

INDEX

No.	Title	Page No.	Date	Staff Member's Signature
	Prac-N0.1			In complete
1]	Demonstrate the use of different file accessing mode, attribute and different method.		25/12/19	Mr. Dinesh 25/12/19
2]	Programs with iterables and iterators.		2/12/19	Mr. Dinesh 16/12/19
3]	Program to demonstrate the Exceptions handling.		16/12/19	Mr. Dinesh
4]	Regular Expression		8/1/20	Mr. Dinesh
5]	use of GUI application		27/1/20	Mr. Dinesh
1)	Pack method			
2)	Radio Button			
3)	Scrollbar			
4)	Message box			
6)	Spin box		10/12/20	Mr. Dinesh
7)	Paned windows			
8)	Canvas			

INDEX

```

fileobj = open("abc.txt", "w")
fileobj.write("Computer with Python" + "\n")
fileobj.close()
>>> ( computer with Python
      Sockin kongjya
      having
      Subject )

```

```

fileobj = open("abc.txt", "r")
str1 = fileobj.read()
print ("use of read method is:", str1)
fileobj.close()

```

```

having
Subject

```

```

>>> ( use of read method is: Computer with Python
      Sockin kongjya having Subject )

```

Step 1: 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.

```

fileobj = open("abc.txt", "w")
fileobj.write("Computer with Python" + "\n")
fileobj.close()
>>> ( use of read method is: Computer with Python
      Sockin kongjya having Subject )

```

Step 2: 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.

```

fileobj = open("abc.txt", "r")
str2 = fileobj.readline()
print ("use of readline is:", str2)
fileobj.close()

```

```

>>> ( The structure of readline is: Computer with Python
      Sockin kongjya
      having
      Subject )

```

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

Q8

```
fileobj = open ("abc.txt", "r")
str3 = fileobj.readlines()
print ("The output of readlines method is, str3")
fileobj.close()
```

>>> (The output of readlines method is computer
with Python In , Sackin Kangjiga In having
In subject \n)

Step 4: Now open the file object in
write mode to write some

another content close subsequently
then again open the file object
in 'wt' mode that is the
update mode and write contents.

```
a = fileobj.name
print ("name of file (name attribute) is, a")
>>> ( name of file (name attribute) is, abc.txt)
b = fileobj.closed
print ("close attribute is, b")
>>> (close attribute is, true)
c = fileobj.mode
print ("file mode is, c")
>>> ( file mode is )
d = fileobj.softspace
print ("softspace is, d")
>>> ( softspace is 0)

fileobj = open ("abc.txt", "w")
fileobj.write ("Sackin")
fileobj.write ("Kangjiga")
fileobj.close()

>>> Sackin
>>> Kangjiga
```

```

file obj = open ("abc.txt", "rt")
str1 = fileobj.read()
print (" output of rt ", str1)
fileobj.close()

>>> ( output of rt : sacking)

fileobj = open ("abc.txt", "r")
str2 = fileobj.read()
print (" output of read mode ! ", str2)
fileobj.close()

>>> ( output of read mode ! : sacking)

file obj = open ("abc.txt", "a")
fileobj.write (" data structure ")
fileobj.close()
str3 = fileobj.read()
print (" output of append mode : ", str3)
fileobj.close()

>>> ( output of append mode : sacking data structure)

```

Step 5: Open the fileobject in read mode, display the update written contents and close open again in 'rt' mode with parameter passed and display the output subsequently.

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

fileobj = open ("abc.txt", "r")
 pos = fileobj.readline()
 print (" tell w:", pos)

26

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 argument with opening the fileobject in read mode and closing subsequently.

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

```
a = open ("abc.txt", "r")
str1 = a.read()
x = input ("Enter a word ")
count = 0
for word in str1:
    if (word == "the"):
        count += 1
print (count)
```

~~Print~~

>>>

6

Practical = 2

`mytuple1 = ("Sackin", "deopat", "Aniket")`

`my_iter = iter(mytuple1)`

`Print(next(my_iter))`

`Print(next(my_iter))`

`Print(next(my_iter))`

`>>> Sackin`

`deopat`

`Aniket`

Step 2: Create a tuple with element that we need.

Step 2: use `iter` and `next` method for iterating element and getting next iterating element in the tuple.

Step 3: Print the next iterating element and get output.

Aim: `iter` method using conditional statement.

Step 1: The similar output can be obtained by using `for` conditional statement.

`>>> Sackin`
`Kangjiga`

Step 2: An iterable variable is to be declared in `for` loop which will iterate.

To write a program using the iterable method for displaying set of odd numbers.

