```
file obj = open (
file obj.write ("Computers science subjects" + "\n")
file obj write ("DBMS \n Python \n DS \n")
file obj close()

file obj = open ("abc.txt", "r")
str1 = File obj read()
print(" The output of read method:", str1).
file obj . Close()
```

>>> The Output of read method : Computer Science subject
DBMS
Python
D S

* readline()

```
file obj = open("abc txt", "r")
str2 = file obj . readline()
print("The output of realine method:" str2)
file obj close()
```

>>> The Output of readline method : Computer Science subj

#Readlines

```
File obj = open ("abc.txt", "r")
Str 3 = File obj realines()
print("The Ouput of readlines method:", str3)
File obj . close()
```

>>>

Practical No. 1

x **AIM** : Demonstrate the use of different accessing mode, different attributes file and Read method.

Step 1 : Create a file object using open method and use the write accessing mode followed up by writting some contents onto the file and than closing the file.

Step 2 : Now open the file in read mode 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 file then file mode in which if is opened whether the file is still open or close and finally the output of the softspace attribute

Step 4 :

Now open the file object in write mode write some another content close the subsequantly there again open the file Object in 'w+' mode that is the update mode and write contents

Step 5: open file object in read mode, display the update written contents and close open again 'r+' mode with parameter passed and display the output subsequently.

Step6: Now open file object in append mode open write method write the content close the file object again. Open the file object in read mode and display the append output

Step7: open the file object in read mode, declare a variable and perform file object dot tell method and store the output consequently in variable

Step8: Use the seek method with the argument with opening the file object in read mode and closing subsequently.

Step9: open file object with read mode also use the readlines method and store

File attributes

```
a= fileobj.name
print ("name of file (name attribute):", a)
>>> (Name of file (name attribute, abc.txt)
b = file obj.closed
print ("(close) attribute :") b)
>>>(close) attribute =, True
c= file obj.mode
print ("file mode", c)
>>> ("filemode", 'r')
b= file obj.softspace
print ("soft space," al)
>>> ("soft space:", 0)
```

```
# wt mode
fileobj = open ("abc.txt", "wt")
fileobj.write("Saurabh")
fileobject.close()
# r+ mode
fileobj=open ("abc.txt", "r")
str1=fileobj.read(r)
print("output of r+", str1)
fileobject.close()
>>>(output of r+", Saurabh)
```

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

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

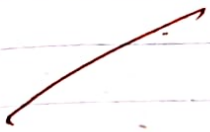
```
# append mode.
fileobj = open ("abc.txt", "a")
fileobj.write ("Datastructure")
file.close()
fileobject = open ("abc.txt", "r")
str3 = fileobj read()
print ("Output of append mode", str3)
fileobj.close()
>>> ("O/p of append mode:", "Sawaab", "Data structure")

# tell()
fileobj = open ("abc.txt", "r")
pos - fileobj.tell()
print ("tell():", pos)
fileobject.close()
>>> ("tell():", pos)

# seek().
fileobj = open("abc.txt", "r")
str4 = fileobj.seek(0,0)
str8 = fileobj.read(10)
print ("The beginning of the line is =", str8)
```

Step 7 – Open the file object in read mode declare a variable & perform file object dot tell method and store the output consequently in variable.

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



iterator class which has
namely __iter()__ and

using iterable objects. for
in range 1 to 10

and
value.

argume
by

object

Code

```
class add:
    def __iter__(self):
        self.num = 1
        return self
    def next(self):
        if self.num <= 10:
            num = self.num
            self.num += 2
            return num
        else:
            raise StopIteration.
```

```
>>> y = count()
>>> z = iter(y)
>>> z.next()
1
>>> z.next()
3
5
>>> z.next()
7
>>> z.next()
9
>>> z.next()
11
```

Code

```
class power
    def __iter__(self):
        self.pow = 0
        return self
    def next(self):
        if self.p<=10:
            self.num=self.p
            self.p+=1
            p0 = 2** num
            return p0
        else:
            raise StopIteration
```

```
>>> p = power()
>>> x = iter(p)
>>> x.next()
    a = 2 1
>>> x.next()
  2 * 1+1 = 2
>>> x.next()
  2 * * 2 = 4.
>>> x.next()
    2** 3 = 8
```

6.2) Write a program using an iterator for calculating the power of a given number. For instance number entered is 2 then value calculated should be $1, 2^1, 2^2, 2^3, 2^4$

Algorithm:

Step 1: Define iter() with argument and intialize value and return the value.

Step 2: Now define next() with an argument and Compare the upper limit by using conditional statement.

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

e3) write a program using iterable concept to find factorial of number in range 1 to 10:

Step 1: Define a iter() with argument and intialize the value. and return the value.

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

Step3: Now create an object of the given class and pass this object in the iter method.

Q.2) write a program using an iterator for calculating the power of a given number. For instance number entered is 2 then value calculated should be $1, 2^1, 2^2, 2^3, 2^4$.

Algorithm:

Step1: Define iter() with argument and intialize value and return the value.

Step2: Now define next() with an argument and compare the upper limit by using conditional statement

Step3: Now Create an object of the given class & pass this object in the iter method.

code

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

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

:: Code

```
class result:
    def __iter__(self):
        self.m = 1
        return self
    def next(self):
        if self.m < 10:
            num = self.m
            self.m += 1
            table = 2 * num
            print("2 *", num, "=", table)
        else: raise StopIteration.
```

```
>>> m = mul()
>>> x = iter(m)
>>> x.next()
  2 * 1 = 2
>>> x.next()
  2 * 2 = 4
>>> x.next()
  2 * 3 = 6
>>> x.next()
  2 * 4 = 8
```

Q.3) Write a program using iterable con concept to find factorial as number in range (1 to 10)

Step1: Define a iter () with argument & initalize the value and return the value.

Step2: Define the next() with an argument and Compare the upper limit by using a conditional statement.

Step3: Now create an object of the given class & pass the this object in the iter method.

Q.4) Write a program using iterable concept to display multiply of 2 in range 1 to 10.

Algorithm:

Step1: Define a iter() with argument & initalize the valueand return the value.

Step2: Define the next() with an argument and compare the upper limit by using a conditional statement.

Step3: Now range an object of the given class & pass this object in the iter method.

TS

Practical NO. 3

AIM: Demonstrate the use of exception Handling.

Theory: An exception is an event which occurs during ~~except~~ exception of program which distrupt the normal flow of program. Thus a exception it must be handled immediately otherwise it will terminate & class the program.

Q.1) Write a program to check the range of the age of the students in given class a P.j age does not fall in given range use value error exception or a otherwise return the valid no.

Algorithm:

Step1: Define a function which will accept the age of the student from standard input.

