattern replace the blank space with no-space use sub() with 3 arguments and print the string without spaces.

Step 5 : Import re module declare a sequence use search method for finding subsequence use the group() with dot(.) operator as search() gives memory creation using group() it will show up the matched string.

Step 6 : import re module declare list with the numbers. use the conditional statement here we have used up for condition statement. use it 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 numbers matches otherwise print FAILED.

no-space :

```
import re
string 'abc dot ghi'
pattern = r'\s+'
replace = ''
v1 = re.sub(pattern, replace, string)
print(v1)
>> abcdotghi
```

Group()

```
import re
sequence = 'python is an interesting language'
v = re.search('\Apytho', sequence)
print(v)
  v1 = v.group()
print(v1)
>>> <_sre.SRE_Match object at 0x028810F00>
    python
```

verifying the given set of phone numbers

```
import re
lst = ['8004596292', '9891235671', '7770123793',
                                   '9820797239']
for value in lst:
    if re.match(r'[8-9][0-9][0-9]', value or
                                len(value) == 10):
        print("criteria matched for all numbers")
    else:
        print("criteria failed")
```


Scanned with CamScanner

```
>>> Criteria matched for cell number
    Criteria matched for cell number
    Criteria failed!
    Criteria matched for cell Number
```

vowels

```
import re
str = 'My name is Aashutosh'
output = re.findall(r'\b[aeiouAEIOU]\w+', str)
print(output)
```

```
>>> ['is', 'Aashutosh']
```

host & domain

```
import re
seq = 'abc.tcse@edu.com, xyz@gmail.com'
pattern = r'[\w.-]+@[\w.-]+'
output = re.findall(pattern, seq)
print(output)
```

```
>>> ['abc.tcse', 'edu.com', 'xyz', 'gmail.com']
```


Step 7 : import re module declare a string use the module with findall() for finding the vowels in the string and declare the same.

Step 8 : import re module declare the host and domain name declare pattern for seperating 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 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 or else increment value and display the value subsequently.

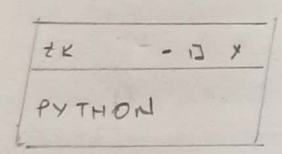
Scanned with CamScanner

38

Code :

```
from tkinter import *
root = Tk()
l = Label(root, text="PYTHON")
l.pack()
root.mainloop()
```

Output :



39

Practical No. 5 :

Aim : GUI components

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

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

Step 3 : Use the pack method along with the object created from text method.

Step 4 : Use the main loop method for triggering of comprending events.

Step 5 : Use the Tkinter library for importing the feature of text widget

Scanned with CamScanner

Step 6 : Create a variable from text method and position it onto parent window

Step 7 : use the pack method along with the object created from text method and use the parameters.

Step 8 : side = LEFT, padx = 20
side = LEFT, pady = 30
side = TOP, ipadx = 40
side = TOP, ipady = 50

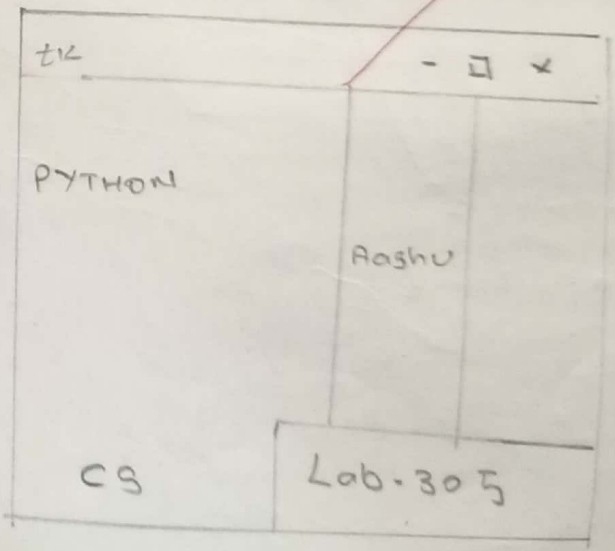
Step 9 : use the mainloop method for triggering the corresponding events

Step 10 : Now repeat the above steps with the label method which takes the Following arguments
(1) Text attribute which defines string
(2) Bg (Background) color
(3) Fg (Foreground) color
(4) Name of parent window
(5) use pack method with the padding attributes

code :

```
from tkinter import *
root = Tk()
l = Label(root, text="PYTHON")
l.pack(side=LEFT, pady=30)
l1 = Label(root, text="Aashu", bg="Yellow",
                                fg="Red")
l1.pack(side=TOP, ipady=40)
l2 = Label(root, text="CS")
l2.pack(side=LEFT, padx=20)
l3 = Label(root, text="Lab-305", bg="azure",
                                fg="Red")
l3.pack(side=LEFT, ipadx=50)
root.mainloop()
```

Output :



Scanned with CamScanner

Radio button

Q Write a program making use of the control variable and button widget for selection of the given option.

Step 1 : Use the tkinter method to import the relevant method.

Step 2 : Use the configuration method along with the label object and all the variables as an argument within the method.

Step 3 : Now define the parent window and define the option using control variable.