Step 1: Define a class and with in that define the iter() method which will initialize the first element with in the container object.

Step 2: Now use the next method and define the logic for displaying the odd values.

Step 3: about code The will throw exception is The object are not able to define the elements.

Step 4: Now create an object from the give class and pass those object as per argument to the iter() method.

Step 5: Now using this conditional object display all values from the given container

28
class odd:
 def __iter__(self):
 self.num = 1
 return self
 def __start__(self):
 num = self.num
 num += 2
 return num
 def __next__(self):
 num = self.num
 self.num += 2
 return num

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

=>> enter number 8
1
3
5
7

```

class myclass:
    def __iter__(self):
        self.a = 1
        return self

    def __next__(self):
        if self.a <= 10:
            x = self.a
            self.a += 1
            return x
        else:
            raise StopIteration

myobj = myclass()
myiter = iter(myobj)
for i in myiter:
    print(i)

```

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

Aim: To write a program using the iterable object to display set of first 20 numbers.

Step 1: Define a iter() method with an argument and initialize to the first value.

Step 2: For extracting the next element from the container used the next method with an argument and compare the number of element required in a container by using the conditional statement.

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

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

Aim : Iter method using map function.

listnum = [0, 4, 5, 7]

```
listnum = list(map(lambda x: x%5, listnum))
print(listnum)
```

Step 1: Declare a listnum variable
and declare some element.

Then

Step 2: Now use map method
with help of lambda function

else:

return "odd"

if

```
if (x%2 == 0):
```

return "even"

Step 3: give two argument display
the output

```
dist = map(even, listnum))
```

Output :

[0, 4, 1, 2]

Step 4: define even function with in
parameter then using conditional
statement check whether the
number is even and odd
and return respectively.

```

118 def square(x):
    return (x**2)
def cube(x):
    return (x**3)
function = [square, cube]
for x in range(10):
    Sackin = list(map(lambda x: x(function), function))
    print (list(Sackin))

```

Aim: To find square and cube of given number using map function.

~~Step 1:~~ Define a function name square with a parameter which will obtain output of square value of given number.

~~Step 2:~~ In similar fashion declare function cube of get the value raised 3 and return the sum

Step 1: Call the declared function using direction call.

~~empty[]~~

Aim: without using map direction

Step 1: Define a list variable with a given set of number.

~~Step 2:~~ Define an empty list which will contain output and use traditional statement and also use append() method.

Step 3: For finding the square value and finally print statement for the output

~~Output:~~ [1]
[1, 4]
[1, 2, 9]

Practical = 3

Aim: To write a program using The exception block method to handle the environmental error.

Step 1: use the block to define the normal process of action

Step 2: for eg: Define file object & open the file in write mode & some content into the file.

Step 3: use the except block with IO error - environmental error & print the appropriate message.

Step 4: else display the message that the operation is carried out successfully.

```
try:  
    fileobj=open("abc.txt","w")  
    fileobj.write("Python is a useful language")  
except IOError:  
    print("It's Environmental Error")  
else:  
    print("operation is done")
```

Output :

Operation is done

The operation was successful
The operation was successful
The operation was successful
The operation was successful

```

try:
x = int(input("Enter a no:"))
except ValueError:
    print("This is value error!")
except IOError:
    print("This is environmental error!")
else:
    print("Operation successful!")

```

Output:

```

>>> Enter a no: 56544
Operation successful!

```

Aim : To write a program for demonstrating the use of value error in the given program statement.

Step 1: Accept the value from the user & if it is a valid value display the enter value terminate the condition by using break statement

Step 2: Define the except block with value error as a keyword & display the appropriate message.

```

>>> Enter a no: T
This is value error!

```

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

Practical - 4

Topic - Regular Expression

Step 1: Import re module declare Pattern and Sequence

Pattern = re.compile("F4CS")
Sequence = "F4CS represents computer science stream"

if re.match(pattern, sequence):

else:

 print("Matched found!")

>>> matched found!

declare arguments : if arguments matched then print the same
otherwise print pattern "Not Found!"

import re
pattern = re.compile('d+')
string = 'hello 123, howdy 789', 'us howru'
output = re.findall(pattern, string)
print(output)

numerical value (segregation)

Step 2: Import re module declare Pattern with literal and meta character declare String value use re.findall() with arguments and print the same.