Step2: Use if conditional to check whether the input age falls in range & so return the age else use value error exception.

```
def accept_age()
    age = int (input ("Enter your age :"))
    if age >30 or age<16 :
        raise value error :
    else :
        Print("your age is", age)

valid = False
while not valid :
    try :
    age = accept_age()
    valid = true
    except value error:
        print ("your age is not in range")
```

```
>>> Enter your age : 15
your age is not in range
Enter your age : 32
your aage i's not range :
Enter your age : 17
your age is 17 .
```

Code

8S.

```
while true
    try:
        a = int(input("Enter a number:"))
        print("valid number")
        break.
        except value error:
            print("Not a valid number ! try again")
```

>>> enter a number: 17 .2
Not a valid .number ! Try again
Enter a number: 17
valid number.

Step3: Define the while loop to check whether the boolean expression holds true. Use the try block to accept the age of student & terminate the looping condition.

Step4: Use except with value error & print the message not a valid range:

Q.2) Write a program to check whether the number in given class & if the number is a floating point we value error or exception for the given input

Algorithm:

Step1: Use try block & accept the input using input() & convert it into integer datatype and subsequantly terminate the block.

Step2: Use the except block with exception as value error & display appropriate message is suspicious code is part of try block.

Code

```
def divide(a, b):
    ans = a/b
    return ans
while True:
    try:
        a = int(input("Enter first number:"))
        b = int(input("Enter second number:"))
        ans = divide(a,b)
        print("division of", "a", "and", "b",
              "is", ans)
        break
    except zerodivision error:
        print("Error!")
```

>>> Enter first number : 1
>>> Enter second number: 1
>>> Division of 1 and 1 is 1
>>> Enter first number : 1
>>> Enter second number : 0
Error!

24/12/19

CODE 1:

```
import re
string = "hello 1234 abc4567"
result = re.findall("\d+", string)
result = re.findall("\D+", string)
print (result)
print (result1)
```

output:

```
>>> ['1234', '4567']
>>> ['hello', 'abc']
```

AIM: Demonstrate the use of regular expression

Theory:

Regular expression represents the sequence of characters which is mainly used for finding & replacing the given pattern in a string and for this purpose we import re module and common usage of regular expression involves following functionalities:

- Searches a given string
- Binding a string
- Breaking a string into smaller substring.
- Replacing part of string

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

ALGORITHM:

Step 1:

Now apply string & pattern in findall() and display the output.

Step 2:

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

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

ALGORITHM:

Step1: Import re module and apply a string.

Step2: Use search() with "\A python" and string as two parameters

Step3: Now display the output.

Step4: Now use if conditional statement for user to know whether the match is found or not.

Q.3) Write a regular expression to check whether the given mobile number starts with 8 or 9 & the total length of digit should be at most 10.

Algorithm:

Step1: Import re module and apply a string of mobile No. 8.

Step2: Now use for conditional statement to find if the number starts with 8 or 9 and the total number should length of 10. Use match() inside for statement to find the match in given string.

CODE2:

```
import re
string = "python is an important language"
result = re.search ("\Apython", string)
print (result)
if result:
        print ("match Found")
else:
        print ("match not found")
```

Output

```
>> <re match object : span = (0,6);
                      match = "python">
```

>> match found.

CODE 3:

```
import re
li = ["9876543210", "8765432109",
      "7654321098", "6543210987"]
for element in li:
    result = re.match ("[8-9]{1}[0-9]{9}",
                       element)
    if result:
        print ("Correct mobile no.")
        print (result.group(1))
    else:
        print ("Incorrect mobile no")
```

Output.

```
>>> correct mobile no.
9867654321O
correct mobile no.
8765432109
Incorrect mobile no.
Incorrect mobile no.
```

step3: Use if Conditional statement to know whether we have a match or not. If we have usegroup c) to digit should be atmost 10.

Algorithm:

Step1: Import re module and apply a string of mobile No 8.

Step2: Now use for conditional statement to find if the ~~and~~ number starts with 8 or 9 and the total number should length of 10. Use match() inside for statement to find the match in given string.

Step3: Use if conditional statement to know whether we have a match or not if we have use group() to display the output and if we dont display incorrect mobile no.

Q4) write a regular expression for extracting a word from given string along with space characters in between the word and subsequently extract the word without space character.

Algorithm: Step1: Import re module and apply a string.

Step2: Use findall() to extract a word from given string.

Step3: Use "\w*" to extract word along with space & use "\w+" to extract word without space.

Step 4: Now display the output.

5) Write a regular expression for extracting first and last word from a string.

Step1: Import re module and apply a string.

Step2: Use findall() in which we "^\w+" as one parameter then use "\w+$" as parameter to find last word of string.

Step3: Now display the result.

```
# code 4
import re
string = "python is Important "
result 1 = re.findall ("\w* ", string)
result2 = re.findall ("\w+", string)
print (result 1)
print (result2)
```

Output :

```
>>> ['python . ' ' 'is', ' ', important, "]
    ['python', 'is', 'important']
```

Code 5 :

```
Import re
string = " python is 'important"
result 1 = re.findall ("^\w+", string)
result 2 = re.findall ("\w+$", string)
print (result1)
print (result 2)
```

Output

```
>>> ['python']
>>> [+important]
```

CODE 6 :

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

write a regular expression for extracting the date in format dd-mm-yyyy by using the findall() where the string has following format Amit 201 24-12-2019

Algorithm:

Step1: Import re module and apply the string.

Step2: Use findall method and use '\d{2}-\d{2}-\d{4}' as an parameter.

Step3: Now display the output.

Write a re for extracting the
(1) Username from email id
(2) hostname from email id
(3) Both username & hostname from email id

Algorithm:

Step1: Import re module and apply a string.

Step2: Use findall() to find username hostname & both of email id.

Step3: Use "^\w+" for username use "t\w+:\" for hostname and use "[\w\.-]+" for both as parameter in findall()

Step4: Display the output.

Code :

```
import re
string = "abc@tcsc.edu"
result1 = re.findall("^\w+", string)
result2 = re.findall("t\w+.\w+$", string)
result3 = re.findall("[\w\.-]+", string)

print(result1)
print(result2)
print(result3)
```

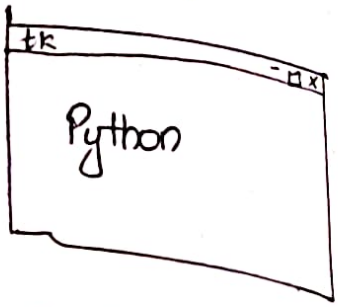
Output:

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

07/01/2021

```
from Tkinter import*
root = Tk()
l = label (root, text = "Python")
l . pack()
root. mainloop()
```

Output:



#2: Label, attributes

```
from Tkinter import*
root = Tk()
l = label (root, text = "Python")
l.pack()
l1= Label (root, text = "CS!"
          fg = "black", font = "10", bg = "grey",
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 1", bg = "yellow",
          fg = "black", font = "10")
l3. pack (side = TOP, ipadx = 40)
l4 = Label (root, text = "CS!", bg = "orange"
          fg = "black", font = "10")
l4 . pack ( side = TOP, ipady = 50)
root.mainloop()
```

Practical No. 5

Topic : Gui Components.

