



# INDEX



No.	Title	Page No.	Date	Staff Member's Signature
	Python - II			
1	File Handling Using Python	23.	26/11/19	Dr. <u>Jitendra</u>
2.	Iter() and next() Method in Python	26	31/12/19	Dr. <u>Jitendra</u>
3.	Exception Handling	29.	17/12/19	Dr. <u>Jitendra</u>
4.	Demonstrate the use of Regular Expression	33.	24/12/19	Dr. <u>Jitendra</u>
5	Demonstrate the use of text widget (GUI components)	36.	9/1/20	Dr. <u>Minal</u>
6.	Demonstrate the use of Radio- button, geometry and scrollbar	38.	14/1/20	Dr. <u>Minal</u>
7.	Demonstrate the use of MessageBox and quit()	42.	21/1/20	Dr. <u>Minal</u>
8.	Demonstrate the use of Multiple Window	44	28/1/20	Dr. <u>Minal</u>

★ ★ INDEX ★ ★

550

else :

    print (" Can't Perform because of invalid choice")

\* Output :-

Enter the 1st Number 50

Enter the 2nd Number 25

Select the Operation Among the choices

1. Addition
2. Subtraction.
3. Multiplication
4. Division
5. Float Division
6. Modulo Division.

Enter Your Choice No. 3

Multiplication is 1250

~~27/19/19~~

PRACTICAL - I

Aim:- Demonstrate the use of 3 different file accessing modes & different attributes read methods.

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

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

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

ESU

Step IV :- Now open the fileobj in write mode  
Write some another content, close  
subsequently then again open the  
file object in 'wt' mode i.e. is the  
update mode and write contents.

Step V :- Open fileobj in read mode and display  
the update, written content and close.  
Open again in 'rt' mode with parameter  
passed and display all the output simultaneously.

Step VI :- Now open fileobj in append mode.  
Open write Method and write the content  
close the fileobj again open the fileobj  
in read mode and display the 'appending'  
output.

```

fileobj = open ("abc.txt", "w")
fileobj.write ("Hey! This is Vicky Kumar \n")
fileobj.write ("Roll No.: 1729")
fileobj.close()

```

```

fileobj = open ("abc.txt", "r")
print (fileobj.read())
fileobj.close()

```

```

>>> Hey! This is Vicky Kumar
    Roll No.: 1729

```

```

# readline()
fileobj = open ("abc.txt", "r")
print (fileobj.readline())
fileobj.close()

```

```

>>> Hey! This is Vicky Kumar.

```

```

# readlines()
fileobj = open ("abc.txt", "r")
print (fileobj.readlines())
fileobj.close()

```

```

>>> Hey! This is Vicky Kumar
    Roll No.: 1729

```

```
# File Attributes
```

```

a = fileobj.name
b = fileobj.closed
c = fileobj.mode
d = fileobj.softspace

```

```
print (a,b,c,d)
```

```
>>> 'abc.txt', 'True', 'r', '0'
```

# w+ mode.

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

fileobj = open("abc.txt", "r")
print(fileobj.read())
fileobj.close()

>>> Loukik Sir
```

# Append Mode

```
fileobj = open("abc.txt", "a")
fileobj.write("Data structures")
fileobj.close()
```

# tell()

```
fileobj = open("file.txt", "w")
fileobj.write("Vikram")
fileobj.close()

fileobj = open("file.txt", "r")
printf = fileobj.tell()
print("tell():", printf)
fileobj.close()

>>> 'tell():', 7
```

# r+ mode

```
fileobj = open("abc.txt", "r+")
s = fileobj.read()
print(s)
fileobj.close()

>>> Loukik Sir.
```

```
fileobj = open("abc.txt", "r+")
print(fileobj.read())
fileobj.close()
```

```
>>> Loukik Sir Data Structures
```

# seek()

```
fileobj = open("file.txt", "r")
s = fileobj.seek(0, 0)
print("seek(0,0) is:", s)
fileobj.close()
```

```
>>> 'seek(0,0) is:', None
```

Aim :- Demonstrate the use of tell() and seek() Method.

Step I :- Open the fileobj in read mode. Declare a variable and perform fileobject to perform tell method and store the output simultaneously in variable.

Step II :- Use the seek Method with the arguments while opening the file obj in read mode and closing subsequently.

J.R  
3/12/19

## PRACTICAL - II (1)

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 argument 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 argument to the `iter()` method.

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

# Note:- Exception concept is used to abrupt termination of the given program.

## # Source Code :-

Class Odd :

026

```
def __iter__(self):  
    self.num = 1  
    return self  
  
def __next__(self):  
    if self.num <= 100:  
        num = self.num  
        self.num += 2  
        return num  
  
    else:  
        raise StopIteration,
```

```
myObj = odd()  
myIter = iter(myObj)  
for num in myIter:  
    print(num)
```

## # Output :-

1  
3  
5  
7  
9  
11  
13  
15  
17  
19  
21  
23  
25  
27  
29  
31  
33  
35  
37  
39  
41  
43  
45  
47  
49  
51  
53  
55  
57  
59  
61  
63  
65  
67  
69  
71  
73  
75  
77  
79  
81  
83  
85  
87  
89  
91  
93  
95  
97  
99



## # iter() and next() method

```
mytuple = ("Ramesh", "Monu", "Sonu")
myiter = iter(mytuple)
print(next(myiter))
print(next(myiter))
print(next(myiter))

>>> Ramesh
Monu
Sonu.
```

## # for loop

```
mytuple = ("Ram", "Rahim", "Kabir")
for name in mytuple:
    print(name)

>>> Ram
Rahim
Kabir
```

## # Square and Cube

```
def square(x):
```

$$y = x * x$$

```
    return y
```

```
def cube(x):
```

$$z = x * x * x$$

```
    return z
```

~~tuple1 = [square, cube]~~

~~for num in range(3):~~

~~value = list(map(lambda x: x(num), tuple1))~~

~~print(value)~~

>>> [0, 0]

[1, 1]

[4, 8]

[9, 27]

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 and 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 and cube with a parameter which will obtain output of square and cube of the given number.

Step IV :- Call the declared function using function call.

Step V :- Using for conditional statement specifying the range use the list type casting with map method declare a 'lambda' i.e. anonymous function and print the same and display the output.

*Dr  
1012*

Ques:- Demonstrate the use of map method and lambda function and display the factorial of a given number.

Algorithm :-

Step I :- Define a function which will accept one argument.

Step II :- Create an empty list to store the values and by append method store the value in it and print it.

Step III :- Using anonymous function like lambda() and map() method , calculate the factorial of a given number.

Step IV :- Map the calculated value into the list and store it in a variable.

Step V :- Accept the integer value from the user and call the function.

Q Factorial using Lambda() & map() Method.