>>> ['123', '789', 'us']

Step 3: Import re module declare Pattern with meta character use re.split() and print the output.

import re
pattern = re.compile('d+')

string = 'hello 123, howdy 789', 'us howru'
output = re.split(pattern, string)
print(output)

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

```

# no space :
import re
string = 'abc def ghi'
pattern = r'1$'
replace = ''
v1 = re.sub(pattern, replace, string)
print(v1)
>>> abcdefghi

## group()
import re
sequence = 'python is easy to learn'
v = re.search('1a python', sequence)
print(v)
v1 = v.group(1)
print(v1)

>>> <- SRE_Match object at 0x0281Dc00>
Python

## verifying the given set of phone numbers
import re
list1 = ['8004567891', '9145673210', '7865432981']

for value in list1:
    if re.match(r'[8-9][1-3][0-9]{3}3210=',
               value or len(value) == 10):
        print("Criteria matched")
    else:
        print("Criteria failed")

>>> Criteria matched
Criteria matched
Criteria failed

```

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 subsequently use the group(), with dot operator as search() gives memory collection using group() it will show up. The method string.

Step 6: Import re module declare list with numbers use the conditional statement here we have used up the for condition statement. Use if condition statement for checking first number is either 8 or 9 and next number are in range of 0 to 9 and check whether the entered number are equal to 1 if criteria matches print cell number otherwise print failed.

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

Step 8: Import re module declare the host and domain name declare pattern for separating the host and domain name use findall() and print the output respectively.

vowels

```
import re
str1 = 'plant is file slide overall'
output = re.findall(r'\b[aeiouAEIOU]\w+', str1)
print(output)
>>> ['is', 'overall']
```

host and domain

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

Ques 10. Write a program to find all the words starting with vowel in a given sentence.

DYK

Counting of first 2 letters:

```

import re
s = "mr,ms,b,ms,c,mr,t"
p = r'[ms /mr]t'
o = re.findall(p,s)
print(o)

m=0
f=0
for v in o:
    if (v=='ms'):
        f = f+1
    else:
        m = m+1
print("no. of males is: ",m)
print("no. of females is: ",f)

```

```

>>> s'mr, 'ms', 'ms', 'mr'
no. of males is: 2
no. of females is: 2

```

Step 9: Import re module enter a string use pattern to display only two element of the particular string use.findall() declare the re variable with initial value as zero use for condition and subsequently use re if condition check whether condition is satisfying add up to re or else increment value and display the value subsequently.

Practical 5

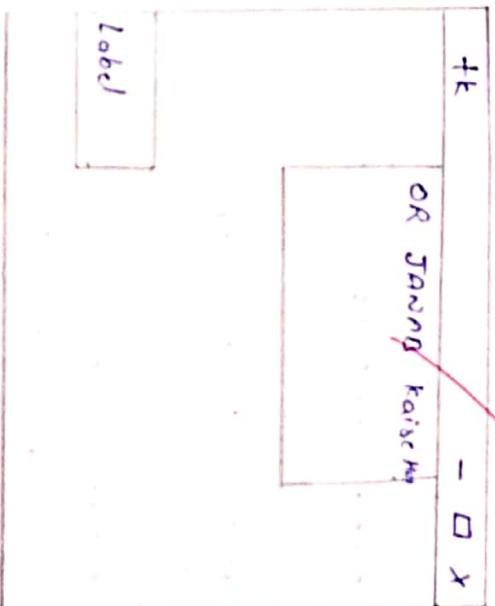
Aim: To make use of GUI Application along with the basic pack method.

Algorithm:

Step 1: use the tkinter library for importing the sentence of text widget.

Step 2: Create a variable from a text variable & position it onto the parent window.

Output:



Step 3: use the pack() along with the object created from text method & use the parameter if Side = TOP, padx = 20 ipadx = 40, pady = 50

Step 4: use the main loop method for triggering corresponding event

```
from tkinter import *
root = Tk()
T1 = Text(root)
T1.insert( END, "OR JANNG kaise HO ! ")
T1.pack(side=TOP, padx=20, pady=30, ipadx=40,
        ipady=50)
L1 = Label(root, text="label", bg="red", fg="blue")
L1.pack(side=LEFT, padx=10, ipadx=20, ipady=30)
root.mainloop()
```

Program :

```

from tkinter import *
def sel1():
    selection = "Sachin"
    label.config(text=selection)
def sel2():
    selection = "Deepak"
    label.config(text=selection)
def sel3():
    selection = "RD"
    label.config(text=selection)
def sel4():
    selection = "RK"
    label.config(text=selection)
root = Tk()
var = IntVar()
L1 = Label(root, text="Select any roll number")
L1.pack(side=TOP)
R1 = Radiobutton(root, text="1744", variable=var,
                 value=0, command=sel1)
R1.pack(anchor=N)
R2 = Radiobutton(root, text="1745", variable=var,
                 value=1, command=sel2)
R2.pack(anchor=N)
R3 = Radiobutton(root, text="1746", variable=var,
                 value=2, command=sel3)
R3.pack(anchor=N)

```

Step 5: Now repeat above step with a label method which takes the following argument.