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

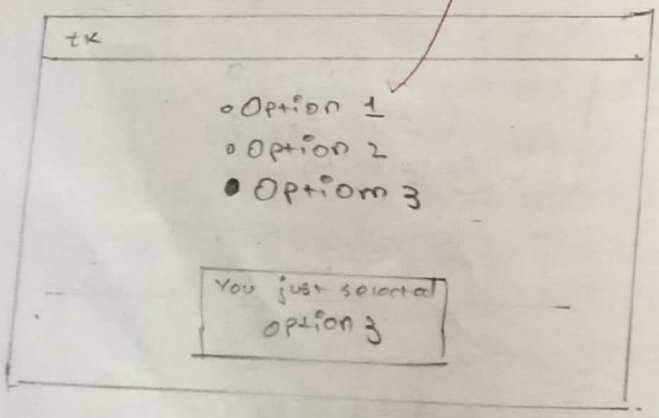
Step 5 : Now create an object from radio button method which will take the following arguments
(1) Positioning on the parent window
(2) Defining the text variable [1, 2, 3, 4]
(3) Define the variable arguments.
(4) Corresponding value and trigger the given function.

04

Source code:

```
from tkinter import *
root = Tk()
def sel():
    selection = "You selected the option" + str(var.get())
    l = label.config(root, text = selection, justify = LEFT)
    l.pack(anchor = S)
    v = IntVar()
r1 = Radiobutton(text = "option 1", variable = "v",
r1.pack()                value = '1', command = sel)
r2 = Radiobutton(text = "option 2", variable = "v",
r2.pack()                value = "2", command = sel)
r3 = Radiobutton(text = "option 3", variable = "v",
r3.pack()                value = '3', command = sel)
root.mainloop()
```

Output:



Scanned with CamScanner

Step 6 : Pack method for the corresponding radio objects so created and specify the attribute as an anchor attribute.

Step 7 : Now define the Label object from the corresponding method and place it on parent window.
Subsequently use pack method for this window and make use of mainloop() method.

Step 8 : Import the relevant method from Tkinter library.

Step 9 : Define the object corresponding to the object window and define the size of parent window in terms of the number of pixels.

Step 10 : Now define the frame object from the method and place it onto the parent window.

Step 11 : Create another frame object the LEFT Frame and put it onto window on its LEFT side.

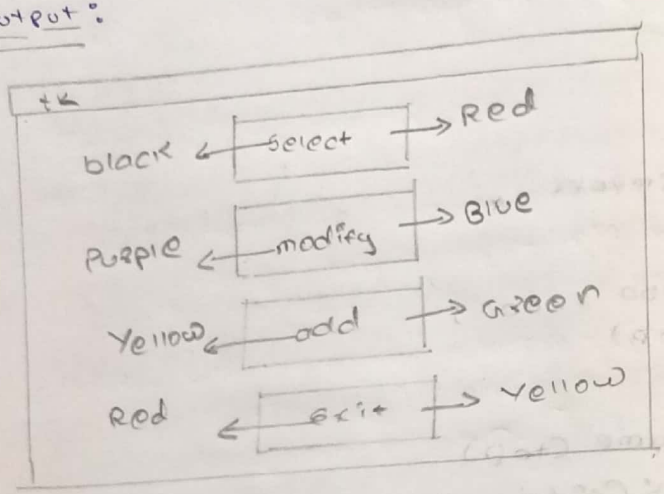
Frame object

Source code :

```
#Frame object
from TKinter import *
top = TK()
top.geometry('100x200')
frame = Frame(top)
frame.pack()
LEFT frame = Frame(top)
LEFT frame.pack(side = LEFT)
b1 = Button(frame, text = "select", activebackground
                                    = "red", fg = "black")
b1.pack()
b2 = Button(frame, text = "modify", activebackground
b2.pack()                           = "blue", fg = "purple")
b3 = Button(frame, text = "Add", activebackground
                                    = "green", fg = "yellow")
b3 = pack()
b4 = Button(frame, text = "EXIT", activebackground
                                    = "Yellow", fg = "red");
b4.pack()
top.mainloop()
```

Scanned with CamScanner

Output :



Step 12 : Similarly define the right frame and subsequently define the button object placed onto the given frame with the attribute as text active background (bg) and foreground (fg).

Step 13 : Now use the pack method along with the side attribute.

Step 14 : Similarly create the button object corresponding to modify option and put it onto the frame object with side equal to right attribute set.

Step 15 : Add another button and put it on right frame object and turn it as EXIT.

Step 16 : Use the pack method for all the objects and finally use the mainloop method

Scanned with CamScanner

PRACTICAL NO: 5B

Aim: GUI components

#1

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

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

Step 3: Now define a function which tells the user about 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 listbox() and insert options on the parent window along with the pack() with specifying anchor attribute.

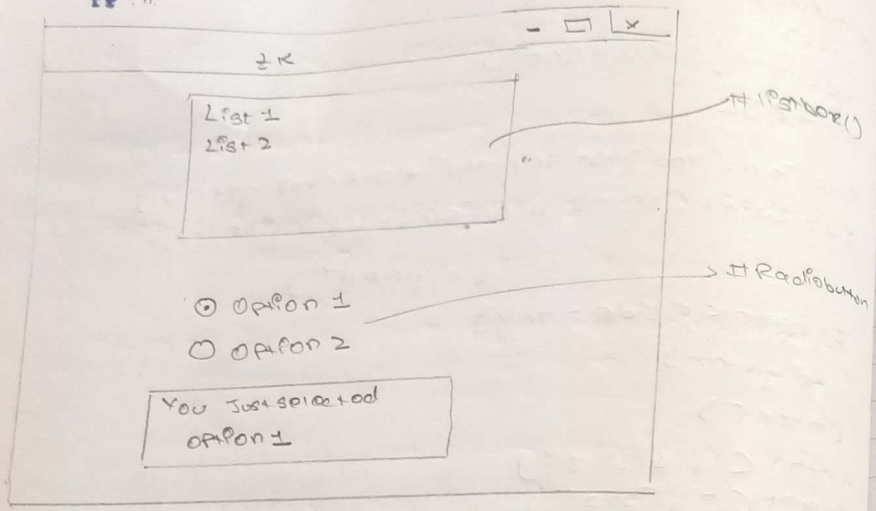
Step 6: Create an object from radio button which take following arguments parent window object, text variable which will take the values of option no. 1, 2, 3 variable argument, corresponding value & trigger the function declared.