```
def fact(num):
    mylist = []
    mylist.append(num)
    print(mylist)
    mylist = (lambda num: [1, 0] [num > 1] or mylist(num)*num)
    a = (map, list)
    print("Factorial is", mylist(num))
    num1 = int(input("Enter the Number:"))
    fact(num1)
```

Q Output :-

Enter The Number : 5

[5]

Factorial is 120.

8503

# IOError :-

try:

```
fo = open ("file.txt", "w")
fo.write (" Exception Handling")
fo = open ("file.txt", "r")
str1 = fo.read()
print (str1)
fo.close()
except IOError:
    print (" Error Occured ")
else:
    print (" Operation Successfull ")
```

Output :-

```
>>> Exception Handling
Operation Successfull.
```

# Value IOError :-

try:

```
fo = open ("file.txt", "w")
fo.write (" Hello World")
fo = open ("file.txt", "r")
str2 = fo.readline()
print (str2)
except ValueError:
    print (" Value Error Detected ")
else:
    print (" Operation Successfull ")
```

**Aim:-** Write a Program to demonstrate the use of `ValueError` in given program statements.

**Algorithm:-**

Step I :- Use the try block to define the normal course of action, for e.g., define the file object and open the file in write mode and write some content on to file. Now open the file in read mode and subsequently read the file.

Step II :- Use the except block with I/O block as an environment error and convey the appropriate message that the operation is carried out successfully.

Step III :- Accept the value from the user and if it is a valid value then display the entered value and terminate the condition by using the break statement

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

Step V :- Define Multiple exception using except statement for finding the different category of errors.

## ESO...

Step VI :- Use the try block to define the normal for e.g. accepting two inputs from the user and using print statement print the division of those numbers.

Step VII :- Use the except block and convey appropriate message for the user, else display that the operation is carried out successfully.

Step VIII :- Accept an integer values from the user. In try use the division method.

Step IX :- For the exception to be raised use the except keyword and i.e. TypeError, Print TypeError detected.

Step X :- Use the except with error of zero division Error, and print the message according to that if entered number is zero not able to perform operation.

Step XI :- Declare static variables and values

Step XII :- For Multiline exception use the error type by separating them with a comma

Output:-

>>> Hello World  
Operation Successful

030

# Arithmetic Error

try:

```
a = int(input("Enter 1:"))
b = int(input("Enter 2:"))
print("Division:", a/b)
except: ArithmeticError:
```

else:

```
print("Operation successful")
```

Output:-

>>> Enter 1: 5
Enter 2: 0
Arithmetic Error Detected.

# Type Error

a = 1

b = input("Enter: ")

try:

print(a/b)

except: TypeError:

```
print("Type Error Detected")
```

>>> Enter: t

Type Error Detected.

## Q80. Multiline Exception

```
a,b = 1, 0  
try:  
    print (1/0)  
    print ('10/10')  
except (TypeError, ZeroDivisionError):  
    print (" Invalid Input Error")  
  
=>> 1/0  
=>> 10/10  
• Invalid Input Error
```

## Q81. Using Except Keyword

```
try:  
    a = open("file.txt", "w")  
    a.write ("Hello Python")  
except IOError:  
    print ("Error Detected")  
else:  
    print ("Operation Successful")  
  
def x():  
    l = [ ]  
    print (len(l))  
  
def y():  
    li = [2,4,4,1]  
    print (len(li))  
  
print (x())  
print (y())
```

- Step XIII :- Use the try block open a file in write mode and subsequently enter values in the file.
- Step XIV :- Use the file error and display message. and
- Step XV :- Define a function with empty list and calculate the length of the list
- Step XVI :- Define another function y initialise or declare some elements in list and calculate the length of same and display.
- Step XVII :- In try block exception input from the user and if user enter character value raise an error that is saying - Up Enter Integer Value!

Output:

032

```
>>> Operation Successful  
0  
None  
4  
None
```

# If Raise keyword.

```
try:  
    a = int(input("Enter a :"))  
    raise ValueError:  
except ValueError:  
    print(" Enter Integer Value")
```

```
>>> Enter a Number: xyz
```

Enter Integer Value.

Jan 2011

~~SEO~~  
#1 match()

```
import re  
pattern = r"EYES"  
sequence = "EYES represents Computer Science Stream"  
if re.match(pattern, sequence):  
    print("Matched Pattern Found")  
else:  
    print("Pattern NOT FOUND!")
```

>>> Matched Pattern 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' \st'  
replace = ''  
print(re.sub(pattern, replace, string))  
>>> abcdefghi
```

## Practical - IV

Aim:- Demonstrate the use of Regular Expression

Step I :- Import re module declare pattern and sequence.

use match method with declare arguments.

if arguments matched then print the same  
otherwise print pattern NOT FOUND.

Step II :- Import re module declare pattern and with

and literals and meta characters. Declare strings

value and use the.findall() method with  
arguments 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 arguments  
and print the string without  
spaces.

- Step **V** :- Import re module, declare a sequence. Use search method for finding, and subsequently use the group() with dot operator or search() given 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 discussed up the for condition statement.
  - Step **iii**] Use if conditional for checking 1<sup>st</sup> no. 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 number Matched otherwise print Match failed.
- Step **VII** :- Import re module and declare a variable named str and store some string value in it. With the help of.findall() method and string and display the same on window.

```
#5 group()
import re
string = "Python is an Interpreted language"
v = re.search('Python', string)
print(v)
print(v.group())
>>> <sre.SRE_Match object at 0x02810F00>
Python.
```

034

#6 Verifying the given set of Cell No.

```
import re
list1 = ['8009126624', '9990098690', '8008230693']
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
str = 'Hello Python is a Programming language'
output = re.findall(r'\b[aeiouAEIOU]\w+', str)
print(output)
>>> ['is', 'a']
```

# host domain.

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

>>> ['abc123.tscs', 'edureka.com', 'fyos', 'edu.com']
```

# Count of first 2 letters