- Name of parent window
- Text attribute which defines the string
- The background colour (bg)
- The foreground colour (fg)

Now use pack() with relevant attribute.

Aim : To make use of Radio Button widget for selection of one of the multiple options.

Algorithm :

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

Step 2: Define a function which tells user about given selection made from multiple option available.

Step 3: Use the config method with label method & cell the variable as an argument within method.

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

8

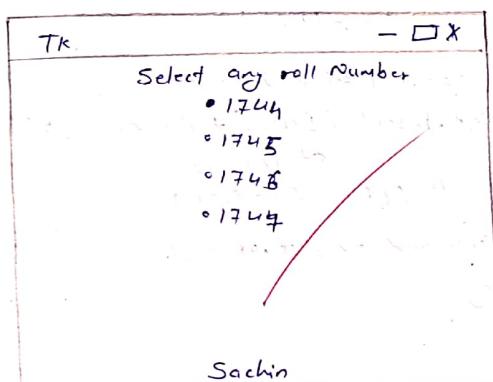
Step 5 : Now create object of Radiobutton -
which will take following arguments
i) Positioning on parent window.
ii) Text Variable.
iii) Define variable argument
iv) Corresponding value and trigger
the given function.

Step 6 : Now call the pack method for
corresponding Radio object so
created and specify argument
as an anchor attribute

Step 7 : Now define a label object
and place it into parent
window using Pack method
& finally use mainloop method.

R4 = Radiobutton (root, text="1747", variable=var
value=3, command=sel4) R4
R4.pack (anchor=N)
label = Label (root)
label.pack (side=Bottom)
root.mainloop()

Output :



Program 1

```
from tkinter import *
```

```
root = Tk()
```

```
root.geometry("400x400")
```

```
root.title("Batch Roll numbers")
```

```
l = Label(root, text="Batch Roll numbers")
```

```
l.pack()
```

```
scroll = Scrollbar(root)
```

```
scroll.pack(side=RIGHT, fill=Y)
```

```
mylist = Listbox(root, yscrollcommand=scroll.set,
```

```
bg="lightblue")
```

```
for num in range(1, 8):
```

```
    mylist.insert(END, "Roll Number: " + str(num))
```

```
mylist.pack(side=LEFT, fill=BOTH)
```

```
scroll.config(command=mylist.yview)
```

```
root.mainloop()
```

Output:

Aim: To make use of Scroll Bar widget

The GUI application.

Algorithm:

Step 1: Import tkinter library to use scroll bar widget.

Step 2: Create an object corresponding to scroll parent window & create an object from scrollbar & place it on the parent window so created.

Step 3: Create an object of label method to provide a heading and place it on parent window.

Step 4: Use pack method along with object of scrollbar method & use argument side & fill.

Step 5: Create an object of list box method and place it on the parent window with attribute of scroll command.

Step 6: use for loop along to insert values is the object of list box by using insert method.

Output:

Step #1 We create method along with
Scylla box object & use
command attribute.

Step 4: directly call the mainloop method.

B Batch Roll Numbers	
*	Roll Number : 1773
*	Roll Number : 1774
*	Roll Number : 1775
*	Roll Number : 1776
*	Roll Number : 1777
*	Roll Number : 1778
*	Roll Number : 1779
*	Roll Number : 1780

丁巳

Practical - 1

```
Program:
from tkinter import *
from tkinter import messagebox
```

def messagebox():
 messagebox.showerror("Action", "do you want to continue?")

messagebox("Error", "Can't load command-msgbox")

the process")

```
root = Tk()
root.config(bg = "yellow")
```

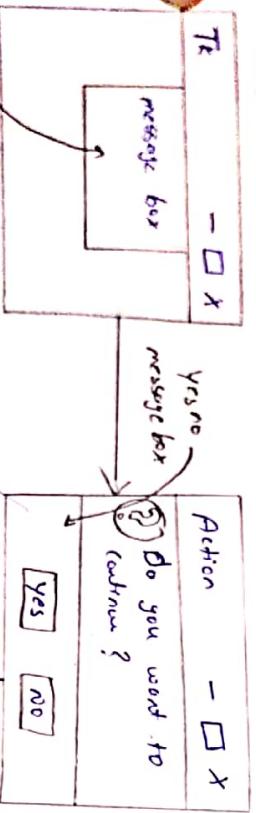
Al = Button(root, text = "message box", bg = "orange")

command=msgbox)

Al.pack()

root.mainloop()

Outputs:



Error
button

>Error
button
can't load the process

Error
messagebox

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

Algorithm:

Step 1: Import relevant method from tkinter library.