Step1: Use the tkinter library for importing the features of the text widget

Step2: Create an object using the $Tk()

Step3: Create a variable using the widget label and Use the text method.

Step4: Use the tkinter - library mainloop() for triggering of the Corresponding above mention events

#2 :

Step1: Use the tkinter library for importing the features of the text widget.

Step2: Create a variable from the text method and position it on the parent window.

Step3: Use the pack() along with the object created from the text() and use the parameter.

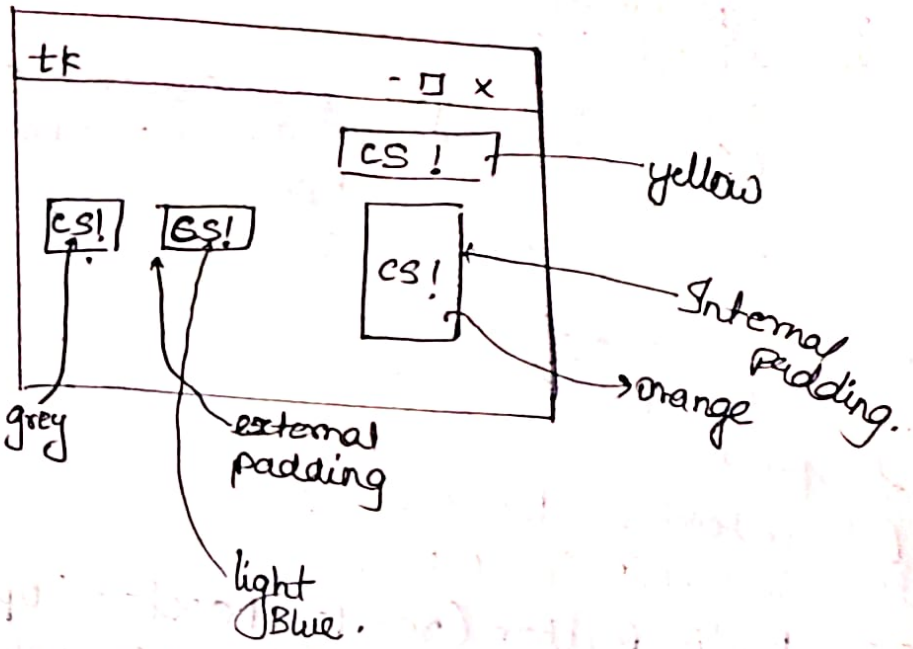
1) side = LEFT , padx = 20
2) side = LEFT , pady = 30
3) side = TOP , ipadx = 40
4) side = TOP , ipady = 50

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 be foreground fg and than use the pack() with a relevant padding attributes.

Output :



```
# Radio button
from Tkinter import*
root = Tk()
root.geometry("500x500")
def select():
    selection = "you just select" + str(var.get())
    l1 = label("text = selection, bg = "white",
               fg = "green")
    l1.pack(side = TOP)

var = StringVar()
l1 = Listbox()
l1.pack()
l1.insert(1,"List 1")
l2.insert(2,"List 2")
r1 = RadioButton(root, text = "option1",
                 variable = var, value = "option1",
                 command = select)

r1.pack(anchor = N)
r2 = Radiobutton(root, text = "option2", variable=v
                 value = "option2", command = sele

r2.pack(anchor = N)
root.mainloop()
```

Practical NO. 5(A)B)

Aim: Gui Components.

#1:

Step1: Import the relevant methods from the tkinter library create an object with the parent window.

Step2: Use the parent window object along with the geometry() declaring spafic pixel size of the parent windows.

Step3: Now define a function which tells the user about the given selection made from multiple option available.

Step4: Now define the parent window and define the option with Control variable

Step5: Now define the parentwindow and define the option with Control variable. Use the Listbox() and insert options on the parent window along with the pack() with specifying anchor attribute.

Step6: Create an object from radio button which will take following arguments & parentwindow Object, that text variable which will take the values object no 1, 2, 3 variable argument corresponding value &

Step6 : trigger the function declared.

Step7 : Now call the pack() for radio object so created and specify the argument using anchor attribute.

Step8: Finally make use of the mainloop () along with parent-object.

#2

Step1 : Import relevant methods from the ~~no~~ tkinter library.

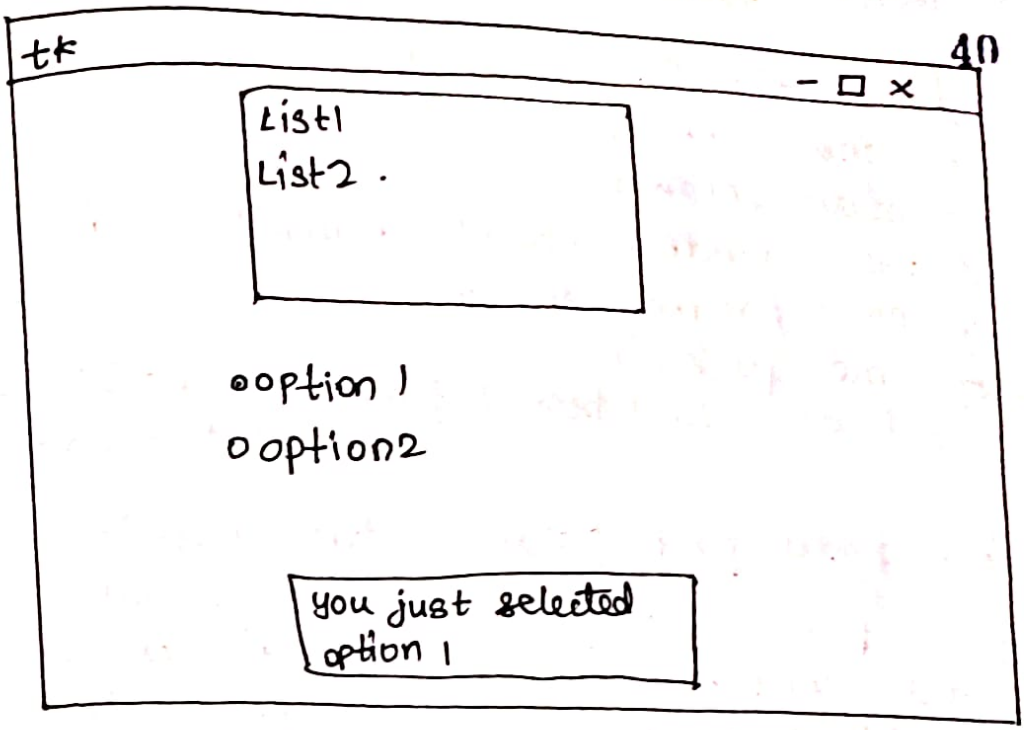
Step2: Create a parent object corresponding to the parent window.

