Practical No. 1

Objective: Demonstrate the use of different file accessing modes, different attribute read method

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

step 2: Now open the file in read mode and than use read(), readline(), and readlines(), and store the output in variable and finally display the contents of variable.

step 3: Now use the file object for finding the name of the file, the file mode in which it is opened whether the file is still open or close and finally the output of the softspace attribute.

Scanned with CamScanner

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

fileobj = open("abc.txt", "r") # read mode
# read()
str1 = fileobj.read()
Print("The output of read method:", str1)
fileobj.close()
>>> ("The outputs of read method:', 'computer
    science subjects in DBMS in Python
    in DS\n')

# readline()
fileobj = open("abc.txt", "r")
str2 = fileobj.readline()
Print("The output of readline method:", str2)
fileobj.close()
>>> ('The output of readline method:', 'computer scien
    subject\n')

# readlines()
fileobj = open("abc.txt", "r")
str3 = fileobj.readlines()
Print("The output of readlines method:", str3)
fileobj.close()
>>> ('The output of readlines method:', ['computer
    science subjects\n', 'DBMS\n', 'Python\n',
    'DS\n'])
```

Scanned with CamScanner

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

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

Step 6: Now open fileobject in append mode open write method. write content close the file object. again. Open the file object in read mode and display the append output

Scanned with CamScanner

file attributes

```
a = fileobj.name
print ("name of file (name attribute): ", a)
>> ('name of file (name attribute)', 'abc.txt')

b = fileobj.closed
print ("(close) attribute : ", b)
>> ('(close) attribute:', 'True')

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

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

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

```
# r+ mode
fileobj = open ("abc.txt", "r+")
str1 = fileobj.read (8)
print ("output of r+", str1)
fileobj.close ()
>>> ('output of r+', 'saurabh')
```

```
# read mode
fileobj = open ("abc.txt", "r")
str2 = fileobj.read ()
print ("output of read mode!", str2)
>>> ("output of read mode", 'saurabh')
```

Scanned with CamScanner

```
# 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:', 'Sauttabh', 'data structure')

# 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
fileobj = open("abc.txt", "r")
str9 = fileobj.readlines()
print("output:" str9)
for line in str9:
        print(len(line))
    fileobj.close()
>>> ('output:', ['college database'])
```

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

step 8: Use the seek method with the arguements with opening the fileobject in read mode and closing subsequently in and print the same

step 9: Open fileobject with read mode also use the readlines method and store the output su consequently in and print the same for counting the length. Use the for condition statement and display the length.

Nafi
Jmnu

Scanned with CamScanner

BSIS

Practical 2

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

Algorithms:

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

Step II: For extracting the next element from the container with an arguement and compare the number of elements required in a container by using the conditional statement.

Step III: Now create an object from the given class and pass this object as arguement to the iter() method

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

Scanned with CamScanner

```
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
```

no space

```
myobj = odd()
myiter = iter(myobj)
for num in myiter:
    print(num)
```

Output:

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

97
99
```

Scanned with CamScanner

```
# iter() and next()
mytuple1 = ("apple", "orange", "banana")
myiter1 = iter(mytuple1)
print(next(myiter1))
myiter2 = iter(mytuple1)
print(next(myiter2))
myiter3 = iter(mytuple1)
print(next(myiter3))
```

Output
>> apple
orange
banana.

```
# for loop:
mytuple1 = ("Tom", "staart", "Jerry")
for x in mytuple1:
    print(x)
```

Output.
>>> Tom
staart
Jerry.

```
# square and cube:
def square(x):
    y = x * x
    return y
def cube(x):
    z = x * x * x
    return z
funct
Co-unt1 = [square, cube]

for x in range[5]:
    valueout = list(map(lambda x: x(x), funct))
    print(valueout)
```

Practical - 2

Aim: To study about iterators.

Algoritm:

Step I: 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 we will get the next iterating element in the tuple. Display the same.

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

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

Step IV: Call the declare function using function.

Scanned with CamScanner

Step V: Using for conditional statement specifying the range, use the list typecasting with map method, declare a lambda function, i.e. anonymous function and print the same.

Step VI: Declare a listnum variable and declare some elements and use the map method with help of lambda function give two arguments. Display the output.

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

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

Step IX: Now use the next() and define the logic for displaying odd value.

Scanned with CamScanner

```
for x in range[x]:
    value = list(map(lambda x: x(x), fun()))
    print(value)
>>> [0,0]
    [1,1]
    [4,5]
    [9,27]
    [16,24]

# map():
listnum = [0,2,3,4,5,6,7,8]
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, list num))
>>> [0,2,4,6,8]

# Odd Number:
    class odd
        def __iter__(self):
            self.num = 1
        def __next__(self):
            num.self = num
            num.self += 2
        def __next__(self):
            num=self.num
            ~~num.self += 2~~
            self.num+= 2
            return num
```

Scanned with CamScanner

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

```
>>> Enter a number: 15
1
3
5
7
9
11
13
```

Factorial:

```
def f(n):
    if (n==1):
        return n
    else:
        return (n * f(n-1))
n = int(input("Enter a number"))
list = []
list.append(n)
a = map(f, list)
print("The factorial is:", a)
```

```
>>> Enter a number: 5
The factorial is: 120
```

step X : Define an object of a class.

step XI : Accept an number from the user till which user want to display the odd number.

step XII : Define a function with if condition if exactly equal to one return else on Parameter.

step XIII : Enter a input from the declare an empty list. Use append method for appending the input value. Print the output variable.

1/11/19

Scanned with CamScanner

Practical No 3

Exception:

Algorithm:

Step 1: Use the try block to define the normal course of action. for eg. Define the file obj and open the file in the write mode and write some content onto the file.

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

Scanned with CamScanner

Program


```
try:
    fo = open("abc.txt", "r")
    fo.write("hi, this is Khushboo")
except IOError:
    Print("Error occured under Environmental
                                    error")
else:
    Print("Operation successfull")
```

Output:

Environmental Error.

Scanned with CamScanner

Program:

```
try:
    a=int (input ("Enter the number"))
except. valueError:
    Print ("Arithematic error")

else:
    Print ("successfull")

try:
    fo=open ("abc.txt", "r")
    fo. write ("hello, how are you??")
except IOError:
    Print ("Environmental error")
else:
    Print ("successfull")
```

Output:

Enter number=14

Arithematic error.

Scanned with CamScanner

Program for demonstrating the use of value error in the given program statement.

Algoritm :

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

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

step 3: You can define the multiple exception using the except statement for finding the different category of errors.

Scanned with CamScanner

Practical 4

Aim: Demonstrate the use of regular Expression

Step I: Impat re module declare Pattern and sequence. Use match method with declare arguements matched then Print the same otherwise Print Pattern not found.

Step II: Import re module declare Pattern and with literals and meta characters. Declare string value and use the findall() method with arguements and Print the same.

Step III: Import re module declare Pattern with literals and meta characters. Use the split() method and Print the output.

Step IV: Import re module declare string and accordingly declare Pattern and replace. To fill the blank space with no-space. Use sub() method with 3 arguements and Print the string without spaces.

Scanned with CamScanner

#1 match()

```
import re
pattern = r"Fycs"
sequence = "Fycs represents computer science stream".
if re.match(pattern, sequence):
    print("Matched Pattern found")
else:
    print("Print Not found")
```

>>> Matched Patern found.

#2 Findall()

```
import re
pattern = r'\d+'
string = 'Hello123', vicky476'
output = re.findall(pattern, string)
print(output)
```

>>> ['123', '476']

#3 split()

```
import re
pattern = r'd+'
string = 'Hello123, vicky476'
output = re.split(pattern, string)
print(output)
```

>>> ['Hello', 'vicky']

4 no_space or sub()

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

>>> abcdefghi

Scanned with CamScanner

```
#group()
import re
string = "Python is an interpreted language"
V = re.search('\A Python', string)
Print (v)
Print (v.group())
```

```
>>>  <_sre.SRE Match object at 0x02810F0>
Python.
```

6 verifying the given set of Cell No.

```
import re.
list = ["8007266524", "7779977551", "8208230693"]
for value in list1:
    if re.match (r'[8-9]{1} [0-9]{9}', value or
            len (value) == 10) :
        Print ("Cell Number found")
    else:
        Print ("Match Failed")
```

```
>>> Cell Number found
Match failed
Cell Number found
```

Vowel

```
import re
str1 = 'Hello Python is a Programming language'
output = re.findall (r'\b [aeiou AEIOU] \w+', str1)
Print (output)
```

```
>>> ['is', 'a'
```

Scanned with CamScanner

step V: Import re. module declare a sequence. Use search method for finding and subsequently. Use the group() with dot operator as search() gives memory location using group() it will show up the matched string

step VI: Verifying cell No.

step i]: Import re module and declare list with numbers.

step ii]: Use the conditional statement here we have used up the for condition statement.

step iii]: Use if conditional for checking 1st Number is either 8 or 9 and rest of the Numbers are in range of 0 to 9. i.e. rest 9 numbers.

step iv]: Check whether the numbers are equal to 10.

step v]: If criteria matches print cell numbers Matched otherwise print Match failed.

step VII: Import re module and declare a variable named str 1 and store some string value in it. With the help of findall() method and string and display the some on window.

Scanned with CamScanner

Step VIII : Import re module. Enter a string value and use the pattern to display only the elements of the particular string. Use findall() method and pass the pattern and string value as argument in it so as to check whether condition satisfies and Hence, point the output subsequently.

Step IX : Import re module. Enter a string value and use the pattern to display only the elements of the particular string. Use findall() and declare two variable with initial value as 0. Use for conditional statement. to go through all the values and by using if conditional statement check whether the condition satisfies or not if satisfies then increment the m else increment value of f. Hence display the same.

Scanned with CamScanner

- host domain

```
import re
seq= 'abc123.tscs @edureka.com, FYCS@edu.com'
pattern= r'[\w\. -]+[\w\. -]'
output= re.findall (Pattern, seq)
print (output)
```

```
>>> ['abc123.tcss'. 'eduréka.com', 'FYcs', 'edu.com']
```

count of first 2 letters

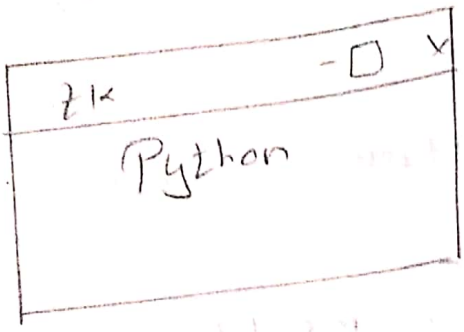
```
import re
s= 'mr.a, ms.b, ms.c, mr.d'
p= r'[ /m, /mr /]+
o = re.findall (P, S)
print (o)
m=0
f=0
for v in o:
    if (v== 'ms'):
        f=f+1
    else:
        m=m+1
    Print ("No. of Males:", m)
    Print ("No. of Females", f)
```

```
>>> ['Mr', 'Ms, 'Ms', , Mr']
['No. of Males": '2]
['No. of Females: '2]
```

Scanned with CamScanner

#1. Creation of Parent window

```
from Tkinter import *
root= TK()
l= Label (root, text= "Python")
l.Pack()
root.mainloop()
```



#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="black", font="20")
l2.Pack(side= LEFT, Pady="30")
l3= Label (root, text= "cs!", bg = "yellow",
            fg="black", font = "10")
l3.Pack (side=TOP, ipadx=40
```

Scanned with CamScanner

Practical 5

Topic: GUI Components

#1.

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 and use the text method.