Step 2: Define a function and use messagebox along with different methods available which contains one or more argument.

Step 3: Create an object from button method and place it onto the Parent window with text and command attribute specified.

Step 4: use pack method and finally use the main loop method.

Program:

44

```
from tkinter import *  
def main ():
```

```
root = Tk()
```

```
root.geometry ("1000x2000")
```

```
root.config ("bg= "red ")
```

```
root.title ("First window")
```

```
step1: Import the relevant method from  
Tkiner library
```

```
step2: Define a function and create a  
object of given window by using  
the tkon method namely config  
title , minsize.
```

```
step3: Define a button object which will be
```

```
placed on the current window to  
traverse and define another  
button which will be used to  
exit from the window and  
place it onto current window.
```

```
step4: Define another function which will
```

```
use the quit method to  
terminate the program.
```

```
tos = Tk()  
tos.geometry ("800x800")  
tos.title ("main window")  
B1 = Button (tos, text = "Continue", command = main)  
B1.grid (ipadx = 50, ipady = 50, padx = 10, pady = 20)  
B2 = Button (tos, text = "Exit", command = term)  
B2.grid (ipadx = 50, ipady = 50, padx = 10, pady = 10)
```

Step 4: main()

Top : Tk()

Top , Geometry ("600x400")

Top . config ("background" "grey")

Top . title ("Second window")

Ok : Button (Top , text : "next page" , command : main)

Grid (Ok . w0 , ipadx : 20 , ipady : 20 , padx : 20 , pady : 20)

Cs : Button (Top , text : "last" , command : term)

Grid (Cs . w0 , ipady : 55 , padx : 70 , pady : 70)

mainloop()

output:

Main window - □ X

button of focus on
main window

First window - □ X

Button found
on Second window

Ok found
on Second window

Step 5: Now create an object of main window

and use various method like config, diff, geometry etc.

Step 6: Define two buttons which will be

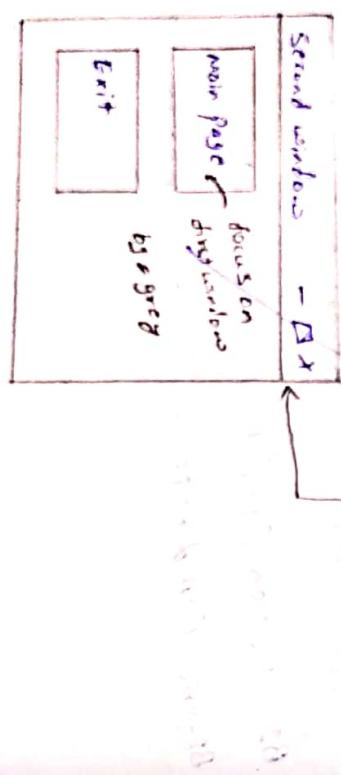
placed on the main window, one to
traverse another window and the
other to terminate the program.

Step 7: Define another function which will
various button placed on third window.

Define two button placed on third
window. Define two button respectively
and use the grid method along
with the two button.

Step 8: Finally we have mainloop method.

Author



Aim: write program by use of Spinbox

from tkinter import *

root = Tk()

s1 = Spinbox (root, from_=0, to=±20)

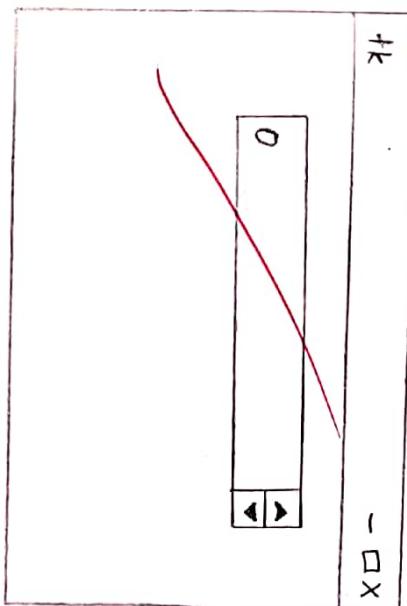
Alg 1:

Step 1: Create an object from tk method and subsequently create an object from the spinbox method.

Step 2: Make the object go created onto the parent window and trigger the corresponding events.

Step 3: Use the pack method to provide the direction using anchor method.