Radio button

```
from tkinter import *
root = Tk()
root.geometry("500x500")
def select():
    selection = "You just selected" + str(var.get())
    l1 = Label(text = selection, bg = "white",
               fg = "green")
    l1.pack(side = TOP)
var = StringVar()
d1 = Listbox()
d1.insert(1, "list 1")
d1.insert(2, "list 2")
d1.pack(anchor = N)
r1 = RadioButton(root, text = "option 1", variable = var,
                 value = "option", 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()
```

Scanned with CamScanner

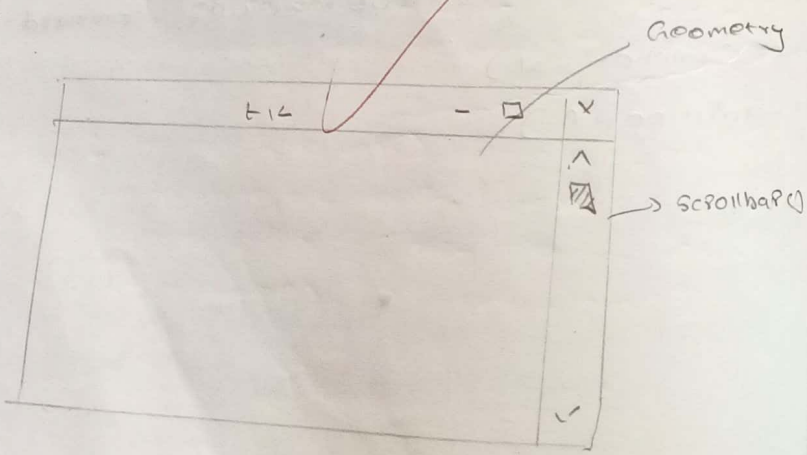
Output:-



tk2 (scrollbar()) :-

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

Output:



Step 7: Now call the pack() for radio object so created and specify the argument using anchor attribute.

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

2 (Scrollbar):

Step 1 : Import relevant method from the tkinter library

Step 2 : Create a parent object corresponding to parent window

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

Step 4 : Create an object and use the Scrollbar()

Step 5 : use the pack() along with the scrollbar object with side and fill attributes

Step 6 : use the mainloop with the parent object.

Scanned with CamScanner

3 (Using frame widget)

Step 1 : Import the relevant libraries from the tkinter module

Step 2 : Create a corresponding object of the main window.

Step 3 : Use the geometry manager with pixel size or any other suitable pixel value

Step 4 : Use the label widget along with the parent object created and subsequently use the pack method.

Step 5 : Use the frame widget along with the parent object created and use the pack method

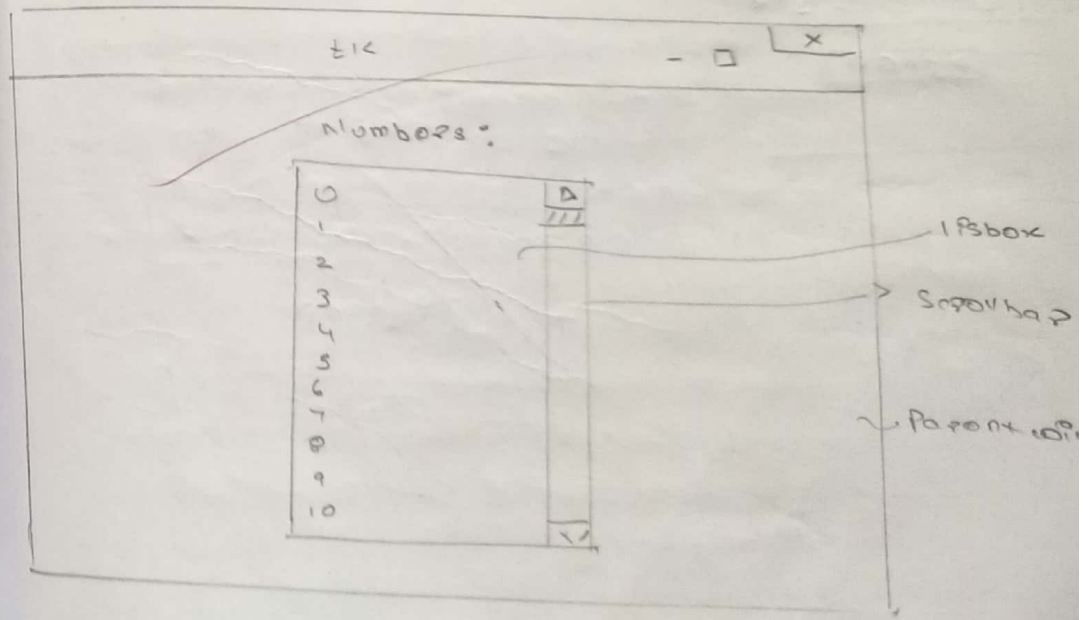
Step 6 : Use the Listbox method along with the attributes like width height font. Do create a listbox methods object use pack() for the same

Step 7 : Use the Scrollbar() with an object use the attribute of vertical then configure the same with object created from the Scrollbar() and use pack()