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

#2:

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

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

Scanned with CamScanner

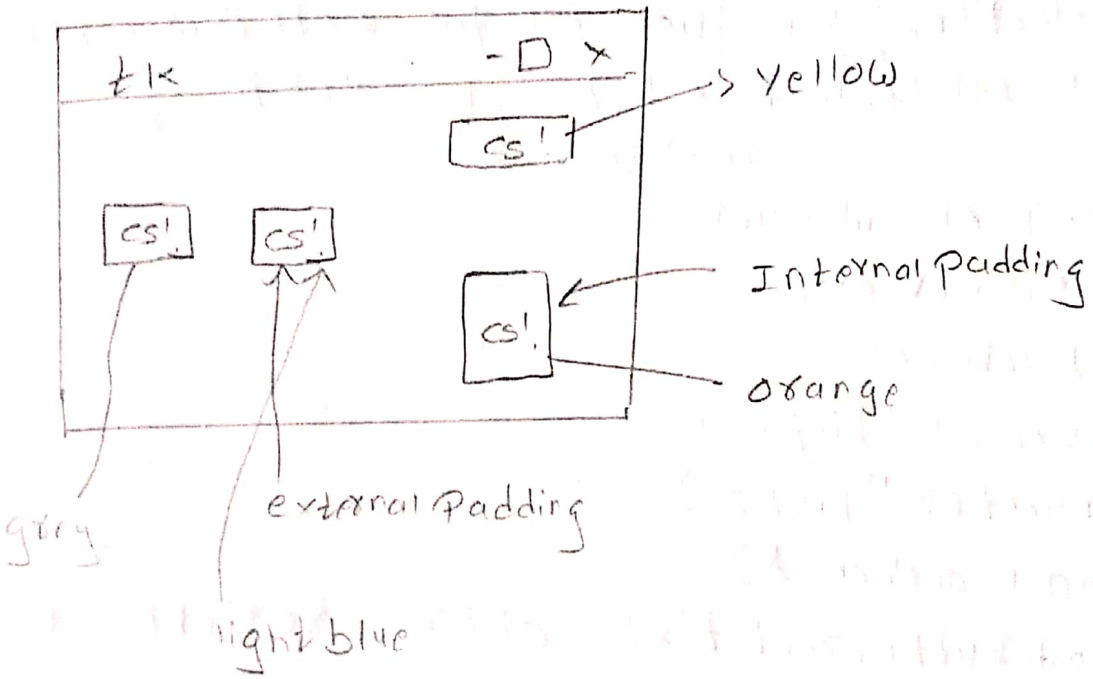
step 3: Use the Pack() along with the object created from the text() and use the Parameter
1) side = LEFT , Padx = 20
2) side = LEFT , Pady = 30
3) side = TOP , iPadx = 40
4) side = TOP , iPady = 50

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 arguments.
1) Name of the Parent window.
2) Text attribute which defines the string.
3) The background color (bg)
4) The foreground fg and than use the Pack() with a relevant Padding attributes

Scanned with CamScanner

```
l4= Label (root, text = "cs!", bg = "orange",
                fg = "black", font = "10")
l4.Pack (side = TOP, ipady = 50)
root.mainloop()
```