Step 4: Use the mainloop method to terminate



```
from tkinter import *
```

```
root = Tk()
```

```
p1 = Paned Window (bg="black")
```

```
p1.pack (fill=BOTH, expand=NO)
```

```
l1=Label (p1, text="SACHIN", bg="white")
```

```
p1.add(l1)
```

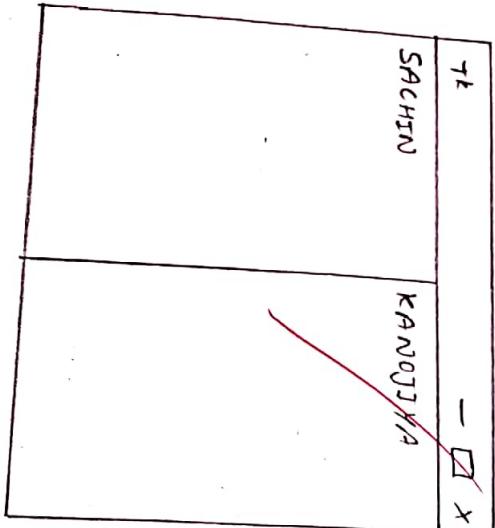
```
p2=Paned Window (p1, orient=VERTICAL, bg="red")
```

```
p2.add(p1)
```

```
l2=Label (p2, text="KANOJJA", bg="orange")
```

```
p2.add(l2)
```

```
root.mainloop()
```



Algo 1:

Step 1: Create an object from the Paned window and use the pack method with the attributes 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 first Paned window with orientation specified.

Step 4: Now create another Label object and place it onto the 2nd Paned window object and add it onto the 2nd place.

Step 5: Now use mainloop method.

from tkinter import *

root = Tk()

c1 = Canvas(root, height=500, width=800, bg="green")

oval = c1.create_oval(30, 150, 260, fill="grey")

line = c1.create_line(20, 30, 40, 50, fill="yellow")

arc = c1.create_arc(10, 180, 130, 50, fill="blue")

c1.pack()

root.mainloop()

Aim: write program by use of canvas.

Algo:

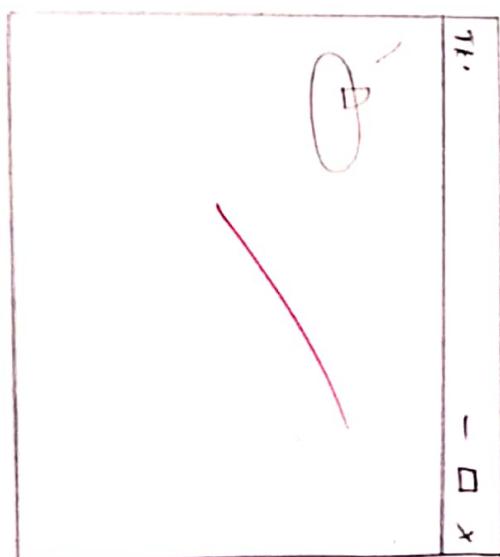
Step 1: Use the `master` method and create the object from `Canvas` method.

Step 2: Use the attribute `height`, `width` by colour and the Parent window object.

Step 3: Use the method `create_oval`, `line` and `arc` along with `canvas` object so created and use the coordinate.

Step 4: Now use the `fill` attribute to assign various colours.

Step 5: Now call `pack` method and ~~mainloop~~ method.



Diagram

Practical = 6

Q

```
import dbm
dbm.open("my self Sachin", flag = "c")
```

```
db = dbm.open("www", flag = None)
if db["www"] == None:
```

```
    print("good")
```

else:

```
    print("Not good")
```

Output:

Not good

Step 1: Import dbm library and use the open method for creating the database by specifying name of the database along with the corresponding flag.

Step 2: use the object for accessing the given web size and the corresponding regular for the web size.

Step 3: check whether the given url address with the regular of the pages is not equal to None then display the message from url address else not defined.

```
import os, sqlite3
connection = sqlite3.connect("stu.db")
```

b7

Step 1: Import the corresponding library taking
of data base connection.

Step 2: Now create connection objects using
sqlite library and connecting method
to create the new database.

Step 3: Now create the cursor objects
using cursor method from the connection
objects create in steps.

Step 4: Now use the executing method for
creating the table with the column
name and respective data type.

Step 5: Now use the cursor object use insert
statements for entering the values
co-ordinating into the different fields
considering the data types.

Step 6: use the commit method to
complete the transaction we
the connection objects.

Output :

```
[("Sachin", 1744), ("Lalit", 1745), ("Manju", 1746)]
```

Step 7: use the execute statement along with the cursor objects for assensing the value . The data base using selecting from our clause.

Step 8: Finally use the fetchall method for displaying the value for the table
use the cursor objects.

Step 9: use the execute method and the close table syntax for terminating the database finally we use the close method.