Step 8 : Trigger the events using mainloop.


```
from tkinter import *
window = Tk()
window.geometry("650x500")
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:

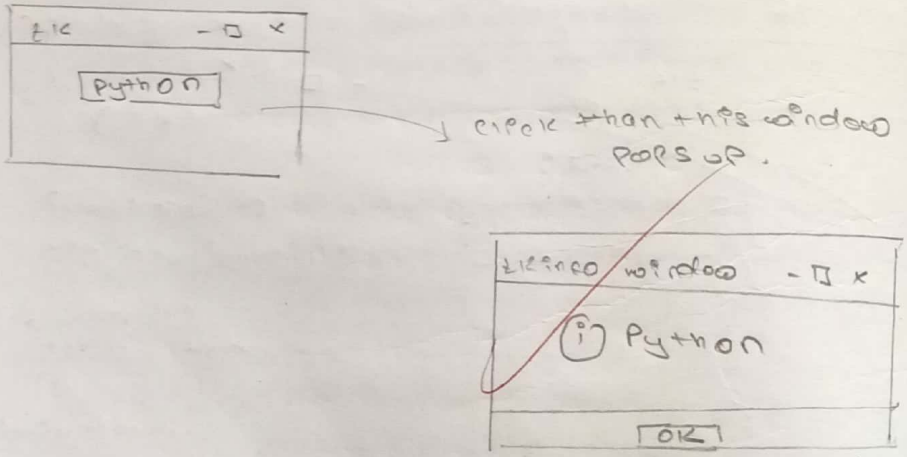


Scanned with CamScanner

message box

```
from tkinter import *
import tkmessageBox
root = Tk()
def function():
    tkmessageBox.showinfo("info window", "Python")
    b1 = Button(root, text = "Python", command = function)
b1 = Button(root, text = "Python", command = function)
b1.pack()
root.mainloop()
```

Output:



message box:

Step 1: Import the relevant methods from tkinter library.

Step 2: Import tk messagebox

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 dialog 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 and finally call mainloop() for triggering of the events called above.

Scanned with CamScanner

47

PRACTICAL NO : 5 C

- Transversing and making use of geometry layout manager method

Step 1 : Define a function and create a object to given window by using the three methods namely config, title and minsize

Step 2 : Create a button object and the text and command attribute for triggering the given event and used grid method with frame and external packing specification similarly create another button object which will allow application to terminate.

Step 3 : Define second function corresponding to second window with attributes config title .minsize for the window object and define one button obj which will shift the Process to third window

Step 4 : Create third window object and for this create two button object for making on to first window for restarting the process and second button for terminating.

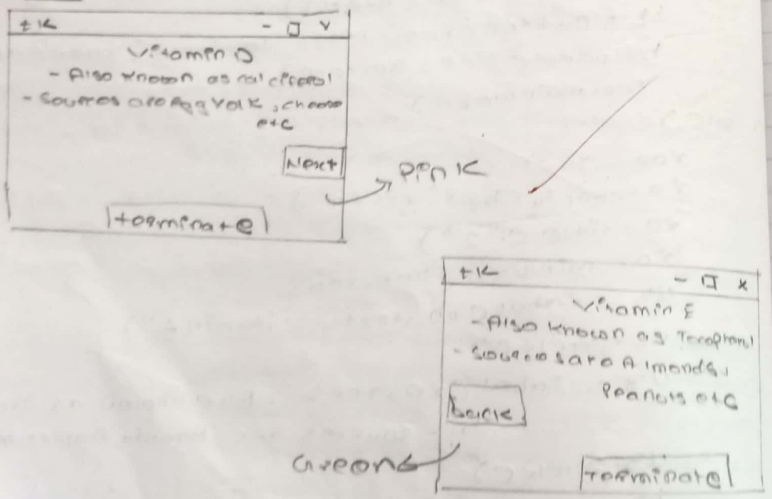
Source code

```
from tkinter import *
root = Tk()
def main():
    root = Tk()
    root.config(bg='pink')
    root = title("main")
    root = minsize(200,200)
    l = Label(root, text = "Vitamin D")
    l.pack()
    l1 = Label(root, text = "- Also known as calciferol
                - Sources are egg yolk, cheese etc")
    l1.pack
    b1 = Button(root, text = "next", command = se)
    b1.pack(side = RIGHT)
    b2 = Button(root, text = "terminate", command = ter)
    b2.pack(side = BOTTOM)
    root.mainloop()
def sel():
    ro = Tk()
    ro.config(bg = "green")
    ro.title("2")
    ro.minsize(400,200)
    l2 = Label(ro, text = "Vitamin E")
    l2.pack()
    l3 = Label(ro, text = "- Also known as Tocopherol
                - Sources are Almonds, Peanuts etc")
    l3.pack()
    b3 = Button(ro, text = "back", command = main
    b3.pack(side = LEFT)
```


Scanned with CamScanner

```
b4=Button(root,text="terminate",
          command=test)
b4.pack(side=BOTTOM)
root.mainloop()
def test():
    quit()
b5=Button(root,text="know your vitamins",
          command=main)
b5.pack()
root.mainloop.
```

Output :



Steps : Define a function for termination and call the quit method and finally call the first function created and the second mainloop method.

* Displaying the image

Algorithm:

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

Step 2 : Create a leftframe object from the frame method and place it onto the parent window with 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 column value should be specified.

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

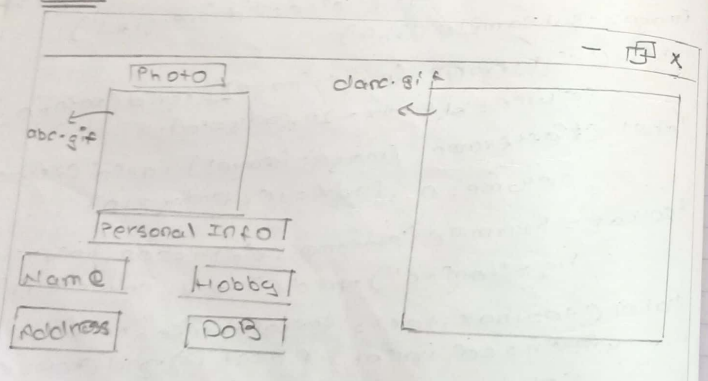
Code :