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

Step4 : Create an object and use the Scrollbar ()

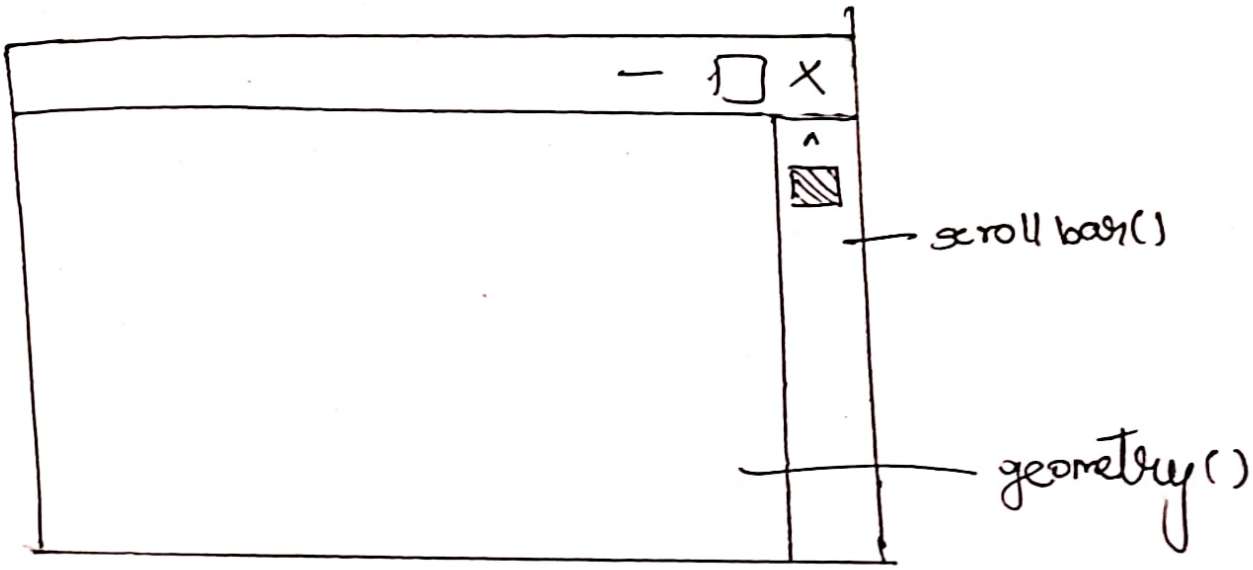
Step5: Use the pack() along with the scrollbar object with side and fill attributes.

Step6: Use the mainloop with the parent object.



#2

```
scroll bar()
From Tkinter import*
root =TK()
root .geometry ["500x 500")
s = scroll bar ()
s. : pack( side ="right", fill="y")
root . mainloop()
```



#3 using frame widget

```
from tkinter import *
window = Tk()
window.geometry("680x500")
Label(window, text="number:").pack()
frame = Frame(window)
frame.pack()
listNodes = Listbox(frame, width=20, height=20,
                    font=("Times New Roman", 10))
listNodes.pack(side="left", fill='y')
scrollbar = Scrollbar(frame, orient="vertical")
scroll.config(command=listnodes.yview)
scrollbar.pack(side="right", fill="y")
for x in range(100):
    listNodes.insert(END, str(x))
window.mainloop()
```

Output:

#3

Step1: Import the relevant libraries from the tkinter method.

Step2: Create an corresponding object of the parent window.

Step3: Use the geometry manager with pixel size (680 & 500) or any other suitable pixel value.

Step4: Use the label widget along with the parent object created and subsequently use the pack method.

Step5: Use the frame widget along with the parent object created and else the pack method

Step6: Use the listbox method along with the attributes like width, height, font etc create a listbox method object use pack() for the same.

Step7: Use the scrollbar() with an object use the attributes of vertical then Configure the same with object created from the scroll bar() and use pack()

Step8: Trigger the events using mainloop

#4

Step1: Import relevant methods from tkinter library.

Step2: Refine the object corresponding to parent window and define size of parent window in terms of no of pixels.

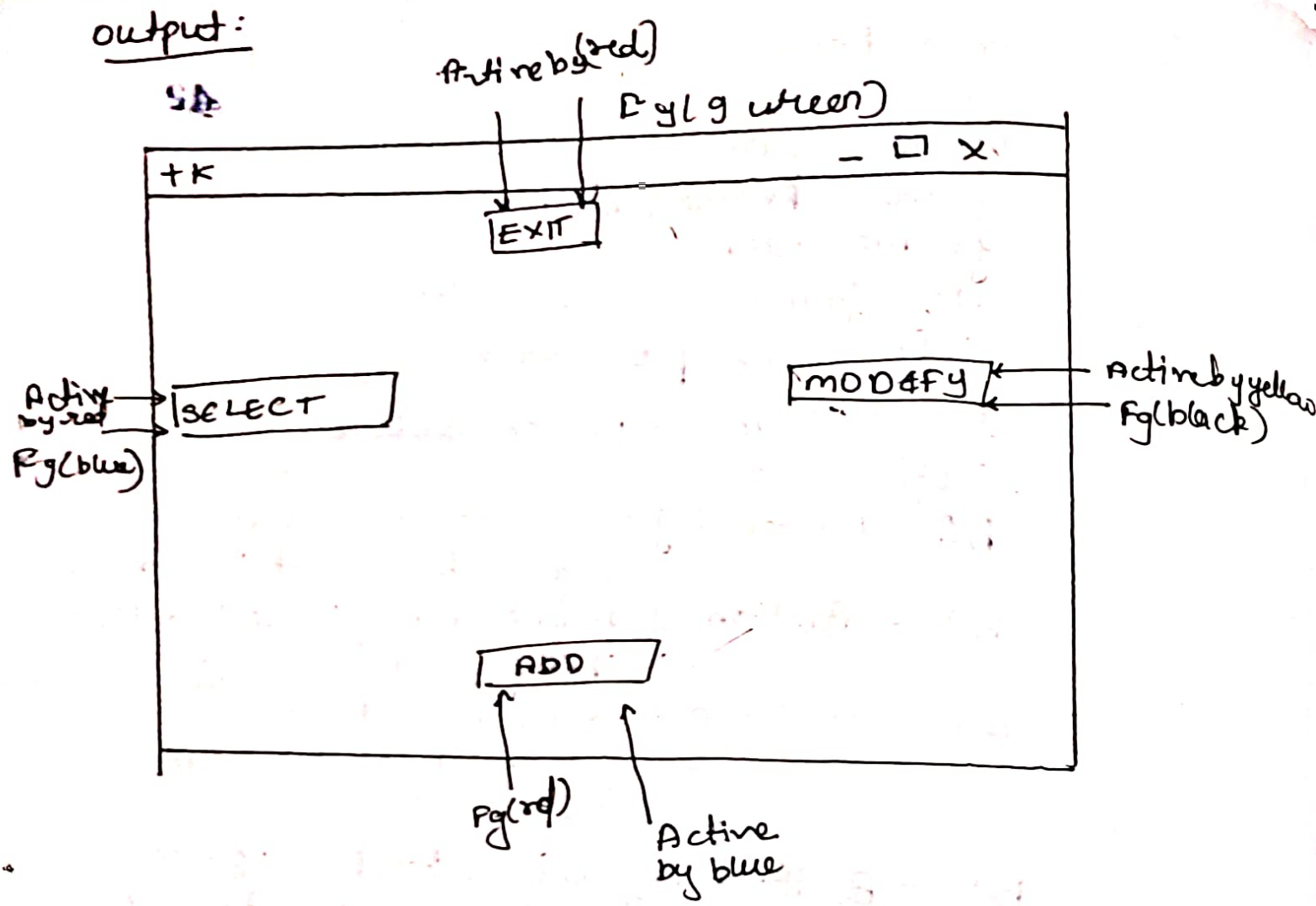
Step3: Now define the frame object from the method and place it on to the parent window.