Dr. 2012

Project

AJ Project Based on GUI

Theory: This project based on GUI and the task is to make copy way to search any website.

Wk

Online website
Entertainment

youtube
wikipedia
wynk
google
w3school

LEARNING WEBSITES

Lynda
Udemy
Codecademy

GAMING WEBSITES

IGN
GameSpot
Kotaku

```

Btn6.grid(row=11, column=0)
new = 7
url7 = "https://www.w3schools.com"
def openweb7():
    webbrowser.open(url7,new=new)
Btn7 = Button(root, text = "w3school",command=openweb7)
Btn7.grid(row=10, column=20)

new = 8
url8 = "https://www.udemy.com"
def openweb8():
    webbrowser.open(url8,new=new)
Btn8 = Button(root, text = "Udemy", command=openweb8)
Btn8.grid(row=11, column=10)

new = 9
url9 = "https://www.codecademy.com"
def openweb9():
    webbrowser.open(url9,new=new)
Btn9 = Button(root, text = "Codecademy", command=openweb9)
Btn9.grid(row=13, column=10)

Label(root, text="GAMING WEBSITES", font=30)
14.grid(row=15, column=10)

new = 10
url10= "https://www.IGN.com"
def openweb10():
    webbrowser.open(url10,new=new)
Btn10 = Button(root, text = "IGN", command=openweb10)
Btn10.grid(row=17, column=0)

new = 11
url11= "https://www.GameFAQs.com"
def openweb11():
    webbrowser.open(url11,new=new)
Btn11 = Button(root, text = "GameFAQs", command=openweb11)
Btn11.grid(row=17, column=20)

new = 12
url12 = "https://www.Gamespot.com"
def openweb12():
    webbrowser.open(url12,new=new)
Btn12 = Button(root, text = "Gamespot", command=openweb12)
Btn12.grid(row=18, column=10)

new = 13
url13 = "https://www.Kotaku.com"
def openweb13():
    webbrowser.open(url13,new=new)
Btn13 = Button(root, text = "Kotaku", command=openweb13)
Btn13.grid(row=19, column=10)
root.mainloop()

```

```

from tkinter import *
import webbrowser

root = Tk()
root.geometry("500x500")
l1=Label(root, text="online website", font=100)
l1.grid(row=0, column=10)
l1.grid(row=0, column=10)

l2=Label(root, text="Entertainment", font=30)
l2.grid(row=1, column=10)

new = 0
url1 = "https://www.youtube.com"
def openweb1():
    webbrowser.open(url1,new=new)
    l1.config("Click")
    l1.grid(row=2, column=0)
    l1.grid(row=2, column=0)

new = 1
url1 = "http://www.google.com"
def openweb10():
    webbrowser.open(url1,new=new)
    l1.grid(row=2, column=20)
    l1.grid(row=2, column=20)

new = 2
url12 = "https://www.wikipedia.com"
def openweb2():
    webbrowser.open(url12,new=new)
    l1.grid(row=2, column=20)
    l1.grid(row=2, column=20)

Btn1 = Button(root, text = " youtube ", command=openweb1)
Btn1.grid(row=2, column=0)

Btn2 = Button(root, text = " wikipedia ", command=openweb2)
Btn2.grid(row=2, column=20)

new = 3
url13 = "https://www.w3school.com"
def openweb3():
    webbrowser.open(url13,new=new)
    l1.grid(row=6, column=20)
    l1.grid(row=6, column=20)

Btn3 = Button(root, text = " w3school ", command=openweb3)
Btn3.grid(row=6, column=20)

new = 4
url14 = "https://www.wynk.com"
def openweb4():
    webbrowser.open(url14,new=new)
    l1.grid(row=8, column=20)
    l1.grid(row=8, column=20)

Btn4 = Button(root, text = " wynk ", command=openweb4)
Btn4.grid(row=8, column=20)

l1=Label(root, text="LEARNING WEBSITS", font=30)
l1.grid(row=9, column=10)

new = 5
url15 = "https://www.Coursera.com"
def openweb5():
    webbrowser.open(url15,new=new)
    l1.grid(row=10, column=20)
    l1.grid(row=10, column=20)

Btn5= Button(root, text = "Coursera", command=openweb5)
Btn5.grid(row=10, column=20)

new = 6
url16= "https://www.Lynda.com"
def openweb6():
    webbrowser.open(url16,new=new)
    l1.grid(row=11, column=20)
    l1.grid(row=11, column=20)

```

B) Project based on database.