```
import re
s = 'mr.a, ms.b, ms.c, mr.t'
p = r'[^ms][mr ]+'
o = re.findall(p, s)
print(o)
m = 0
f = 0
for v in o:
    if (v == 'mr'):
        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]
```

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 , print 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 mere increment value of f . Hence display the same .

*Jin  
Jill*

Practical -5

Aim:- Demonstrate the use of Tkinter ( GUI Components )

## Q) Algorithms:-

Step I :- Use the Tkinter libraries for importing the features of the root widget.

Step II :- Create an object using TK() method.

Step III :- Create a variable from the text() method and position it on to the parent window.

Step IV :- Use the pack() method along with label method and use side = RIGHT attribute yet

Step V :- Use the mainloop() method for triggering of the corresponding event

## Q) Algorithms:-

Step I :- Use the Tkinter libraries for importing the features of the root widget

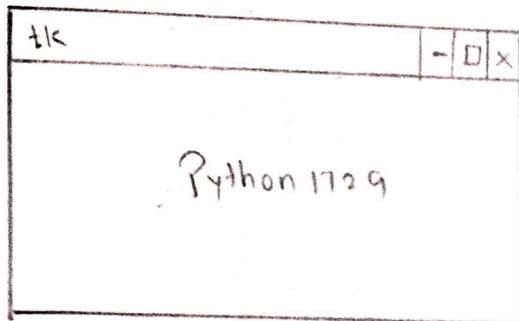
Step II :- Create a variable from the root method and position it on the parent window

```

from tkinter import *
root = Tk()
l = Label(root, text = 'Python 1729')
l.pack()
root.mainloop()

>>> Output:

```

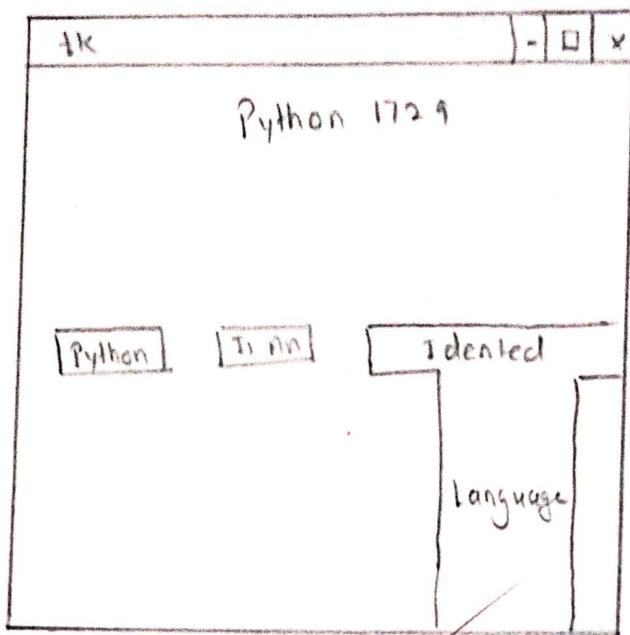


```

from tkinter import *
root = Tk()
l = Label (root, text = " Python 1729")
l.pack()
l1 = Label (root, text = " Python", bg = "blue", fg = "white",
            font = "10")
l1.pack(side = LEFT, padx = "20")
l2 = Label (root, text = " Is An", bg = "white", fg = "white",
            font = "20")
l2.pack(side = LEFT, pady = "20")
l3 = Label (root, text = " Indented", bg = "white", font = "30")
l3.pack(side = TOP, ipadx = "40")
l4 = Label (root, text = " Language", font = "40")
l4.pack(side = TOP, ipady = "50")
root.mainloop()

```

>>> Output



Step III : 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 IV : Use the mainloop() for triggering of the corresponding events

Step V : Now repeat the above steps with the label()

which takes the following arguments

- (1) Name of the parent window
- (2) Text attribute which takes input as string
- (3) The background and the foreground colour termed as bg and fg respectively.
- (4) Use the pack method accordingly.

*Jr. M.L.A*

PRACTICAL-4(B)

Aim:- Demonstrate the use of Radiobutton and geometry() (GUI Components)

#1 Radiobutton.

Step I :- Import the relevant method from the Tkinter libraries.  
Create an object with the parent window.

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

Step III :- Now define a function which tells the user allow the  
given selection made from multiple options available.

Step IV :- Now define the parent window and define the option  
with central variable.

Step V :- Use the listbox() and insert options on the parent  
window along with the pack() with specifying  
anchor attribute.

Step VI :- Create an object from Radiobutton which will take  
following arguments (parent window object, text variable  
which will take the values option 1, 2, 3 , variable  
argument, corresponding value and trigger the function  
declared,

Step VII :- Now call the pack() for Radiobutton object so created.  
and specify the argument using anchor attribute.

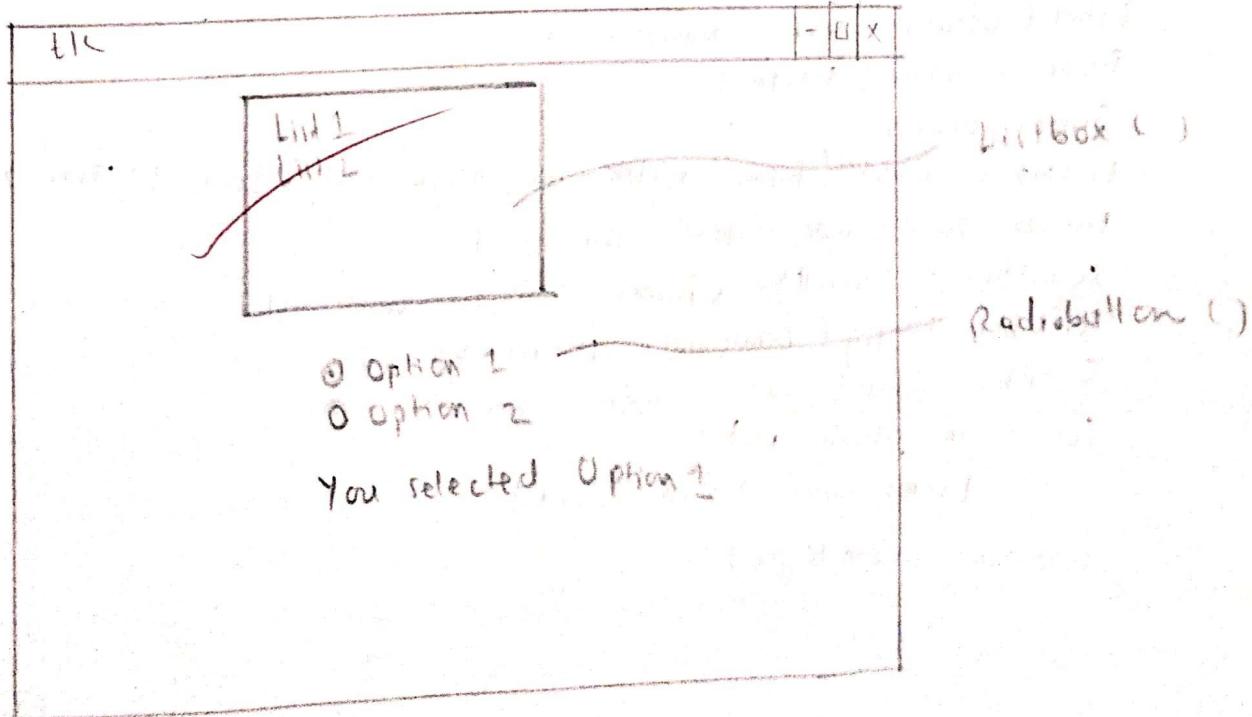
## #1 Radiobutton.

038