Radio button

```
from Tkinter import *
root = Tk()
root.geometry("500x500")
def select():
    selection = "you just selected" + str(var.get())
    l1 = Label(root, text = selection, bg = "white",
               fg = "green")
    l1.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", variable = var,
    value = "Option1", 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

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 mode from multiple option available.

step 4: Now define the parentwindow 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.

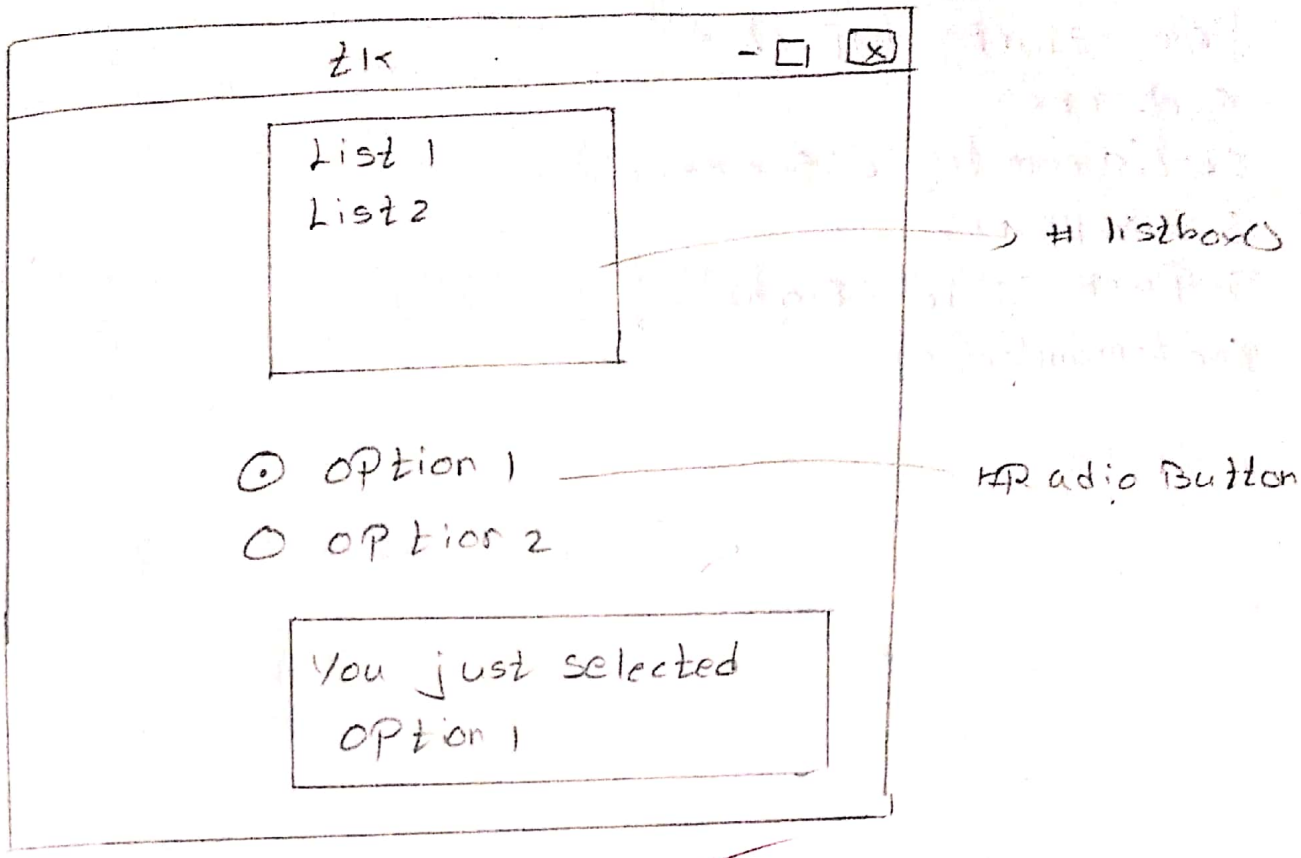
step 6: Create an object from radio button which will take following arguements parent window object, text variable which will take the values option no 1, 2, 3 .... variable arguement, corresponding value and trigger the function declared.

Scanned with CamScanner

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

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

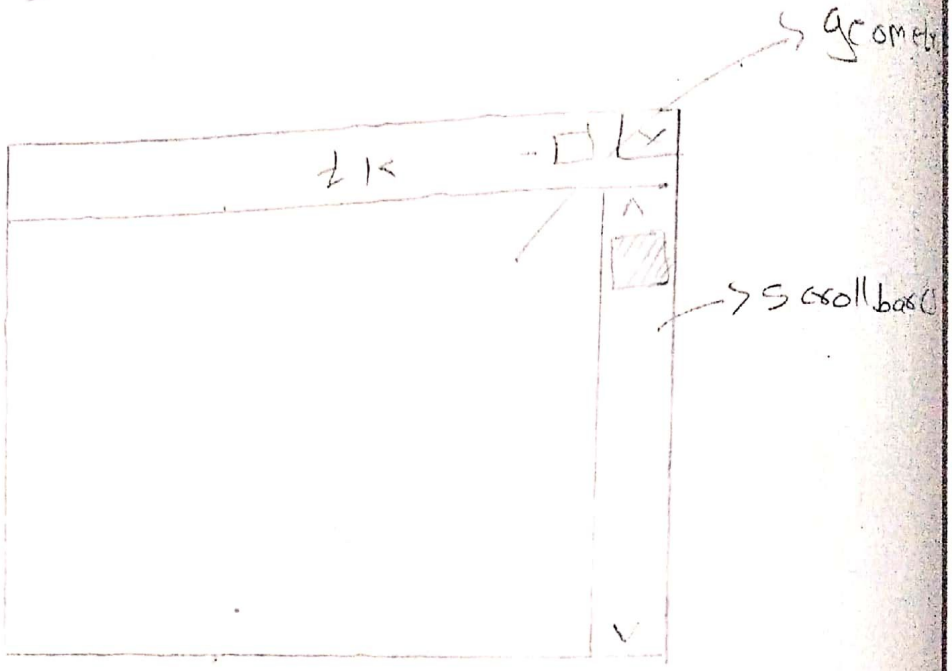
Scanned with CamScanner


tk
List 1
List 2
listbox()
Option 1
Option 2
Radio Button
You just selected
Option 1

1. Scrollbar

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

Output.



2:

```
from tkinter import *
window= TK()
window.geometry ("680x500")
frame= Frame (window)
frame.Pack()

leftframe= Frame (window)
leftframe.Pack (side= "LEFT")