Step4: Create another frame object termed as the left frame and put it on the parent window on its LEFT.

Step5: 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.

```
import*
window =TK()
window.geometry("680x500")
frame = Frame(window)
frame.pack()
left frame = Frame(window)
left frame.pack(side = "left")
right frame = Frame(window)
right frame.pack(side="right")
b1 = Button(frame, text="select", active
    background = "red", fg="blue")
b2 = Button(frame, text="select, active
    background = "red 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:



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

step 7: Similarly create the button object corresponding to the MODIFY operation put it init frame object on side="right"

step 8: Create another button object & place is on to the right frame & label the button as ADD.

step 9: Add another button & put it on the top of frame and label it as EXIT.

step 10: Use the pack() simultaneously for all the objects & finally use the mainloop ()

S.B

Practical 5(a) No 5(a)

Aim: GUI COMPONENTS

Step1: Import the relavant method from tkinter library

Step2: Import tkmessagebox

Step3: Define a parent window object along with the parent window.

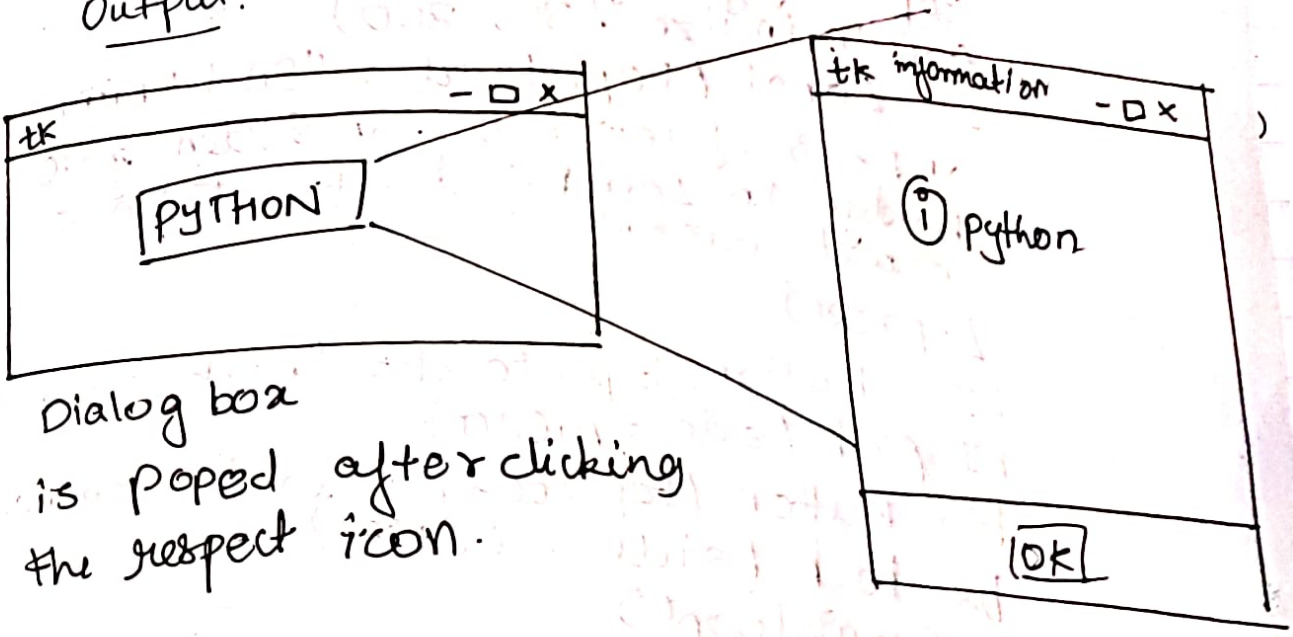
Step4: Define a function which will use tkmessagebox with showinfo method & along with info window attribute.

Step5: Declare a button with parent window object along with the command attribute

Step6: place the button widget onto the parent windaw and finally call mainloop() for triggering of the events called above.

```
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:



Dialog box is poped after clicking the respect icon.

14.

```
# multiple window
#Different button(Relief())
from Tkinter import*
root =Tk()
root.minsize(300,30

def main():
    top = Tk()
    top.config(bg ="black")
    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 =seco
    b1.pack(side =RIGHT)
    b2= Button(top, text = "exit", command =termin
    b2.Pack(side = LEFT)
    top.mainloop()
```

Step1: Import the relevant methods from the tkinter library along with parent window object declared

Step2: Use parent window object along with min size function for window size.

Step3: Define a function main, declare parent window object and use config() title (), mini size(), label() as well as button() and use pack() & mainloop() simultaneously.

Step4: Similarly define the function second and use the attributes accordingly.

Step5: Declare another function button along with parent object and declare button with attributes like FLAT, RIDGE, GROOVE, RABED, SUNKEN along with relief widget

Step6: Finally called mainloop() for event driven programming.

Practical No. 5(D)

AIM: GUI Components.

Step1: Import relevant methods from the tkinter library.

Step2: Create parent window object and use the config method along with background color attribute specified.