```
from tkinter import *
root = Tk()
root.geometry("500x500")
def select():
    selection = "You selected " + str(var.get())
    l1 = Label(text=selection, bg="white", fg="green")
    l1.pack(side=TOP)

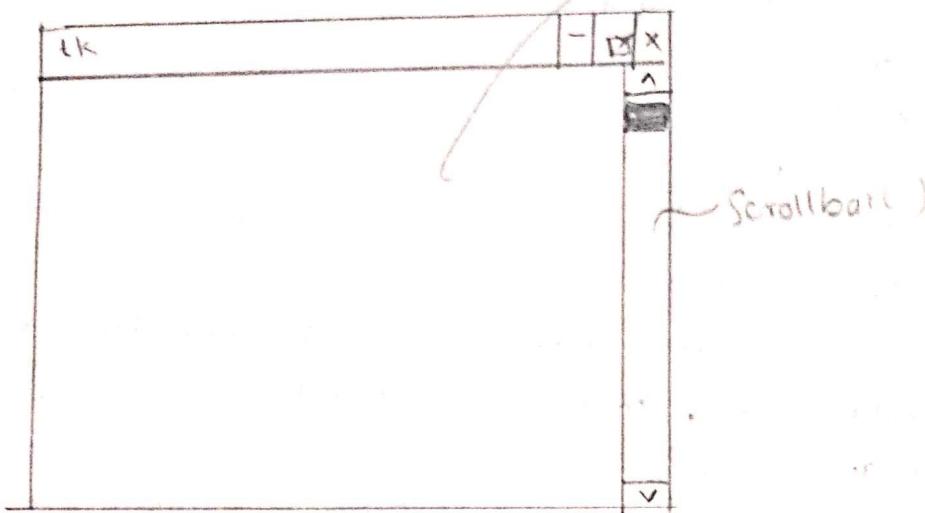
var = StringVar()
l1 = Listbox()
l1.insert(1, "List 1")
l1.insert(2, "List 2")
l1.pack(anchor=N)
r1 = Radiobutton(root, text="Option 1", variable=var,
                  value=1, command=select)
r1.pack(anchor=N)
r2 = Radiobutton(root, text="Option 2", variable=var,
                  value=2, command=select)
r2.pack(anchor=N)
root.mainloop()
```

>>>



# 2.

860  
from tkinter import \*  
root = Tk()  
root.geometry("500x500")  
s = scrollbar()  
s.pack(side="Right", fill="Y")  
root.mainloop()



# 3. Using frame Widget

from tkinter import \*  
window = Tk()  
window.geometry("680x500")  
Label(window, text="Numbers:").pack()  
frame = Frame(window)  
frame.pack()  
listNo = listBox(frame, width=20, height=20, font=("Times New Roman"))  
listNo.pack(side="Left", fill="Y")  
scrollbar = Scrollbar(frame, orient="Vertical")  
scrollbar.config(command=listNo.yview)  
scrollbar.pack(side="Right", fill="Y")  
for x in range(10):  
 listNo.insert(END, str(x))  
window.mainloop()

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

#2.

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

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

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

Step IV :- Create an object and use the scrollbar()

Step V :- Use the pack() along with the scrollbar() object with side and fill attributes

Step VI :- Use the mainloop() with the parent object

#3. Using frame widget

Step I :- Import the relevant libraries from the tkinter method.

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

Step III :- Use the geometry manager with pixel size (680x100) or any other suitable pixel value

Step IV :- Use the label widget along with the parent object created and subsequently use the pack()

## 880

Step IV:- Use the frame widget along with the parent object created and use the pack()

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

Step VI:- Use the scrollbar() with an object and use the attributes of Vertical. then configure the same with object created from Scrollbar () and use pack()

Step VII :- Trigger the event using function mainloop().

### # 4

Step I:- Import relevant methods from tkinter library.

Step II:- Define the object corresponding to parent window and define the size of parent window in terms of no. of pixels.

Step III:- Now define the frame object from the method and place it on to the parent window.

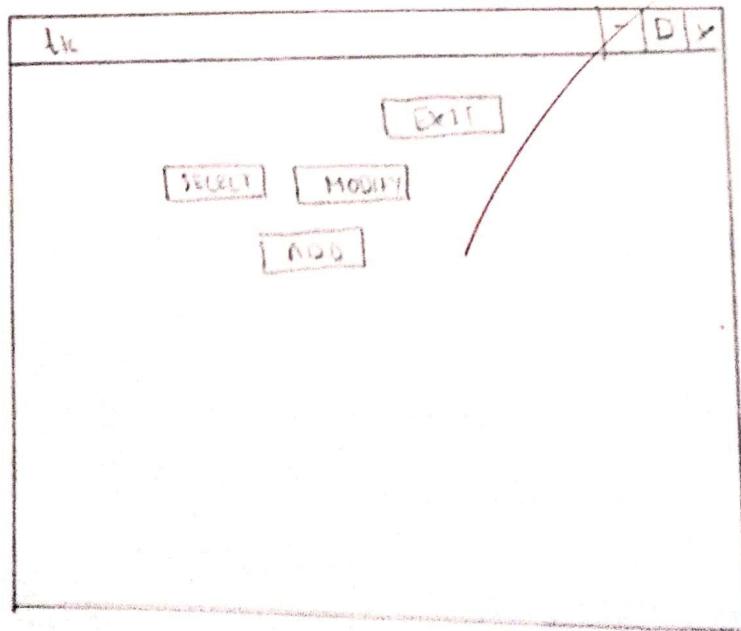
Step IV:- Create another frame object termed as the left frame and put it on the parent window on its LEFT side.

Step V:- Similarly, define the RIGHT frame and subsequently define the button object placed onto the given frame with the attributes as text, activebackground and foreground.

Number	1	2	3	4	5	6	7	8	9	10
Number	1	2	3	4	5	6	7	8	9	10

#4:-