rightframe= Frame (window)
rightframe.Pack (side= "RIGHT")
```

Scanned with CamScanner

GUI Component (A)

#1: Scrollbar.

Step1: Import relevant methods from the tkinter library.

Step2: Create a parent window object corresponding to the parent window.

Step3: Use the geometry () for laying of the window.

Step4: Create an object and use the scrollbar().

Step5: Use the pack () along with scrollbar object with side and fill attribute

Step6: Use the mainloop with the parent object.

#2:

Step1: Import relevant methods from tkinter library.

Step2: Define the object corresponding to parent window and define the size of parent window in terms of number of pixels

Scanned with CamScanner

step 3: Now define the frame object from the method and place it on to the parent window.

step 4: Create another frame object termed as the left frame and put it on the parent window on its LEFT side.

step 5: Similarly define the RIGHT frame and subsequently define the button object placed onto the given frame with the attribute as text, active background and foreground.

step 6: Now use the pack() along with the side attribute.

step 7: Now- similary create the button object corresponding to the MODIFY operation put it into frame object on side = "right".

step 8: Create another button object and place it on to the RIGHT frame and label the button as ADD.

step 9: Add another button and puts it on the top of frame and label it as Exit.

step 10: Use the pack() simulataneously for all the objects and finally use the mainloop()

Scanned with CamScanner

b1 = Button (frame, text = "select", active background = "red", fg = "black")

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

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

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

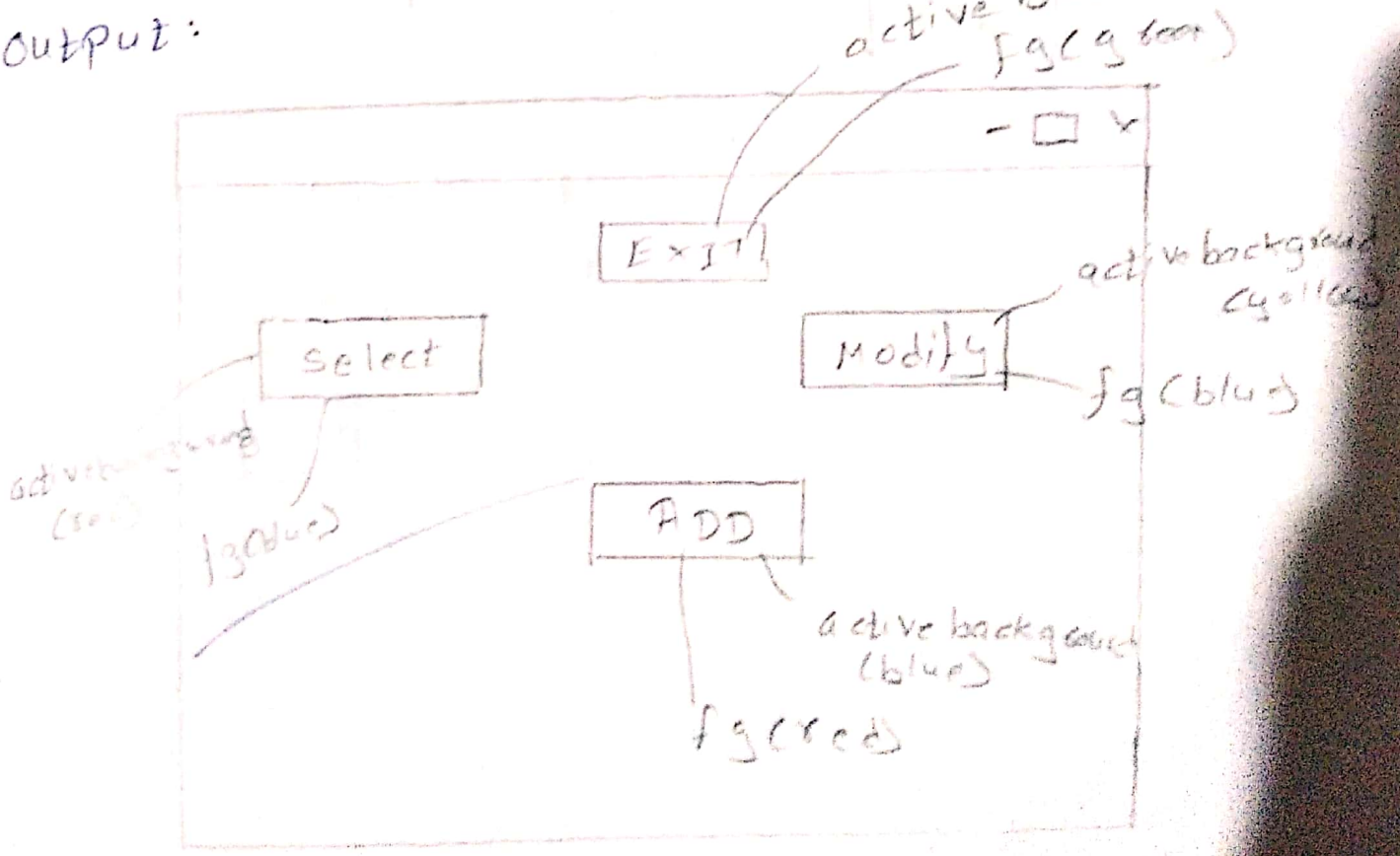
b1.Pack (side = "LEFT", padx = 20)

b2.Pack (side = "RIGHT", padx = 30)

b3.Pack (side = "bottom", pady = 20)

b4.Pack (side = "TOP")

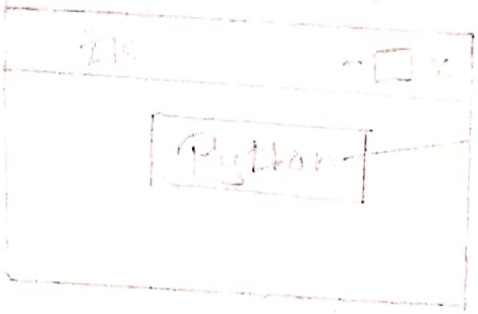
Output:



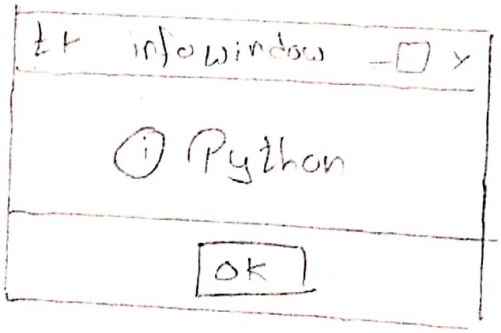
Message box:

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

Output:



click then this wind is pop up



Scanned with CamScanner

3: Message box

step 1: Import the relevant methods from tkinter library.

step 2: Import tkMessagebox.

step 3: Define a Parentwindow object along with the parent window..

step 4: Define a function which will use tkMessage box 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 and finally call mainloop() for trigerring of the events called above.

Scanned with CamScanner

GUI components (c)

Step 1: Import the relevant methods from the tkinter library along with Parent window object declared.

Step 2: Use Parentwindow object along with minsize function for window size.

Step 3: Define a function main, declare parent window object and use config() title(), minsize(), Label() as well as button() and use Pack() and mainloop() simultaneously.

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

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

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

Scanned with CamScanner

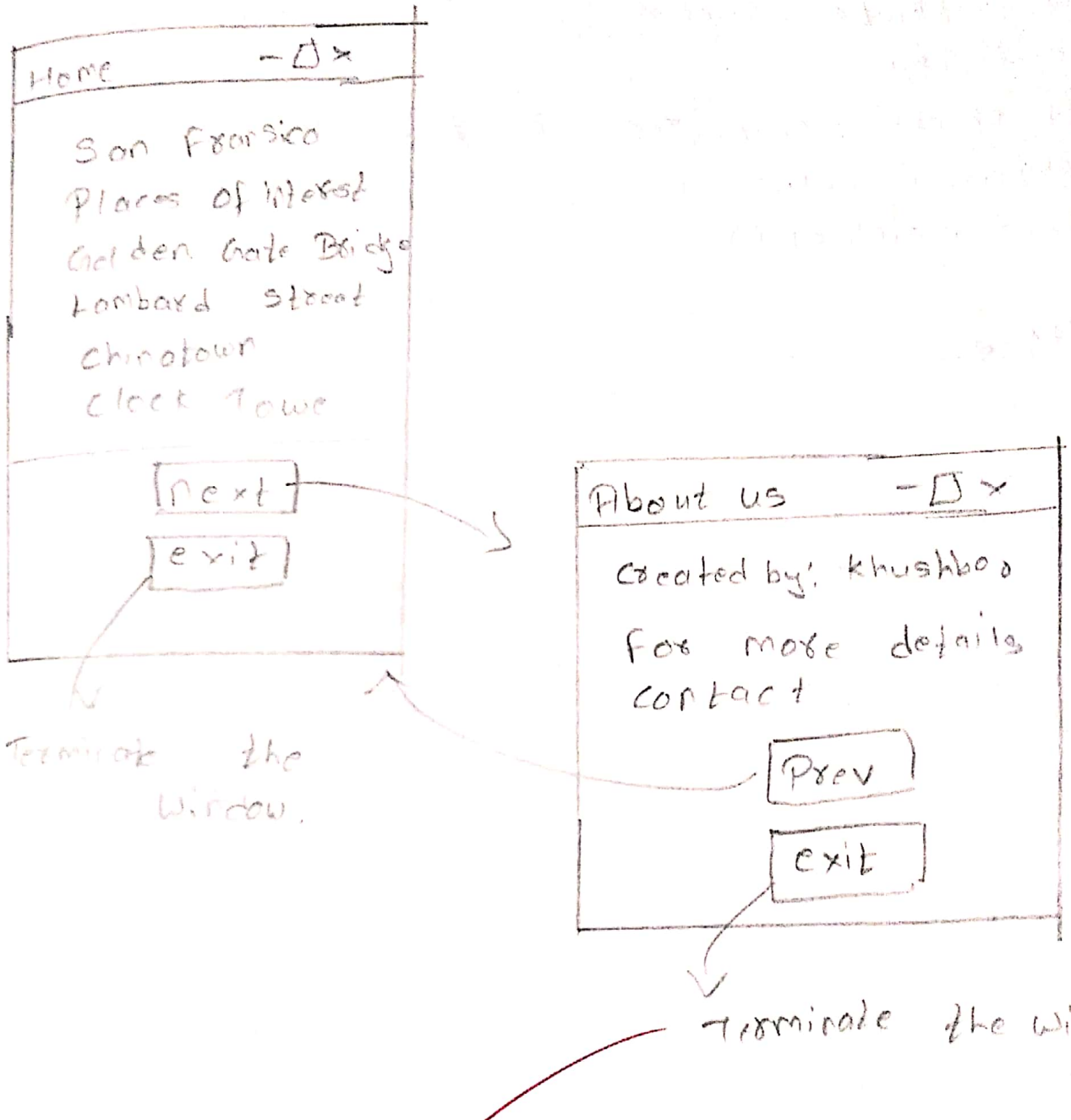
Multiple window
Different button (Relief cs)