Step3: define a function info use a listbox finish with the messagebox widget or which will display a message and i'e a warning message

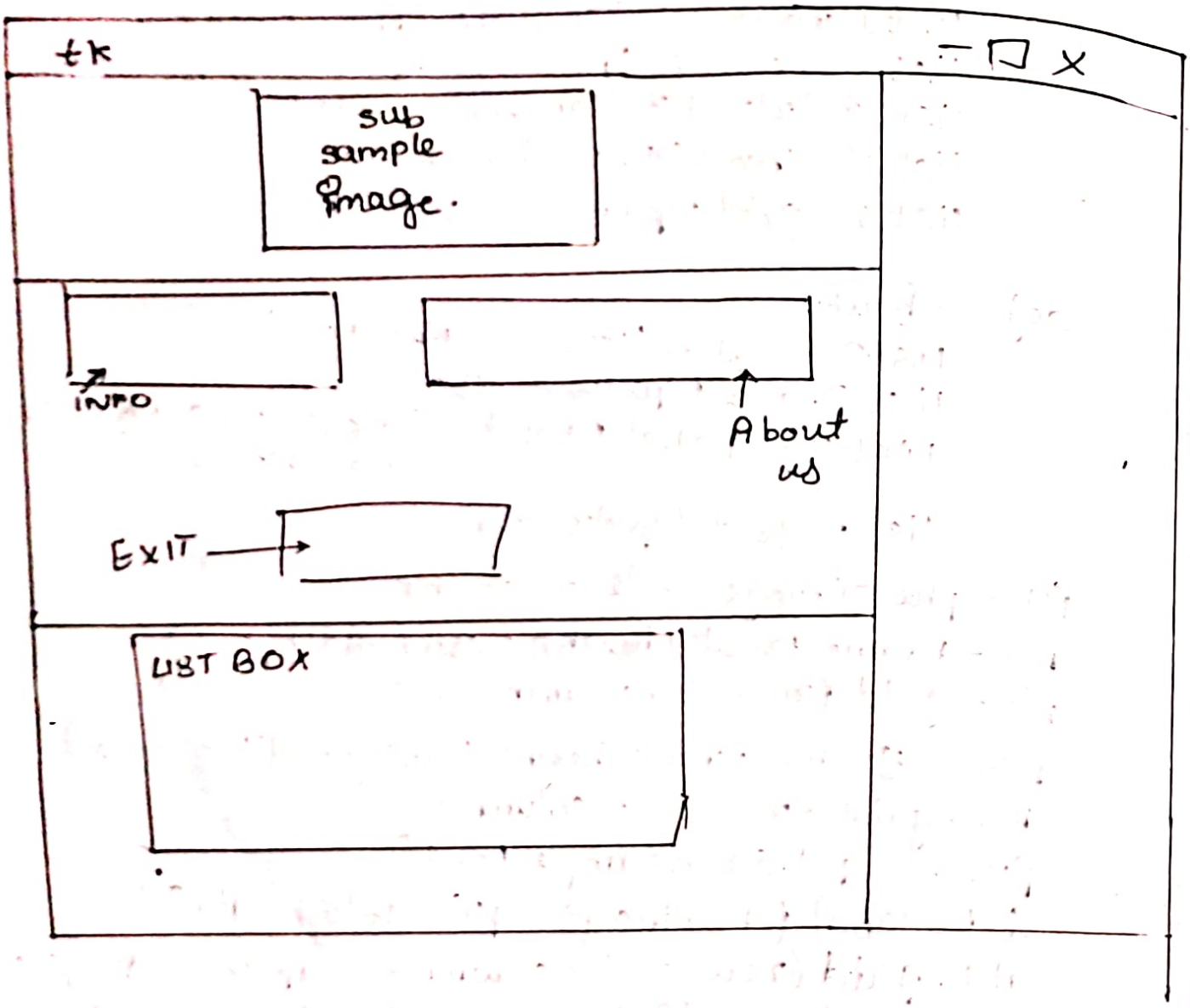
Step 4: define a function info use a listbox widget along with a object of the same. use the listbox object along with insert method and insert the same and finally use the geuid() with ipadx attribute.

Step5: Define a function about us with label widget and text attribute and subsequently use the geuid()

```
root = TK()
root = config( bg= "grey")
def finish (·) :
    messagebox . askokcancel ("Warning ", This will
                                end the program!!)
    quit ()
def info ():
    list 1 = List box ()
    list 1.insert ( 1, "co.name : Apple ")
    list 1 · insert ( 2, "products : iphone ")
    list 1 insert ( 3, "language : swift ")
    list 1 · insert (4, "os : IOS" )
    list 1 · grid (ipadx = 30)

def aboutus():
    list2 = label (text = "About us")
    list 2. grid (ipadx = 30)
    list 3 = label ( text = "steve jobs theater
                              march 2020")
    list 3. grid (ipadx = 24)
P1 = photoimage (file = "download · gif")
F1 = Frame (root , height = 35 , width = 5)
F1 · grid (row = 1 , column = 0)
F2 = Frame (root , height = 250 , width = 500)
F 2. geiid (row = 1 , column = 1)
P2. = P1.Subsample (5, 4)
l 1 = label ( F1 , image = p2 , relief = FLAT)
l 1 . geiid (row = 1 , column = 0, ipadx = 20 pady = 15)
l 2 = label ( f2 , image = P1, relief = SUNKEN )
l 2 · geiid ( padx = 25 , pady = 10 )
b1 = Button (f1, text = "Information", relief = SUNKEN
        command = info)
```

```
b1 grid (row1, column = 0)
b2 = Button (f1, text = "Aboutus", relief = SUNKEN
            command = aboutus)
b2 grid (row=1, column = 2, padx = 5)
b3 = button (f1, text = "Exit", relief = RAISED,
            command = finish)
b3 grid (row = 2, column = 1, ipadx = 15)
root mainloop()
```



step6: Use the photoimage widget with file & filename with gif attr.

step7: Create a frame object along with the frame object (→ parent window object along height width specific subsequently

Practical No. 6

Aim: WAP to draw a human face with GUI

Algorithm:

Step 1: Import relevent methods from tkinter library.

Step 2: Create a object corresponding to the parent window from TK()

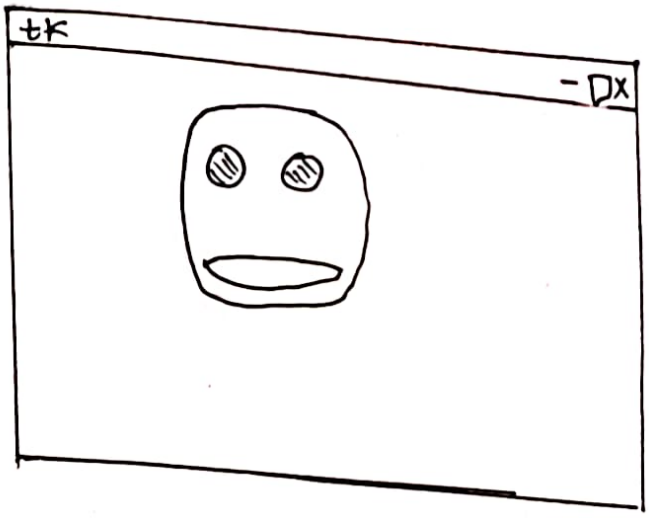
Step 3: Create an obj from canvas() place it onto parent window along with height & width.