```
from tkinter import *
root = Tk()
root.geometry("680x500")
frame = Frame(root)
frame.pack()
lframe = Frame(root)
lframe.pack(side = "Left")
rframe = Frame(root)
rframe.pack(side = "Right")
b1 = Button(frame, text = "SELECT", activebackground = "white", fg = "white")
b2 = Button(frame, text = "MODIFY", bg = "white", fg = "blue")
b3 = Button(frame, text = "ADD", bg = "white", fg = "blue")
b4 = Button(frame, text = "EXIT", bg = "white", fg = "blue")
b1.pack(side = "LEFT", padx = 20)
b2.pack(side = "RIGHT", padx = 30)
b3.pack(side = "BOTTOM", pady = 20)
b4.pack(side = "TOP")
root.mainloop()
```



Step VI :- Now use the pack() along with the side attribute.

Step VII :- Similarly, create the button object corresponding to the MODIFY operation and put it on to the frame object on side = "Right".

Step VIII :- Create another button object and place it onto the right frame and label the button as ADD.

Step IX :- Add another button and put it on to the top of frame and label it as EXIT.

Step X :- Use the pack() simultaneously for all the objects and finally use the mainloop().

Dr. V. V. S. N. Iyer  
Date: 21/11/2023

Aim:- Demonstrate the use of `MsgBox` with the help of `Tkinter` library.

Step I:- Import the relevant methods from `Tkinter` library.

Step II:- Import `tkMessageBox` (if Python 2.x only)

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

Step IV:- Define a function which will use `tkMessageBox` with `showinfo` method along with the info window attribute.

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

Step VI:- Declare the button widget onto the parent window and finally call `mainloop()` for triggering of the event.

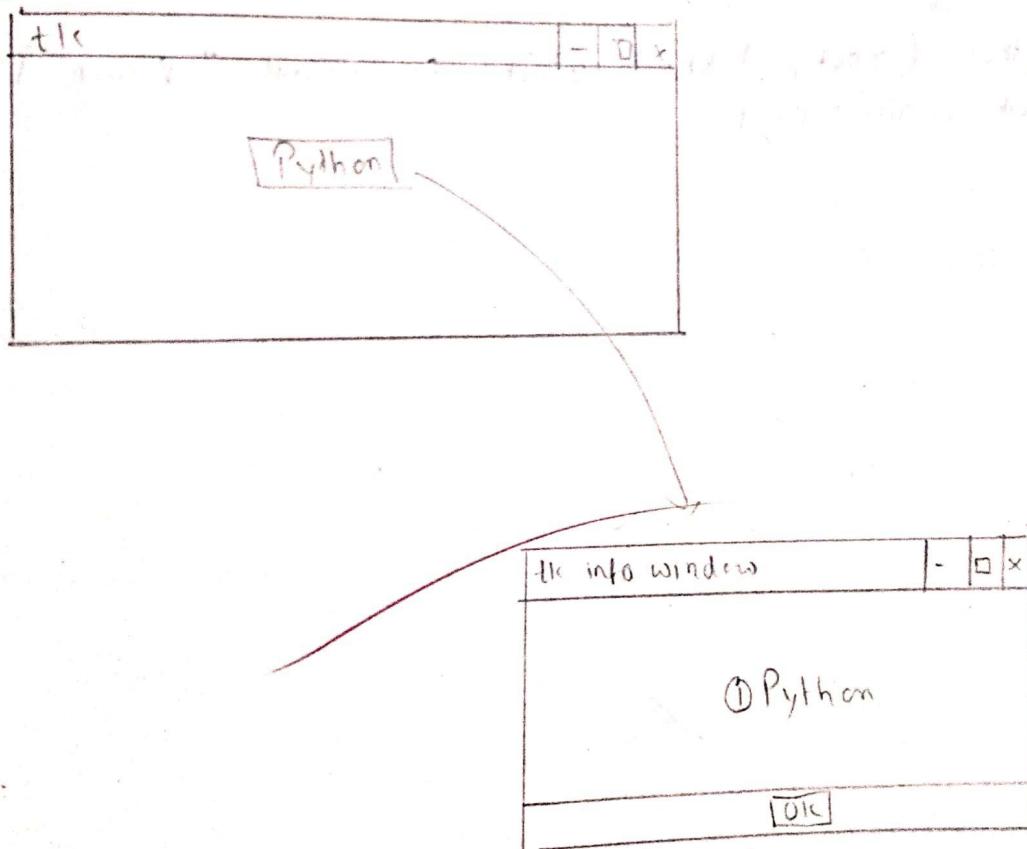
# messageBox .

042

043

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

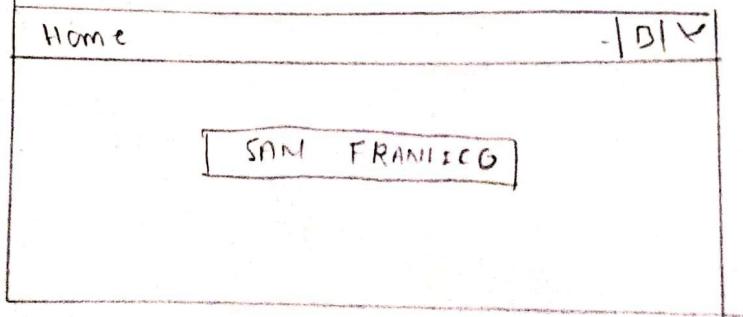
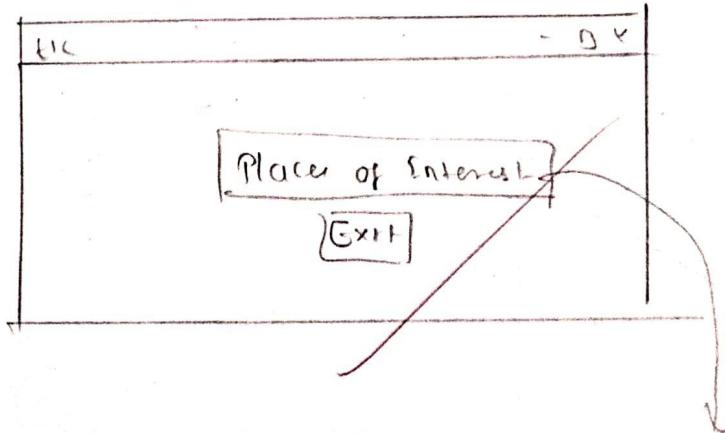
\* Output



840

840