```
from tkinter import *
root = Tk()
root.title("Python")
root.maxsize(1000,900)
root.config(bg='black')
leftframe = Frame(root, bg="pink", height="400",
                  width="200")
leftframe.grid(row=0, column=0)
rightframe = Frame(root, bg="red", height="400",
                   width="250")
rightframe.grid(row=0, column=0)
label(leftframe, text="Photo", height=2, width=20).
      grid(row=0, column=0)
image1 = PhotoImage(file="abc.gif")
image1.subsample(1,2)
image2 = PhotoImage(file="dance.gif")
image2.subsample(3,4)
Label(rightframe, image=image2).grid(row=0,
      column=0, padx=20, pady=0)
label(leftframe, image=image1).grid(row=0,
      column=0, padx=10, pady=10)
toolbar = Frame(leftframe, width=200, height=100,
          bg="white").grid(row=2, column=0)
label(toolbar, text="Personal Info", height=2,
      width=20, relief=RAISED).grid(row=0,
      column=0, padx=20, pady=0)
def name():
    print("Name :")
def hob():
    print("Hobby :")
```

Scanned with CamScanner

```
def add():
    print("Address : mumbai")
def dob():
    print("DOB : 20/08/2000")
Button(toolbar,text="Name",height=1,width=6,
    command=name).grid(row=1,column=0)
Button(toolbar,text="Hobby",height=1,width=6,command=
    hob).grid(row=1,column=1)
Button(toolbar,text="Address",height=1,width=6,
    command=add).grid(row=2,column=0)
Button(toolbar,text="DOB",height=1,width=6,
    command=DOB).grid(row=2,column=1)
root.mainloop()
```

Output :



Step 5 : Now use the photoImage method with the file attribute specified.

Step 6 : use the subsample method with the object of the image and give the x,y coordinate values.

Step 7 : use the label method and position it onto the leftframe and placing the image using 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 specifying it as (0,0)

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

Step 10 : Now define the hobby function for different toolbar options provided in the leftframe.

Scanned with CamScanner

18

Step 11 : From the Label method position the text onto the toolbar use the rowspan and corresponding grid value and incorporate the internal padding.

Step 12 : Create the Label method position for the toolbar with the next title as personal information and position the same row but next column.

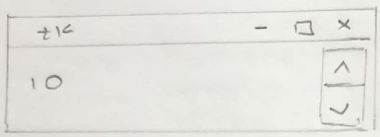
Step 13 :- Now make use of mainloop method.

52

Source Code:

#Spinbox :-

```
from tkinter import *
master = Tk()
sb = Spinbox(master, from_=0, to=10)
sb.pack()
master.mainloop()
```



53

PRACTICAL NO. 5(D)

- Write a program to make use of spinbox widget.

Step1 : Use of Tkinter library to import the relevant methods

Step2 : Create the parent window object

Step3 : Create an object from the spinbox and place it onto the parent window with the option specified.

Step4 : Now use the pack method to make make the object visible onto the parent window and call the mainloop method.

Scanned with CamScanner

23

#Paned window

Step1 : Create an object from the panedwindow() and use the pack() to make this object visible.

Step2 : Now create an object from entry widget and place it on to the paned window and use the add method.

Step3 : Similarly create an object of the paned window and add it on to the existing window.

Step4 : Create an object from the scale method and place it on to the previously paned window and use the add method accordingly

Step5 : Create a Button widget on to the paned window and define a functionality along with button event.

Step6 : use the pack and mainloop method corresponding unit to be triggered

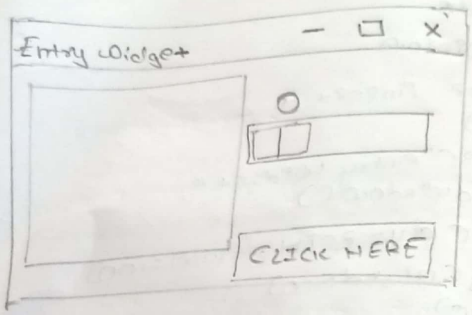
54

Source code

#Panedwindow

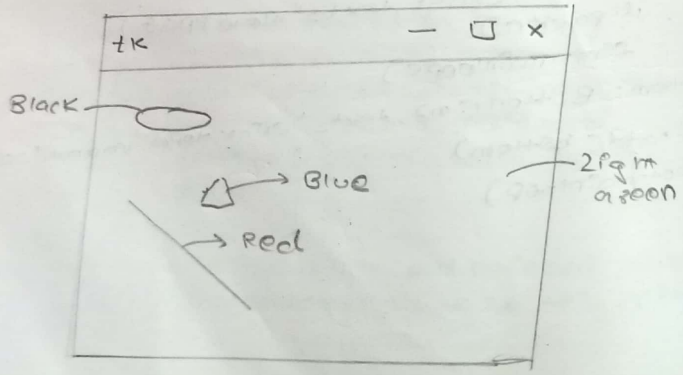
```
from tkinter import *
root = Tk()
root.title("Entry widget")
m1 = PanedWindow()
m1.pack(fill=BOTH, expand=1)
e = Entry(m1, bd=10)
m1.add(e)
m2 = PanedWindow(m1, orient=VERTICAL)
m1.add(m2)
top = Scale(m2, orient=HORIZONTAL)
m2.add(top)
def sel():
    first = Tk()
    first.title("New Page")
    first.minsize(200,200)
    first.config(bg='Red')
    l = Label(first, text="NEW PAGE")
    l.pack()
    first.mainloop()
bottom = Button(m2, text="Click Here", command=sel)
m2.add(bottom)
root.mainloop()
```

Scanned with CamScanner



#. Canvas