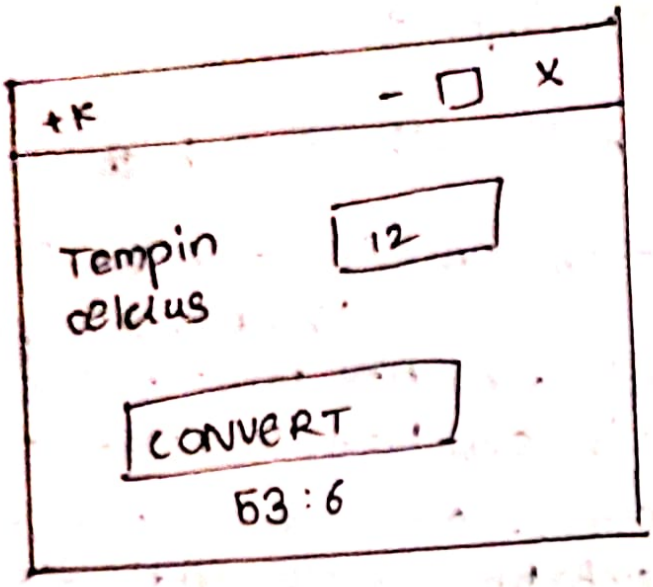
Step 4: Now use pack() for positions of widget onto the parent window.

Step 5: Now Create an object face & use object Create_oval() with 10-ordinates 50, 50, 350, 350 & outline = 'black', fill = 'yellow' as an attribute & to create fall.

Step 6: Now Create an object fa & against oly create Oval() with appropriate coordinates along with fill as attribute to Create left eye.

```
from tkinter import*
root = Tk()
c = canvas (root, width = 500, height =500 )
c.pack ()
face = c.create_oval (50,50, 350, 350, outline=
                      "black", fill = yellow")
eye1 = c.create_oval (125, 125, 175, 175, Fill="black")
eye2 = c.create_oval (125, 125, 275, 175, Fill="black")
mouth = C.Create_arc (125, 225, 275 ,275, start=0
        extent = -180, width = 5, Fill="red"
root.
root.mainloop ()
```





step7: Now repeat the same step to Create right eye.

step8: Create an object month & use object Create_arc() with appropriate Co-ordinates, start=0, extent=-180 & fill="red", width=5 as attributes to create month.

step9: Finally use the mainloop()

#2

~~Coding~~ [Conversion of Farenheit from ℉ to Celcius]

Step1: Import elevant method from Tkinter library.

Step2: Make a parent window & decline a function for conversion of one scale to another.

Step3: Define farenheit & celcius & give them variable which hold value.

Step4: Use label width & place it on parent window also use text to specify the type of connection.

Step 5: Use guid () & specify row & column attributes.

Step 6: Use entry widget & and use text variable for the guid ()

Step 7: Use button widget & guid () specifying row & column.

Step8: Use mainloop() to terminate prog.

#2 Coding [Conversion of farenheit ~~from~~ to celcius]

```
From tkinter import*
window=TK()
Farenheit = Double Varl)
Farenheit .set(32.0)
def convert (celcius):
        Farenheit.set ((9.0/5.0)* celcius +32)
        L+ = label (window ,test ="Temp in celcius")
        L1 .grid (row=0, column=0)
        e = Entry(window, teset = "Temp in celcius")
        E: grid (row = 0, column =0)
        celcius =Int var()
        L2 = label (window, text variable= Farenhit)
        L2.grid (row =2, column =0, column span=2)
        B = Button (window, text = "calculate",
                    command = Lambda. convert(celcius,
                                              get)
        B.grid (row=1, column = 0, colum span=2)
window.mainloop ().
```

Coding :-

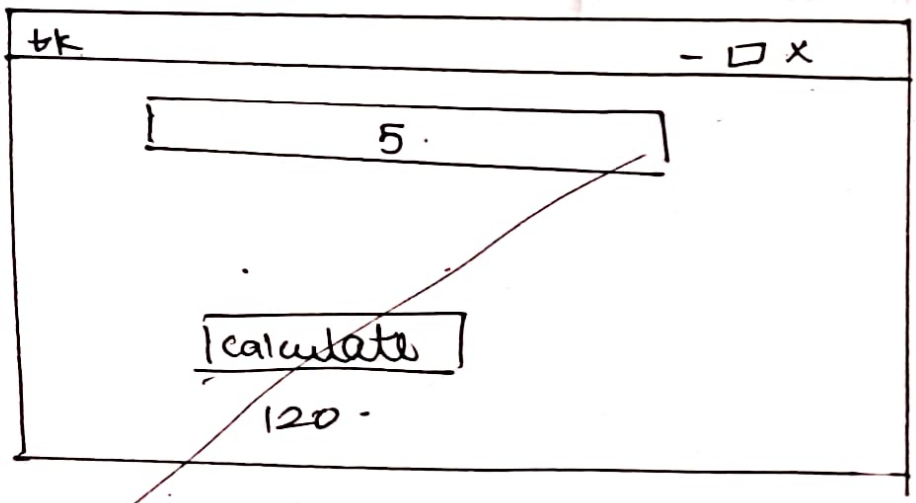
```
From tkinter import*
def Factorial(n):
    if n==0 or n==1:
        return 1
    else:
        return n* factorial (n-1)

def calculate():
    result = factorial (int (entry_tex.get()))
    info.config (text = result)

root =Tk()
entry_test = Entry (root)
entry_test.pack()
bt = Button (root, text = "Calculate", command=calculate)

bt.pack()
info = label (root, text = "Factorial")
info =pack()
root.mainloop()
```

Output:



Practical No. 7

Aim: WAP to find factorial of no. & use arithmatic operation on 2 no.s using GUI.

Algorithm:

Step1: Import relevant method from tkinter library.

Step2: Now Define a function for factorial to calculate factorial using recursive function

Step3: Define another function calculate to call factorial function.

Step4: Now create an obj with entry () and use pack() to position the obj.

Step5: Now create a button() along with command after & create a label() to show output

Step6: finally use the mainloop().

*2 CODE 2: ARITHEMETIC OPEN ON 2NOS.

Step1: Import Relevant methods from tkinter. library.

Step2: Now Create an obj. corresponding to parent window

Step3: Now define a function calculate to carry out operation.

Step4: Now create obj with label() as Num 1 & Num 2 and use grid() to place it onto parent window.

Step5: Create objects with entry() to take input from user()

Step6: Now initialize it as Integers using ~~Insert()~~ IntVar().