```
from tkinter import *
root = Tk()
root.minsize(300, 200)
def main():
    top = Tk()
    top.title("Home")
    L = Label(top, text="SAN FRANCISCO")
    L.pack()
    Button(root, text="Place of Interest", command=main).pack()
    def finish():
        quit()
    Button(root, text="Exit", command="finish")
    root.mainloop()
```



Step I :- Import the relevant method from the tkinter library.

Step II :- Define a main function which will show the place of interest of user.

Step III :- Define a window object along with the main function.

Step IV :- Define a function along with parent window object to finish the parent window.

Step V :- Declare the button widget over the parent window and finally call the mainloop()

Dr

215

Aim:- Demonstrate the use of Multiple windows

Step I :- Import the relevant method from the tkinter libraries.

Step II :- Create an object with the parent window.

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

Step IV :- Now create a label object and put it on to the parent window. Similarly, create a label object and pack it to left by using anchor attribute.

Step V :- Now define a function which allow to make two another window according to the given selection made from multiple options available.

Step VI :- Under show() (user defined function) check whether the value of button is 1 or not and accordingly create 2 windows in which 1 will be Technical and other will be Non-Technical.

Step VII :- Use Radiobutton() to display the books in both the windows respectively.

Step VIII :- Create a button in both the windows in which will go to Home window i.e. parent window onclick,

## # Multiple Window (Traversing)

044

```
from tkinter import *
root = Tk()
root.title("Home")
root.geometry("200x180")
Label(root, text = "Library", relief = RAISED, bg = "light blue").pack()
Label(root, text = "Type of Book").pack(anchor = W)
def show():
    rec = var.get()
    if rec == 1:
        r1 = Tk()
        r1.title("Technical")
        r1.geometry("200x180")
        s = Scrollbar(r1).pack(side = RIGHT, fill = "y")
        Tech = [ ("Linux", "Linux"), ("Algorithm", "Algorithm"),
                 ("Data Structure", "Data structure"), ("Python", "Python"),
                 ("Java", "Java"), ("Ethical Hacking", "Ethical Hacking")]
        book = StringVar()
        for text, val in Tech:
            Radiobutton(r1, text = text, value = val, variable = book,
                        pack(anchor = W))
        Button(r1, text = "Go To Home", command = lambda:
               r1.withdraw()).pack(anchor = W)
    else:
        r2 = Tk()
        r2.title("Non-Technical")
        r2.geometry("200x180")
        s = Scrollbar(r2).pack(side = RIGHT, fill = "y")
```

Q10

```
Non-Tech = [ ("calculus", "Calculus"), ("stats", "Stats"),  
             ("English", "English"), ("Hindi", "Hindi") ]
```

```
book = StringVar()
```

```
for text, val in Non-Tech:
```

```
    Radiobutton(r2, text = "Go Back To Home",  
                command = lambda: r.withdraw()), pack(anchor = w)
```

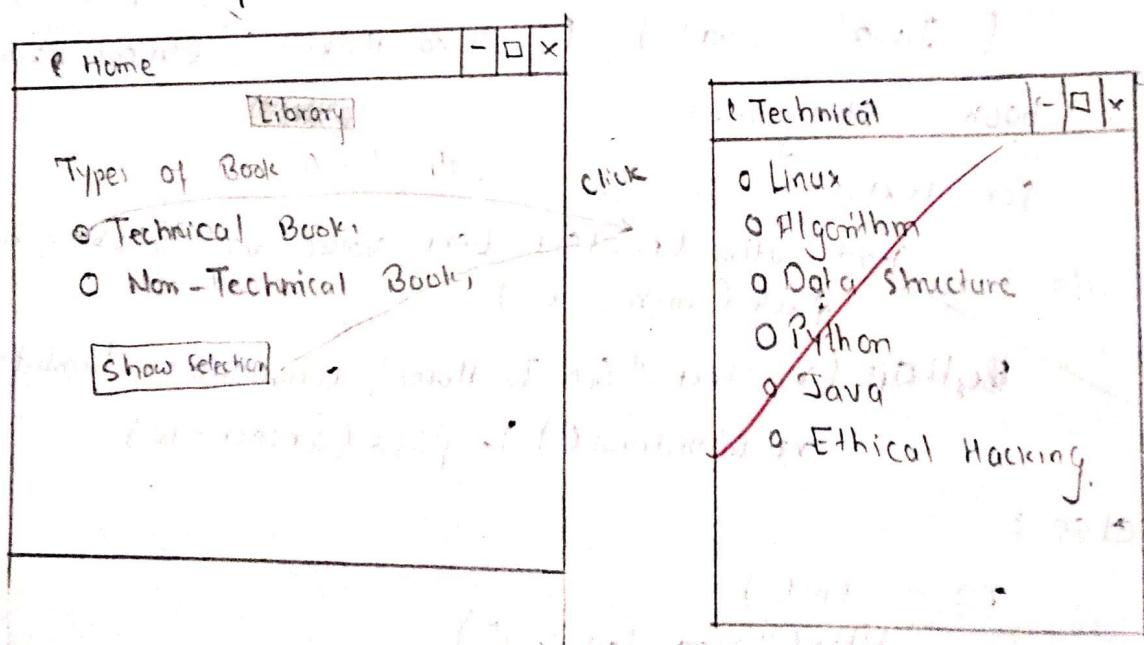
```
var = IntVar()
```

```
r0 = Radiobutton(root, text = "Technical Books", variable = var, value = 1)  
r0.pack(anchor = w)
```

```
r1 = Radiobutton(root, text = "Non-Technical Books", variable = var, value = 2)  
r1.pack(anchor = w)
```

```
b = Button(root, text = "Show Selection", command = show)  
b.pack(anchor = w)
```

```
mainloop()
```



Step IX :- Now, create the two Radiobuttons in which show the Types of Books

Step X :- Create a Button and on click of button call the show() which is a user defined function

Step XI :- Finally call the mainloop for triggering of the events.

Final

Ques:- Demonstrate the use of Paned Window and SpinBox.

### #Paned Window

Step I :- Create an object from PanedWindow and use the pack() with the attribute fill and expand.

Step II :- Create an object from the Label() and put it onto the PanedWindow with the text attribute and use the add() to embed the new object.

Step III :- Similarly, create a second PanedWindow object and add it on to the first PanedWindow with orientation specified.

Step IV :- Now, create another Label object and place it onto the second PanedWindow object and add it onto the second PanedWindow created so.

Step V :- Now use the mainloop() to trigger the events.

## # Paned Window ()

046