```
from tkinter import *
root = Tk()
c = canvas(root, height = 100, width = 200, bg = "Light Green")
arc = c.create_arc(10,20,30,40, start = 0, extent = 50, fill = "Blue")
line = c.create_line(10,31,45,50, fill = "Red")
oval = c.create_oval(10,15,30,20, fill = "Black")
c.pack()
root.mainloop()
```



Canvas:

Step 1 : Creat an object from the canvas widget by using the attribute height, width, background colour and the parent window object

Step 2 : use the corresponding method for drawing the simple geometrical shape like arc, oval and line. Specify the coordinate values.

Step 3 : Similarly use the create_line and create_oval method alongwith coord values and fill attribute for specifying the colour.

Step 4 : finally use pack and the mainloop method.

Scanned with CamScanner

PRACTICAL NO. 6

Aim : Database connectivity

 # 1 :

Step 1 : Import the (DBM) dbm library & use the open() for creating the database by specifying the name of the database along with 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 not equal to None then display the message that particular found / match or else not found / unmatched.

Step 4 : use the close() to terminate database library.

 # 1 :

```
>>> import dbm
>>> db = dbm.open("database", flag='c', mode=0o666)
>>> db["name"] = "name"
>>> if db["name"] != None :
        print("Database Not empty //match !")
    else :
        print("Database empty ! //Not match")
Database Not empty //match !
>>> db.close()
```

Scanned with CamScanner

#2 :

```
import sqlite3
conn = sqlite3.connect("employee-db")
cur = conn.cursor()
cur.execute("create table emp (Name char, RollNo int)")
cur.execute("insert into emp values ('Aashutosh', 1818)")
                                    ('Ayush', 1111)")
conn.commit()
cur.execute("select * from emp")
print(cur.fetchall())
conn.close()
```

Output :

```
[('Aashutosh', 1818), ('Ayush', 1111)]
```

#2:

step 1 : Import corresponding library to make database connection i.e. sqlite3

step 2 : Now create the connection object using SQL the .3 library & the connect() for creating New database obj.

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 the column name & respective 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

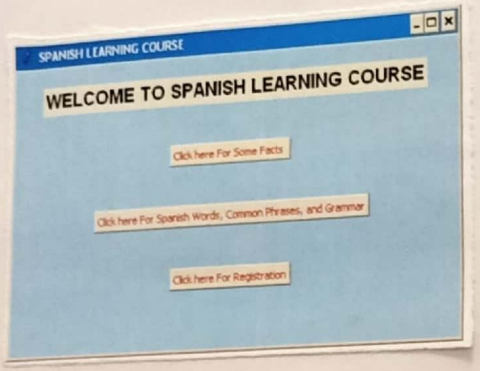
step 7 : use the execute statement alongwith cursor object from accessing the values from the database using the select from table clause.

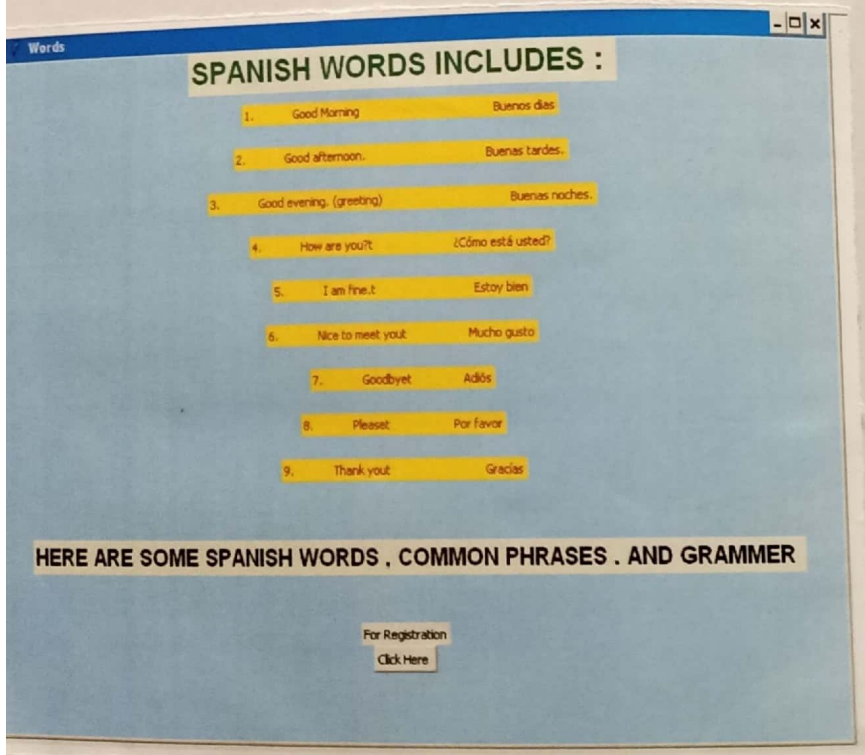
Scanned with CamScanner

Step 8 : Finally use the fetch() or fetchall() for displaying the values from the table using the cursor_object.

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

Scanned with CamScanner

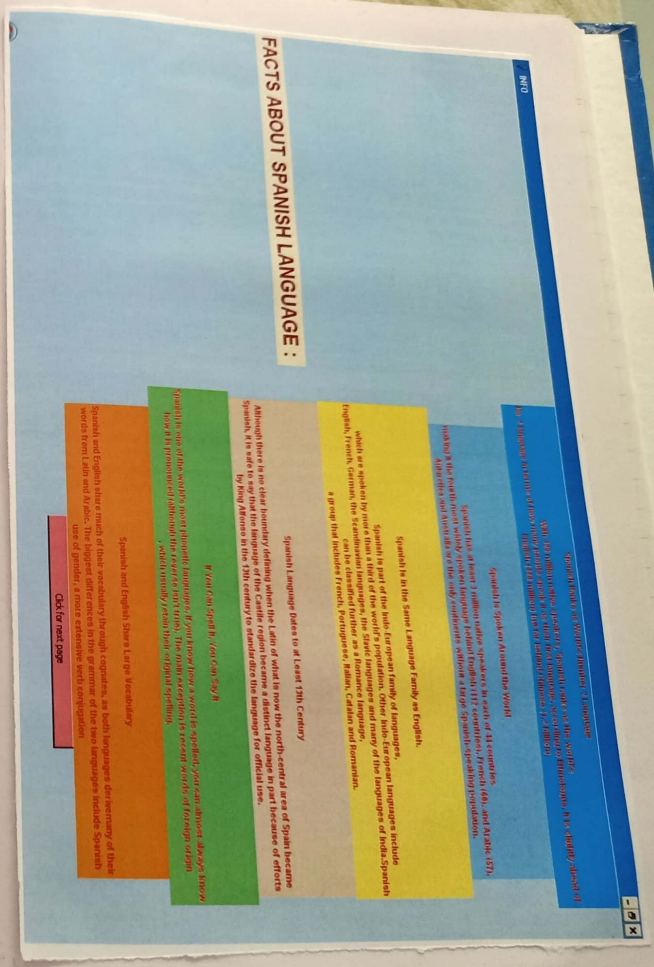




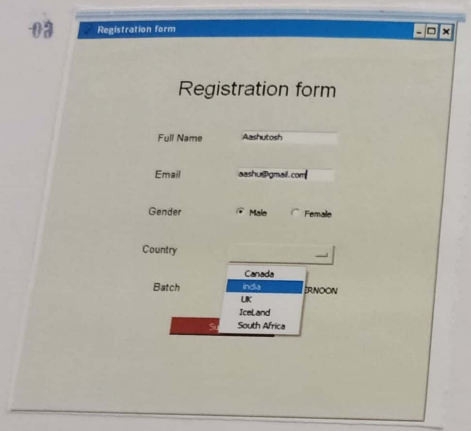