```
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 \n Places
            of Interest \n Golden Gate Bridge
            \n Lombard street \n chinatown \n
            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()
def second():
    top2 = Tk()
    top2.config(bg="orange")
    top2.title("About us!!")
    top2.minsize(300,300)
    L = Label(top2, text="Created by: Khush \n for
    more details contact")
    L.Pack()
    b3 = Button(top2, text="Prev", command = main)
    b3.Pack(side = LEFT
```

Scanned with CamScanner

```
b2.= Button (top2, text = "exit", command= terminate)
b2 Pack (side = RIGHT)
top2. mainloop()

def button():
    top3 = Tk()
    top3. geometry ("300 x 300")
    b1= Button( top3, text = "Flat button", relief= FLAT)
    b1. Pack()
    b2 = Button (top3, text = "raised Button", relief =
        RAISED)
    b2. Pack()
    b3 = Button (top3, text = "groove button", relief =
        GROOVE)
    b3. Pack()
    b4= Button (top3, text = "SUNKEN button", relief=
        SUNKEN)
    b4. Pack()
    b5 = Button (top3, text = "ridge button",
        relief = RIDGE)
    top3. mainloop()
def terminate():
    quit()
b5 = Button (root, text = "Your Details", command= main)
b5. Pack()
b6 = Button (root, text = "Button Details", command=
        button)
b6. Pack()
root. mainloop()
```


Home
San Francisco
Places of interest
Golden Gate Bridge
Lombard street
Chinatown
Clock Towe
next
exit
Terminate the window.
About us
Created by: khushboo
For more details contact
Prev
exit
Terminate the window

17/02

Source code:

```
from tkinter import *
root = TK()
s1 = Spinbox(root, from_ = 0, to_ = 10)
s1.pack(anchor = 6)
root.mainloop()
```

Output:

Scanned with CamScanner

spinbox

step1: Create an object from the tk methond and subsequently create an object from the spinbox method.

step2: Make the object so created onto the parent window and trigger the corresponding events.

step3: Use the pack method to provide direction using anchor method.

step4: Use the mainloop to terminate.

Scanned with CamScanner

Paned window.

step 1: Create an Object from Paned window and use the Pack method with the attribute fill and expand

step 2: Create an object from the label method and Put it onto the Paned window with the text attribute and use the add method to Embed the new object

step 3: Similarly create a second Paned window object and add it onto the 1st Paned window with Orientation specified.

step 4: Now create another label object and Place it onto the 2nd Paned window object and add to onto the 2nd Pane.

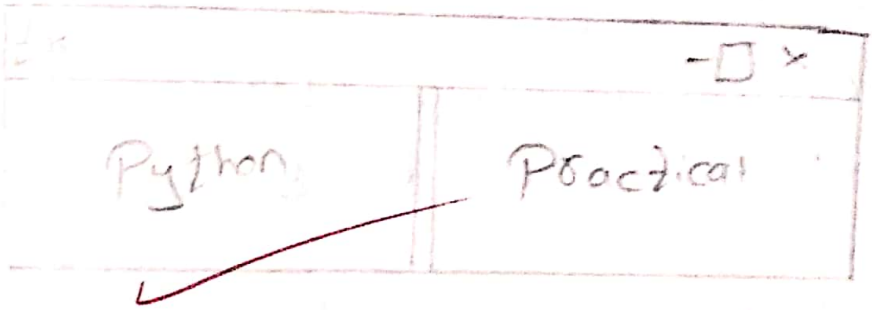
step 5: Now use the mainloop

Scanned with CamScanner

Source code:


```
from tkinter import *
root = Tk()
p = Panedwindow(bg = "blue")
p.pack(fill = BOTH, expand = 10)
l1 = Label(P, text = "Python", bg = "red")
p.add(l1)
p1 = Panedwindow(P, orient = VERTICAL, bg = "yellow")
p.add(P1)
l2 = Label(P1, text = "khushboo", bg = "blue")
P1.add(l2)
root.mainloop()
```

Output:



```
from tkinter import *
root=Tk()
root.config(bg="grey")
def finish():
    messagebox.askokcancel("Warning", This will
     end the Program")
    quit()
def info():
    list1= Listbox()
    list1.insert(1, "Co. Name : Apple")
    list1.insert(2, " Products: iPhone")
    list1.insert(3, " language : Swift")
    list1.insert(4, " OS : IOS")
    list1.grid(ipadx=30)
def aboutus():
    list2= Label(text= "About us")
    list2.grid(ipadx=30)
    list3= Label(text= stew jobs Theatre March
                 2020")
    list3.grid(ipadx=24)
P1 =PhotoImage(file= " download.gif")
f1 = Frame(root, height = 35, width = 5)
f1.grid(row1, column=0)
f2= Frame(root, height = 250, width = 500)
f2.grid(row1, column =1)
P2= P1.subsample(5,4)
l1 = Label(f1, image= P2, relief = FLAT)
l1.grid(row=1, column=0, Padx= 20, Pady=15)
l2= Label(f2, image= P1, relief = SUNKEN)
l2.grid(Pad x=25, Pady=10)
```

Scanned with CamScanner

Image

step 1: Import relevant methods from the tkinter library.

step 2: Create parent window object and use the config method along with background color attribute specified.

step 3: Define a function finish with the messagebox widget which will display a message and subsequently terminate the program.

step 4: Define a function info. Use a listbox widget along with the object of the same. Use the listbox object along with insert method and insert the same and finally use the grid() with ipady attribute.

step 5: Define a function about us with label widget and text attribute and subsequently use the grid()

step 6: Use PhotoImage widget with file and filename with gif attribute

Scanned with CamScanner

Step 7: Create a frame object along with the Frame() along with parent window object height and width specified and subsequently use the grid() with row and column attribute specified.

Step 8: Similarly create another frame object as declared by step 7.

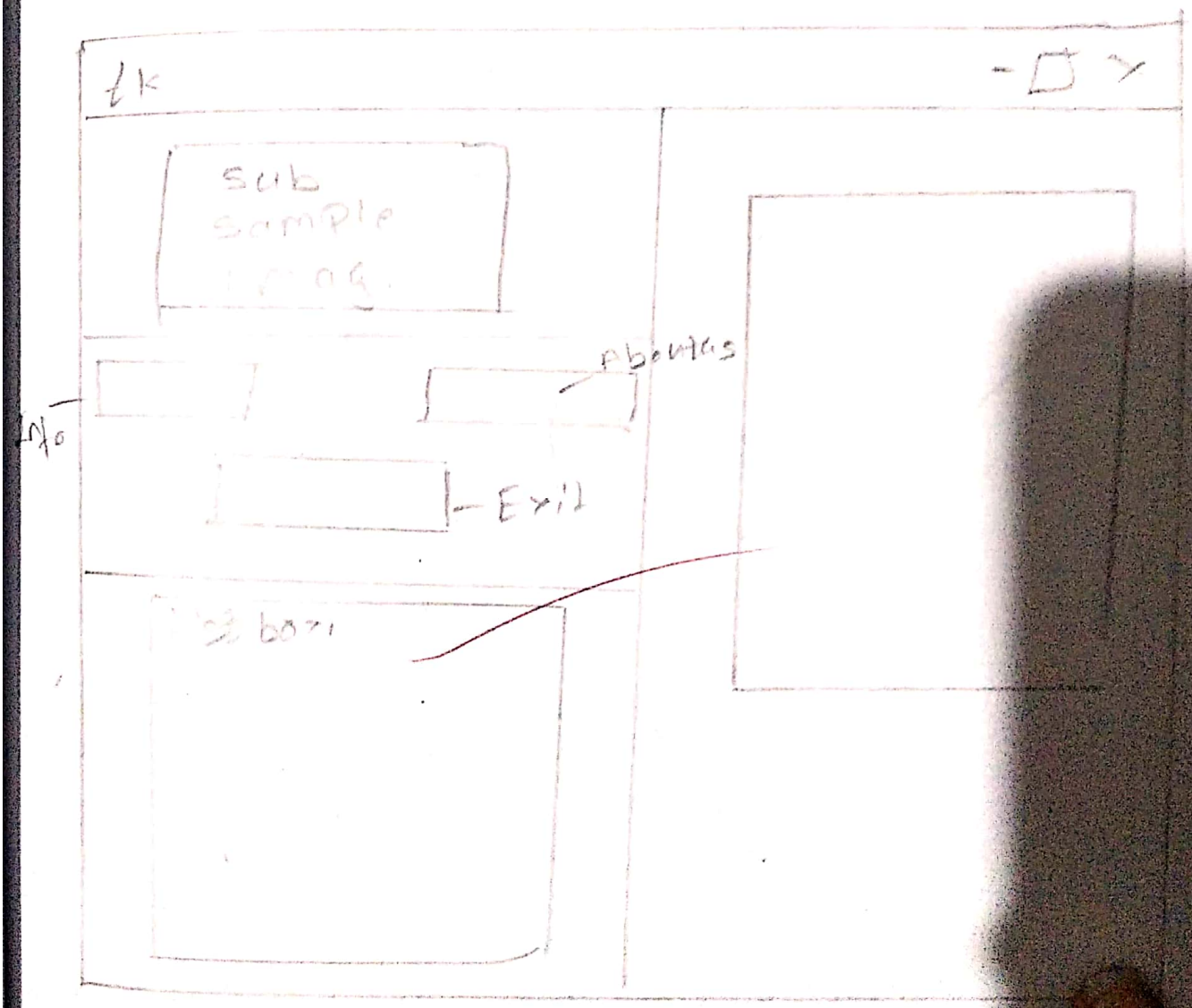
Step 9: Create another object and use the subsample (5/4)

Step 10: Use label widget along with the frame object, relief attribute and subsequently use the grid().

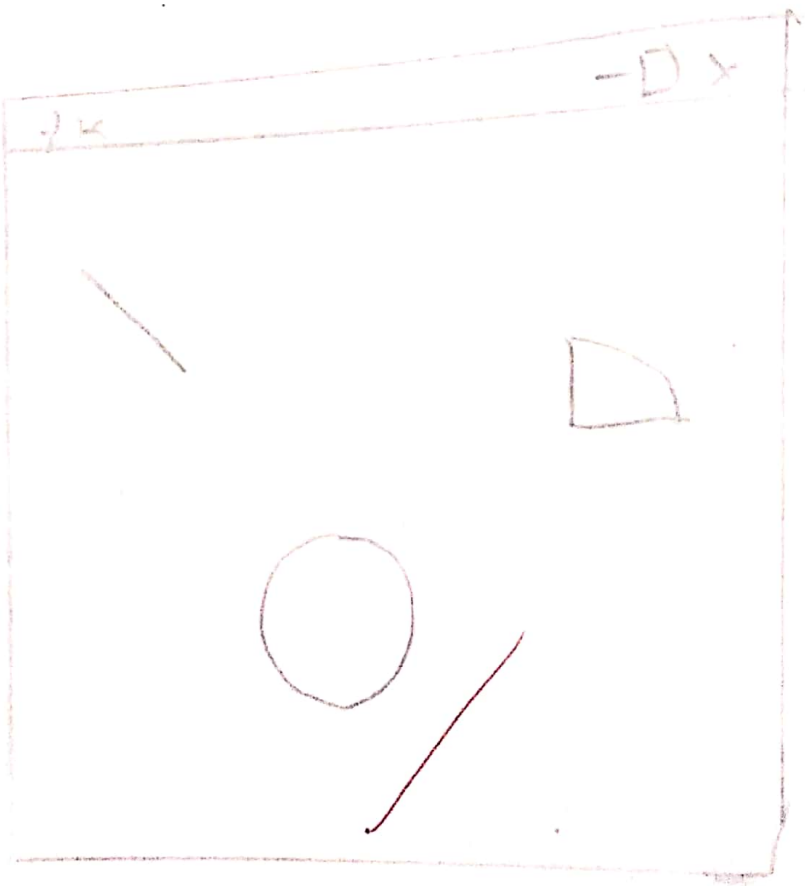
Step 11: Now create button object dealing with different section of frame.

Scanned with CamScanner