```
from tkinter import *
```

```
root = Tk()
```

```
P = PanedWindow(root, orient = VERTICAL)
```

```
P.pack(fill = BOTH, expand = 10)
```

```
L1 = Label(P, text = " Hey Here")
```

```
P.add(L1)
```

```
P1 = PanedWindow(P, orient = HORIZONTAL), pack(fill = BOTH, expand = 1)
```

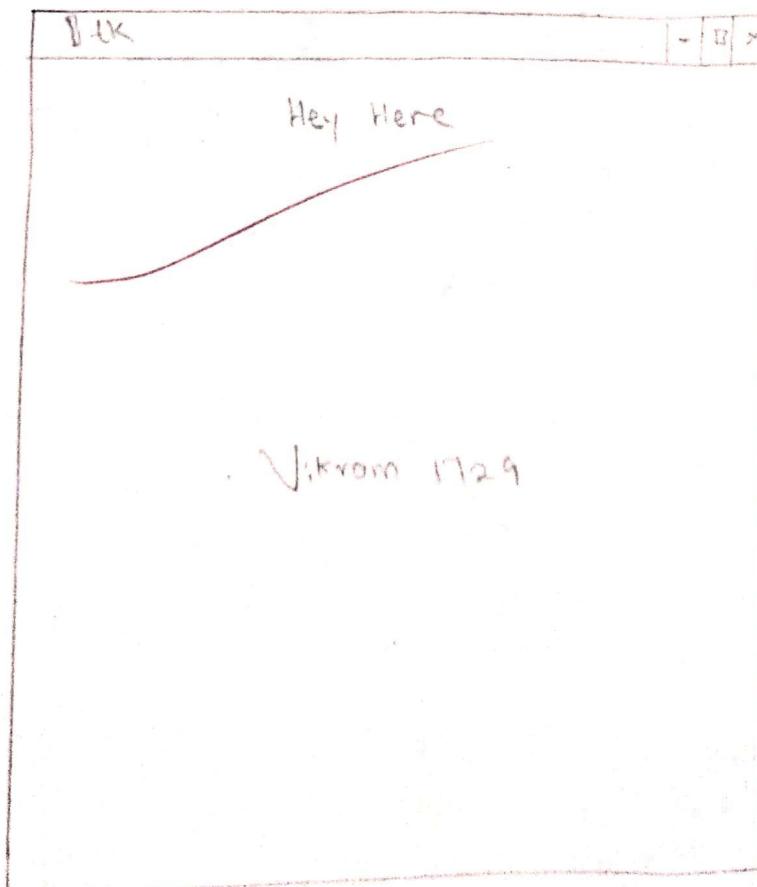
```
P.add(P1)
```

```
L2 = Label(P1, text = " Vikram 1729")
```

```
P1.add(L2)
```

```
mainloop()
```

## # Output

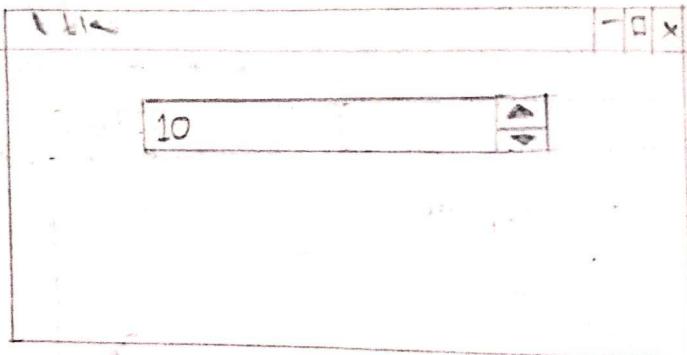


840

### # SpinBox

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

### # Output



## # Spin Box

Step I :- Create an object from the TK() and  
Subsequently create an object from the  
Spinbox()

Step II :- Make the object so created onto the  
parent window and trigger the corresponding  
events

Step III :- Use the pack() to provide the direction  
using anchor attribute.

Step IV :- Use the mainloop() to terminate and  
trigger the events

Jan 10

580

## # PhotoImage

Step I :- Import relevant methods from the tkinter library

Step II :- Create parent window object and use the config method along with background colour attribute specified.

Step III :- Define a function finish with the messagebox widget which will display a message i.e. a warning message and subsequently terminate the program.

Step IV :- Define a function info we a listbox widget along within the object of the same use the listbox object along with insert() and insert the same and finally use the grid() with ipadx attribute

Step V :- Define a function about w within Label widget and text attribute and subsequently use the grid()

Step VI :- Use photoimage widget with file and filename with .gif extension.

Step VII :- 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.

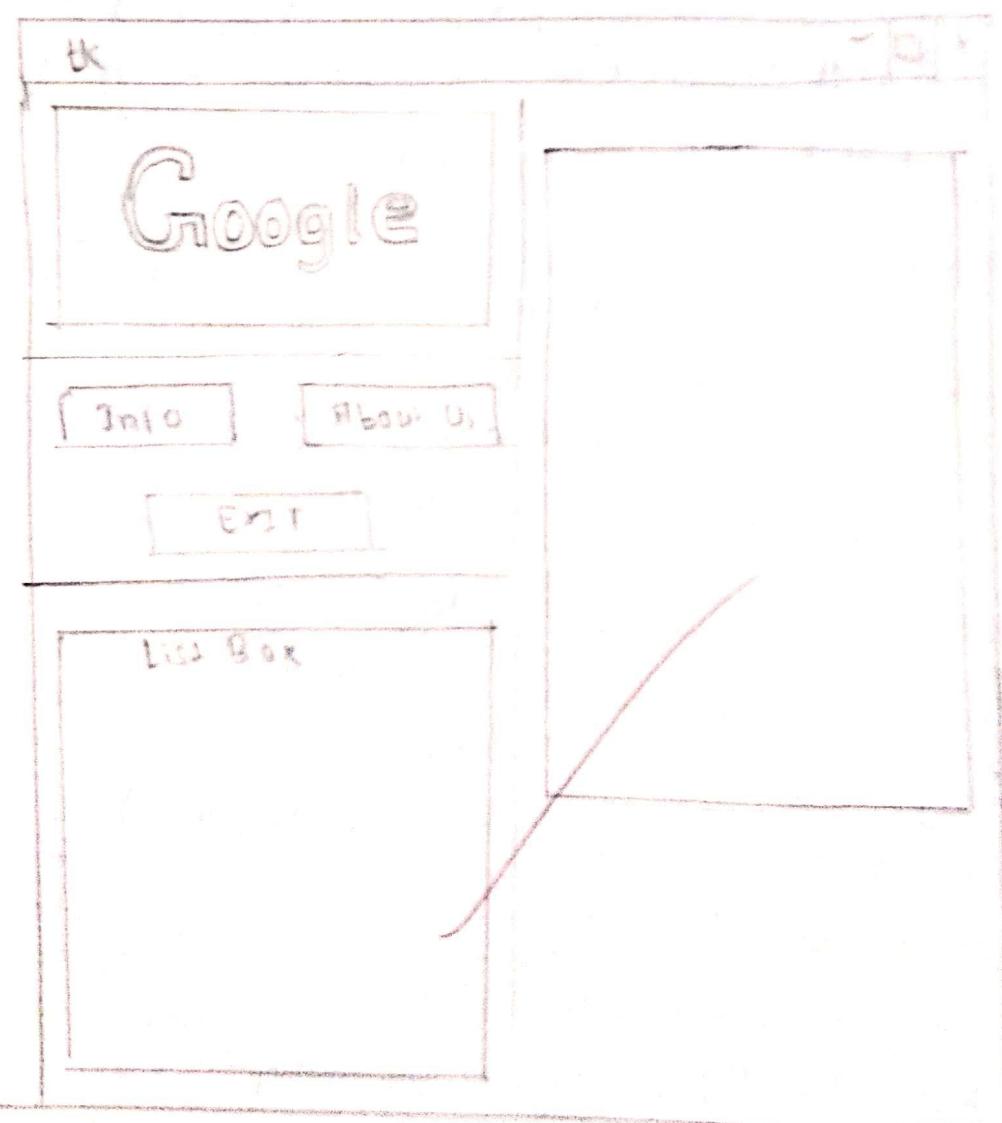
## # PhotoImage

048