```
from tkinter import *
from tkinter import messagebox
top=Tk()
top.geometry("500x300")
top.config(bg="powder blue")
top.title("SPANISH LEARNING COURSE")
Label(top,text="WELCOME TO SPANISH LEARNING
COURSE",font=('arial',15,'bold')).pack(padx=20,pady=20)
def second():
  second=Tk()
  second.title("INFO")
  second.geometry("1200x800")
  second.config(bg="powder blue")
  Label(second,text="FACTS ABOUT SPANISH LANGUAGE : ",font=('arial',20,'bold'),fg="steel
blue").pack(side=LEFT)
  Label(second,text="Spanish Ranks as World's Number 2 Language\n\nWith 329 million native
speakers, Spanish ranks as the world's"
    "\nNo. 2 language in terms of how many people speak it as their first language, according to
Ethnologue. It is slightly ahead of "
    "\nEnglish (328 million) but far behind Chinese (1.2
billion).",bg="cyan",font=('arial',8,'bold'),fg="Red").pack()
  Label(second,text="\n\nSpanish Is Spoken Around the World \n\nSpanish has at least 3 million native
speakers in each of 44 countries"
    "\nmaking it the fourth-most widely spoken language behind English (112 countries), French (60),
and Arabic (57)."
    "\nAntarctica and Australia are the only continents without a large Spanish-speaking
population.",font=('arial',8,'bold'),fg="Red",bg="sky blue").pack()
  Label(second,text="\n\n\nSpanish Is in the Same Language Family as English.\n\nSpanish is part of
the Indo-European family of languages,"
    "\nwhich are spoken by more than a third of the world's population. Other Indo-European
languages include"
    "\nEnglish, French, German, the Scandinavian languages, the Slavic languages and many of the
languages of India.Spanish"
    "\ncan be classified further as a Romance language,"
    "\na group that includes French, Portuguese, Italian, Catalan and Romanian. ",bg="light
goldenrod",font=('arial',8,'bold'),fg="Red").pack()
  Label(second,text="\n\n\nSpanish Language Dates to at Least 13th Century"
    "\n\nAlthough there is no clear boundary defining when the Latin of what is now the north-central
area of Spain became"
    "\nSpanish, it is safe to say that the language of the Castile region became a distinct language in
part because of efforts "
    "\nby King Alfonso in the 13th century to standardize the language for official
use.",bg="AntiqueWhite3",font=('arial',8,'bold'),fg="Red").pack()
  Label(second,text="\n\nIf You Can Spell It, You Can Say It"
    "\n\nSpanish is one of the world's most phonetic languages. If you know how a word is spelled,
you can almost always know"
    "\nhow it is pronounced (although the reverse isn't true). The main exception is recent words of
foreign origin"
    "\n, which usually retain their original
spelling.",font=('arial',8,'bold'),fg="Red",bg="PaleGreen3").pack()
  Label(second,text="\n\nSpanish and English Share Large Vocabulary"
```

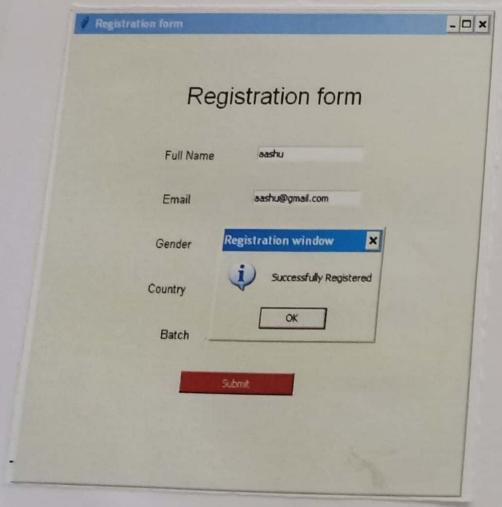
Scanned with CamScanner