```
b1= Button (f1, text = "Information", relief = SUNKEN,
    command = info)
b1.grid (row=1, column=0)
b2= Button (f1, text= About us", relief = SUNKEN,
    Command= About us)
b2.grid (row=1, column=2, Pad2 =5)
b3= Button (f1, text="EXIT", Relief= RAISED,
    Command= finish)
b3.grid (row=2, column=1, ipadx=15)
root.mainloop ()
```



```
from tkinter import*
root=TK()
cl= Canvas(root, height=400, width=400, bg="black")
oval= cl.create_oval(20,140,150,250, fill="white")
line= cl.create_line(30,40,50,60, fill="blue")
arc= cl.create_arc(20,40,60,80, fill="yellow")
cl.pack()
root.mainloop()
```



Canvas widget

step1: Use the tkinter method and create an object from the canvas method and use the attribute height, width, bgcolor 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 co-ordinate value. Also use the fill attribute to assign various colours.

step3: Now call the pack methon and mainloop method.

Scanned with CamScanner

Practical 6

Aim: Database Connectivity

Algorithm:

Step I: Import the library and use the open for creating the database by specifying the name of the database along with the corresponding flag.

Step II: Use the object so for created for accessing the given website and corresponding regular name for the website.

Step III: 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 does not found / unmatched.

Step IV: Use the close() to terminate the database library.

Scanned with CamScanner

```
>>import dbm
db= dbm.open("database","c")
db["name"]= "name"
if db["name"]==None:
    print("database empty")
else:
    print("Match found")
db.close
```

Output:

Match found

Scanned with CamScanner

```
import os, sqlite3
con = sqlite3.connect("student db")
c = con.cursor()
c.execute("CREATE TABLE student(roll INTEGER, name TEXT)")
c.execute("INSERT INTO studodent values
        (1720, 'Khushboo'), (1777, 'Abhishek'),
c.execute("SELECT * from student WHERE
        roll="1777")
print(c.fetchall())
con.commit()
c.close
```

Output:

```
[(1777, 'Abishek')]
```

Scanned with CamScanner

Algorithm:

step I: Import corresponding library to make database connection, os and sqlite3

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

step III: Now use the execute() for creating the table with the column name and respective datatype.

step IV: Now create cursor object using the cursor() and from the connection object created.

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

step VI: Use the commit() to complete the transaction using the connection object.

step VII: Use the execute statement along with cursor object for accessing the values from the database using the select from where clause.

Scanned with CamScanner

Step VIII: Finally use the fetch() or fetchall() and print the value from the table using cursor object.

Step IX: Finally close() the cursor object.

Scanned with CamScanner