```
from tkinter import *
import messagebox
root = Tk()
root.config(bg="grey")
def finish():
    messagebox.askokcancel("Warning", "This will end the program")
    if response == 'yes':
        quit()
def info():
    list1 = Listbox()
    list1.insert(1, "Name : Android")
    list1.insert(2, "Developed by: Google")
    list1.grid(ipadx=30)
def aboutus():
    list2 = Label(text="About Us")
    list2.grid(ipadx=30)
    list3 = Label(text="Google Android II")
    list3.grid(ipadx=24)
p1 = PhotoImage(file="download.gif")
p1 = Frame(root, height=35, width=5)
p1.grid(row=1, column=0)
f1 = Frame(root, height=250, width=500)
f1.grid(row=1, column=1)
p2 = p1.subsample(5, 4)
l1 = Label(p1, image=p2, relief=FLAT)
l1.grid(row=1, column=0, padx=20, pady=15)
l2 = Label(f1, image=p1, relief=SUNKEN)
l2.grid(padx=25, pady=10)
```

```
b1 = Button( f1, text = "Information", relief = SUNKEN, command = info )  
b1.grid( row = 1, column = 0 )  
b2 = Button( f1, text = "About Us", relief = SUNKEN, command = choices )  
b2.grid( row = 1, column = 2, padx = 5 )  
b3 = Button( f1, text = "EXIT", relief = SUNKEN, command = finish )  
b3.grid( row = 2, column = 1, ipadx = 15 )  
  
mainloop()
```

# Output :-



Step VIII :- Similarly create another frame object as declared by step VII.

Step IX :- Create another object and else the subsample with the values 5,4.

Step X :- Use Label widget along with the frame object, relief attribute and subsequently use the grid()

Step XI :- Now create button object dealing with different section of frame

Step XII :- Now finally call the mainloop() to trigger the events.

Ans

## # Canvas

Step I :- Import the relevant library from the Tkinter.

Step II :- Create parent window object and then with the ~~title~~ method.

Step III :- Now create an object from canvas(). use attribute height and width.

Step IV :- Use method create\_line, create\_oval, create\_arc along with canvas object and use the co-ordinates and simultaneously use pack()

Step V :- And Finally use the mainloop() to trigger the events.



# Canvas

050

```
from tkinter import *
```

```
root = Tk()
```

```
root.title("Pie")
```

```
c1 = Canvas(root, height=1000, width=1000)
```

```
coord1 = 200, 70, 10, 200
```

```
c1.create_arc(coord1, start=10, extent=180, fill='red')
```

```
coord2 = 200, 70, 70, 200
```

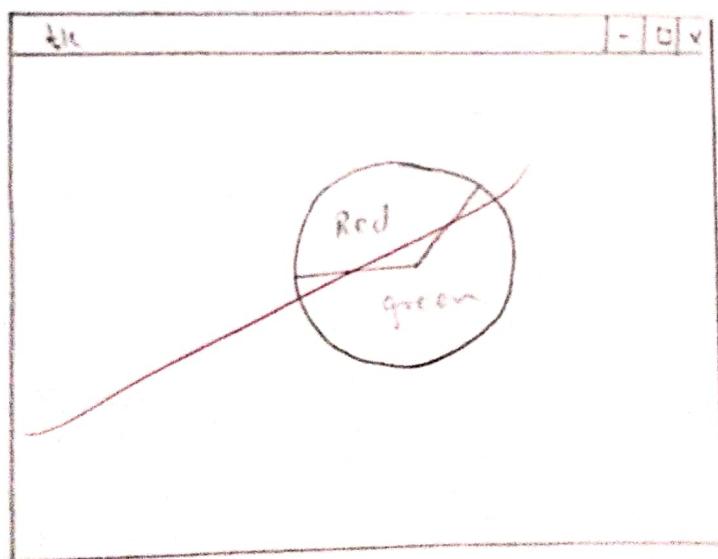
```
c1.create_arc(coord2, start=180, extent=180, fill='green')
```

```
c1.pack()
```

```
mainloop()
```

# Output

Drawing



020

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



Aim :- Database Connectivity

### # Algorithm

Step I :- Import the dbm 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 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 1 match or else not found 1 unmatched.

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

# 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 create cursor object using the cursor() and from the connection object created.

Step IV :- Now use the execute() for creating the table with the column name and respective datatype.

Step V :- Now with 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.~~

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 .

## # Source Code

```
import os, sqlite3  
con = sqlite3.connect("Student.db")  
c = con.cursor()  
c.execute("CREATE TABLE Student (roll INTEGER, name TEXT)")  
c.execute("INSERT INTO Student Values(1720, 'Shivam'),  
          (1729, 'Vikram'), (1731, 'Aadarsh')")  
c.execute("SELECT * FROM Student WHERE roll = 1729")  
print(c.fetchall())  
con.commit()  
c.close()
```

## # Output :-

[ (1729, 'Vikram') ]

Ans 19