```
        "\n\nSpanish and English share much of their vocabulary through cognates, as both languages
derivemany of their"
        "\nwords from Latin and Arabic. The biggest differences in the grammar of the two languages
include Spanish "
        "\nuse of gender, a more extensive verb
conjugation",bg="Orange3",font=('arial',8,'bold'),fg="Red").pack()
    Button(second,text="Click for next page",command=word).pack(ipadx=100)



def word():
    word=Tk()
    word.geometry("800x600")
    word.title("Words")
    word.config(bg="Powder Blue")
    Label(word,text="SPANISH WORDS INCLUDES : ",fg="forest green",font=('arial',20,'bold')).pack()
    Label(word,text="1.\tGood Morning \t\t\t Buenos dias",bg="gold",fg="magenta4").pack(pady=10)
    Label(word,text="2.\tGood afternoon. \t\t\t Buenas
tardes.",bg="gold",fg="magenta4").pack(pady=10)
    Label(word,text="3.\tGood evening. (greeting) \t\t\t Buenas
noches.",bg="gold",fg="magenta4").pack(pady=10)
    Label(word,text="4.\tHow are you?t\t\t ¿Cómo está
usted?",bg="gold",fg="magenta4").pack(pady=10)
    Label(word,text="5.\tI am fine.t\t\tEstoy bien",bg="gold",fg="magenta4").pack(pady=10)
    Label(word,text="6.\tNice to meet yout\t\tMucho gusto",bg="gold",fg="magenta4").pack(pady=10)
    Label(word,text="7.\tGoodbyet\t\tAdiós",bg="gold",fg="magenta4").pack(pady=10)
    Label(word,text="8.\tPleaset\t\tPor favor",bg="gold",fg="magenta4").pack(pady=10)
    Label(word,text="9.\tThank yout\t\tGracías",bg="gold",fg="magenta4").pack(pady=10)
    Label(word,text="HERE ARE SOME SPANISH WORDS , COMMON PHRASES . AND GRAMMER
",font=('arial',15,'bold')).pack(pady=40)
    Label(word,text="For Registration").pack()
    Button(word,text="Click Here ",command=form).pack()
    word.mainloop()


def msg():
    messagebox.showinfo("Registration window","Successfully Registered")
```



Scanned with CamScanner






```
def form():
    root=Tk()
    root.geometry("500x500")
    root.title("Registration form")
    global entry
    entry=StringVar()
    Label0=Label(root,text="Registration form",width=20,font=("bold",20))
    Label0.place(x=90,y=53)
    Label1=Label(root,text="Full Name ",width=20,font=("bold",10))
    Label1.place(x=80,y=130)
    entry1=Entry(root,textvariable=entry)
    entry1.place(x=240,y=130)
    label2=Label(root,text="Email",width=20,font=("bold",10))
    label2.place(x=68,y=180)
    entry2=Entry(root)
    entry2.place(x=240,y=180)
    label3=Label(root,text="Gender",width=20,font=("bold",10))
    label3.place(x=70,y=230)
    var=IntVar()
    Radiobutton(root,text="Male",padx=5,variable=var,value=1).place(x=235,y=230)
    Radiobutton(root,text="Female",padx=20,variable=var,value=2).place(x=290,y=230)
    label4=Label(root,text="Country",width=20,font=("bold",10))
    label4.place(x=70,y=280)
    list1=['Canada','india','UK','IceLand','South Africa']
    c=StringVar()
    droplist=OptionMenu(root,c,*list1)
    droplist.config(width=15)
    c.set('Select Your Country')
    droplist.place(x=240,y=280)
    Label5=Label(root,text="Programming",width=20,font=("bold",10))
    Label5.place(x=85,y=330)
    var1=IntVar()
    Checkbutton(root,text="JAVA",variable=var1).place(x=235,y=330)
    var2=IntVar()
    Checkbutton(root,text="PYTHON",variable=var2).place(x=290,y=330)
    Button(root,text="Submit",width=20,bg='Brown',fg="white",command=msg).place(x=180,y=380)
    root.mainloop()
b2=Button(top,text="Click here For Some Facts ",fg="Red",command=second).pack(pady=20)
b3=Button(top,text="Click here For Spanish Words, Common Phrases, and
Grammar",fg="red",command=word).pack(pady=20)
b4=Button(top,text="Click here For Registration",fg="Red",command=form).pack(pady=20)
top.mainloop()
```

Scanned with CamScanner