Theory: These project creates when you want security in your page you required entry password and email id.

```
from tkinter import *
from tkinter import messagebox
import os,sqlite3
main=Tk()
main.title("Main")
main.geometry("400x500")
main.minsize(200,300)
con = sqlite3.connect("File.db")
cursor = con.cursor()
cursor.execute("CREATE TABLE rh(fname text, lname text, username text, email text, contact integer, password text)")
text.submit()
def submit():
    con = sqlite3.connect('File.db')
    c = con.cursor()
    if username.get() != '' and email.get() != '' and password.get() != '':
        VALUES(:l_name,:username,:email,:contact,:password)''{'fname':f_name.get(),'lname':l_name.get(),'username':username.get(),'email':email.get(),'contact':contact.get(),'password':password.get()}
        cursor.execute("INSERT INTO rh",VALUES)
        messagebox.showinfo("Info","Registration successful. Go back to Login to your Account.")
    else:
        messagebox.showerror("Error","Please Enter The Details.")
```

```
def login():
    if(e1.get()):
        if(e2.get()):
            logged = Toplevel()
            logged.title("Home")
            logged.geometry("400x500")
            Label(main,logged,text=e1.get()).pack()
```

```
else:
    messagebox.showerror("Error","Incorrect Password!!")
```

```
def signUp():
    sign = Toplevel()
    sign.title("Sign up")
    sign.geometry("600x1200")
```

```
global f_name
global l_name
global username
global email
global contact
global password
```

Enter Username or Email:

Enter Password:

Login

Forgot Password?

[Click here for Sign Up](#)

New User?

```
Page",fg="red",height=2,width=30).grid(row=0,column=0,columns=2)
Label(sign, text="\n").grid(row=3,column=0)
Entry(sign,width=20,textvariable=f_name).grid(row=2,column=1)
f_name=StringVar()
Label(sign, text="First Name:"'width=10).grid(row=2,column=0,sticky=W)
Entry(sign,width=20,textvariable=f_name).grid(row=3,column=1)
Label(sign, text="\n").grid(row=4,column=0)
Label(sign, text="Last Name:"'width=10).grid(row=4,column=0,sticky=W)
Entry(sign,width=20,textvariable=l_name).grid(row=3,column=1)
username=StringVar()
Label(sign, text="\n").grid(row=4,column=0)
Label(sign, text="Username:"'width=10).grid(row=4,column=0,sticky=W)
Entry(sign,width=20,textvariable=username).grid(row=4,column=1)
email=StringVar()
```

```
sqlite222
Label(sign, text="\n").grid(row=5, column=0)
Label(sign, text="Email:", width=10).grid(row=5, column=0)
Entry(sign, width=20, textvariable=email).grid(row=5, column=1)
```

```
contact=StringVar()
Label(sign, text="\n", bg="steel blue").grid(row=6, column=0)
Label(sign, text="Contact:", width=10).grid(row=6, column=0)
Entry(sign, width=20, textvariable=contact).grid(row=6, column=1)
```

```
password=StringVar()
Label(sign, text="\n").grid(row=11, column=0)
Label(sign, text="Password:", width=10).grid(row=11, column=0)
Entry(sign, width=20, textvariable=password).grid(row=11, column=1)
```

```
button(sign, text="Submit", height=2, width=30, command=submit).grid(row=12, column=0, columnspan=2)
```

```
Label(sign, text="\n").grid(row=13, column=0)
Label(sign, text="Already have an account?").grid(row=14, column=0, columnspan=2)
```

```
Button(sign, text="Go Back To Login Page", command=lambda:sign.withdraw()).grid(row=15, column=0, columnspan=2)
Label(main, text="\n").grid(row=2, column=0)
```

```
e1=StringVar()
```

```
Label(main, text="Enter Username or Email:", fg="black", height=2, width=30).grid(row=3, column=1, sticky=w)
Entry(main, width=30, textvariable=e1).grid(row=3, column=2)
```

```
e2=StringVar()
```

```
Label(main, text="\n").grid(row=5, column=0)
Label(main, text="Enter Password:", fg="black", height=2, width=30).grid(row=6, column=1, sticky=w)
Entry(main, width=30, textvariable=e2).grid(row=6, column=2)
```

```
Label(main, text="\n").grid(row=8, column=0)
Button(main, text="Login", height=3, width=25, command=logIn).grid(row=8, column=1)
```

```
Label(main, text="\n").grid(row=10, column=0)
Label(main, text="\tForgot Password?", fg="red").grid(row=10, column=1)
```

```
Label(main, text="\n").grid(row=12, column=0)
Label(main, text="New User?").grid(row=13, column=1)
Button(main, text="Click here for sign up", command=signup).grid(row=12, column=2)
```

```
con.commit()
```

```
con.close()
```

```
mainloop()
```