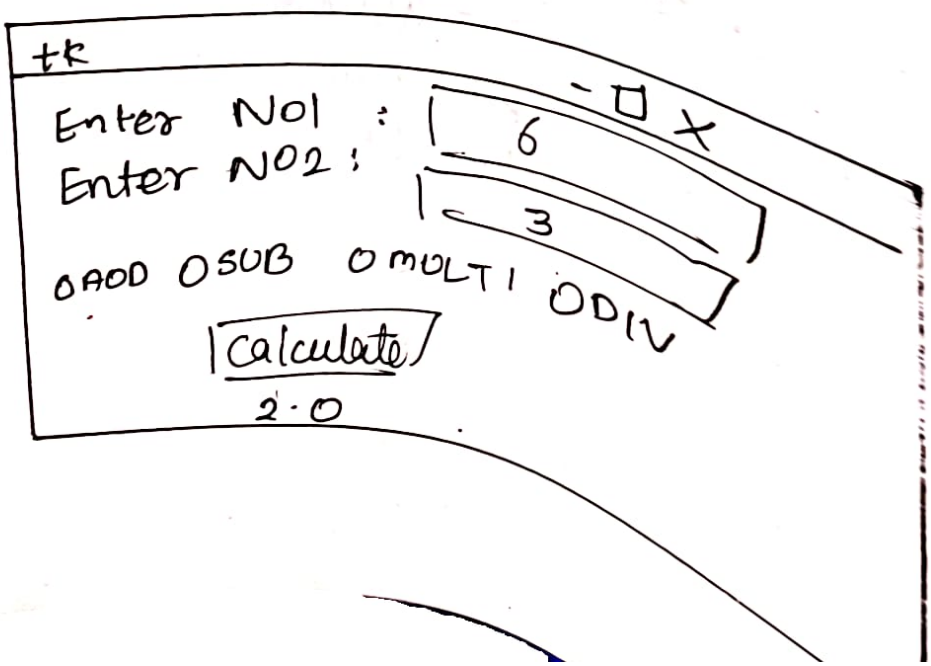
Step 7: Now Create 4 objects with Radio button() to choose any one of Arithmetic operators & use grid()

Coding: #2

```
From tkinter import*
def calculate():
    if int(u.get()) == 1:
        res = int(e1.get()) + int(e2.get())
        l3.config(text = res)
    elif int(v.get()) == 2:
        res = int(e1.get()) * int(e2.get())
        l3.config(text = res)
    else:
        res = int(e1.get()) / int(e2.get())
        l3.config(text = res)
root = TK()
l1 = Label(root, text = "Enter a number")
l1.grid(row=0, column=0)
e1 = Entry(root)
e1.grid(row=0, column=1)
l2 = Label(root, text = "Enter 2ND")
l2.grid(row=1, column=0)
e2 = entry(root)
e2.grid(row=1, column=1)
v = Intvar()
```

```
v1=Radiobutton(root, text="Add", variable=v,
               value=1)
v1.grid(row=2, column=0)
v2=Radiobutton(root, text="sub",
               variable value=2)
v2.grid(row=2, column=1)
v3=Radiobutton(root, text="Multiply", variable=v,
               value=3)
~~v4=grid~~
v3.grid(row=2, column=2)
v4=Radiobutton(root, text="Div", variable=v,
               value=4)
v4.grid(row=2, column=3)
B=Button(root, text="Calculate", comm and
         calculate)
B.grid(row=3, column=1, Column space=2)
L3=Label(root)
L3.grid(row=4, column=1)
root.mainloop()
```

Output:



Step8 : Now create an object with button() along with command attribute to carry out the arithemetic operator of user's choice.

Step9 : Now create a object with label() to show input().

Step10 : Finally use the mainloop()

Practical No. 8

Aim: Demonstrate the use of socket module and server, client program.

Algorithm:

Step1: Import the socket module relevant methods.

Step2: Define a function as Server - program to ~~get get~~ to get host name.

Step3: Now get value for port variable to initialise port no above to24.

Step4: Use Socket () to get instance.

Step5: Now use bind () function to be bind host address & port together to configure how many client the server can list simultaneously.

Step6: Use Accept() to accept new connection.

Step7: Now print address.

Step8: Use while loop as due to view data stream.

```
import socket
def server program:
    host = socket .get host name()
    host = 5000
    Server -socket = socket .socket()
    server -socket .bind((host , post))

server_socket . listen (2)
conn, address = server - socket .accept()
print ("Connection from : " + SH(address))
while (True) :
        data = conn . rcu (1020) . decode()
        if not data :
            break
        print ("From connected user : " +SH(data ))
data = input (" ⟶ ")
conn.send (data . encode ())
conn. close ()
```

Output:

```
Python 36 socket . server . py
connection for ('127.0.0.1', 57822)
From connected use : HI
- >HELLO
FROM CONNECTED USE : HOW ARE YOU?
⟶ HELLO GOOD
From connected User : Awesome!
→ ok then , Bye.!
```

code 2

```
import socket
def client_prog:-
host = socket.gethostname()
host = 5000
client_socket = socket.socket()
client-socket.connect((host, Port))
message = input("—>")
while message.lower().strip() != 'bye':
    client_socket.send(message.encode())
    data = client_socket.rcv(1024).decode()
Print("Recieved from server.", data)
message = input("—>")
client_socket.close()
```

output:

```
Python 3.6 socket-client.py
→ Hi
Received from server : Hello
→ How are you?
→ Received from server : Good
→ Awesome
Recieved from server : Ok then Bye
→ bye.
```

Step 9: Now close the program.

#2 For socket client program.

Algorithm:

Step1: Impact socket module to import methods that relevant.

Step2: Define a function client program get hwrt host name & give host a value 5000

Step3: Now again initiate by using socket.socket()

Step4: Use connect() to connect to the server

Step5: Now take the input.

Step6: Use while loop to send a less age.

Step7: Show the data

Step8: Close the program.

Practical No. 9

Aim: Demonstrate the use of database Connectivity.

Algorithm:

Step1: Import sq lite 3 module to import relevant methods

Step2: Now initialize variable conn to connect by using connect() to a new data base using extension.db

Step3: Now initialize a variable to connect to cursor()

Step4: Now use are execute() to Create a table insert values into tables & use DML, DDL, Statements to manipulate the data in this database.

Step5: Use fetch all() to show the output.

Step6: Use Commit to save all changes

code in shell environment

```
>>> import sqlite3
>>> conn = sqlite3.connect("student.db")
>>> cur = conn.cursor()
>>> cur.execute('Create table student (roll_no int(5)),
    primary key, name varchar (50) not null, class
    varchar (50), dob date)')
<sqlite3.cursor object at 0x0322EBED>
>>> cur.execute (insert into students value(101,'Abhay',
    "vasai", FYCS", 24/01/2002"))
<sqlite3.cursor object at 0x0322EBED>
>>> cur.execute (insert into students values(102,'Veeru',
    'Boriwali', FYCS", 02/08/2002"))
<sqlite3 cursor object at 0x0322EBED>
>>> cur.execute ("select* from student")
<sqlite3.cursor object at 0x0322EBED>
>>> cur.fetchall()
[(101,'Abhay','vasai','FYCS','24/01/2002')
 (102,"Veeru",'Boriwali','FYCS','02/08/2001')]
```

```
>>> cur.execute("update student set
                 where rollno=101")
<sqlite3.Cursor object at 0x0322EBED>
>>> cur.fetchall()
[(101, 'Abhay', 'Vasai', 'FYCS', '13/08/2001')]
>>> cur.execute('commit')
<sqlite.Cursor object at 0x0322EBED>
>>> cur.close()
```

Step 7: Use down() to terminate the program.