```
file obj. open ("abc text","co")
file.obj. write ("Dance tunes"+"\n")
fileobj. write ("bangarang in revolution in getlow ")
fileobj. close ( )
file = open ("abc.text","r")
# read
str= fileobj. read ( )
print ("The output of read method:",str)
fileobj. close( )
>>>("The output of readmethod:","Dance tunes in bangarang in revolution
                                in getlow \n')
# readline ( )
file obj. open ("abc. txt","r")
str 2 = fileobj. readline ( )
print ("output:',str2)
fileobj. close ( )
>>> ('output :','Dance tunes in')
# readlines ( )
file obj= open ("abc.txt ","r")
str3 = file obj. readlines ( )
print ("output:", str 3)
file obj. close ( )
>>> ('output:',['Dance tunes in bangrang in revolution in getlow']
# file attributes
a= file obj. name
print ("name of file (name attribute): "), a)
>>>(name of file [name attribute]:'abc.txt')
b= file obj. closed
print ("(close) attribute :", b)
>>> ("(close)attributes :','true')
```


Practical No. 1

Objective :- Demonstrate the use of different file ac
modes different attributes read method

Step1:
Create a file object using open method and u
access mode followed by writing some contents
file and than closing the file

Step2:
Now open the file in read mode and than
readline ( ) and readlines ( ) and store the o
variable and finally display the contents of v

Step3:
Now use the file object for finding the nam
file, the file mode in which is opened whe
file is still open or close and finally the
softspace attributes

Step 4:
Now open the fileobj in write mode write some another content close subsequently. Then again open the file obj in 'w+' mode that is the update mode and write content

Step 5:
Open file obj in read mode display the update written content and close open again in 'r+' mode with parameter passed and display the output subsequently

Step 6:
Now open file obj in append mode open write method write contents close the fileobj again open the file obj in read mode and display the appending output


```
c= fileobj.mode
print ("filemode",c)
>>>('File mode', 'r')
d= fileobj.softspace
print ("softspace:" d)
>>> ('softspace:', 0)

# 'w+' mode
fileobj.open("abc.txt","w+")
fileobj.write ("skrillex")
file obj. close ( )

# r+ mode
fileobj = open ["abc.txt","r+"]
s1 = fileobj.read (5)
Print ["output of r+", s1)
file obj. close ( )
>>>(output of r+', 'skrill')
# append mode
fileobj = open ("abc.txt","a")
fileobj.write (Ragga bomb)
fileobj.close ( )
file obj = open ("abc.txt","a")
s3=file obj.read ( )
print ("output of append mode:", s3)
fileobj. close ( )
>>>(output of append mode:', 'skrillex Ragga bom
```

```
# write mode
fileobj= open("abc
fileobj.write ("S
fileobj.close( )

# read mode
fileobj = open (
s = fileobj.read
print ("output
>>>output of re
```

```
# tell()
file obj = open ("abc.txt","r")
Pos = fileobj.tell()
print ["tell():", Pos)
fileobj.close()
>>> ["tell():", 0]

# seek
fileobj = open ("abc.txt","r")
st = fileobj.seek(0,0)
print (seek (0,0) is :", st ).
fileobj.close ( )
>>> [seek (0,0) is : ', none]
file obj = open ('abc.txt', "r")
pos 2 = fileobj.seek (0,1)
st 2 = fileobj.readline (8)
print close ( )
>>> ['seek (0,1) is : ', 'skrillex')
# find length of different linear exist lines
fileobj = open ("abc.txt","r")
stat = fileobj.readlines ( )
print ("output","stat")
for line in stat
    print (len(line))
fileobj.close( )
>>> ('output:' ['skrillex Ragga bomb'])
>>> 19
```


Step 7:

Open the fileobj in read mode, declare a varia and perform fileobject dot tellmethod and store th output consequently in variable

Step 8:

Use the seek method with the argument opening the file obj in read mode and closi subsequently

Step 9:

Open file obj with read mode also use the method and store the output consequently in the same for counting the length use the statement and display the length

Practical no. 2

Objective :- Iterators

Step1 :- Create a tuple with elements that we need to iterate using the iter and next method number of time we use the iter and next method we will get the next iterating element in the tuple Display the same

Step2 :- The similar output can be obtained by using for conditional statement. An iterable variable is to be declared in for loop which will iterate

Step3 :- Define a function name square with a parameter which will optain output of square value of the given number. In similar fashion declare cube to get the values raised 3 and return the same

Step4 : Call the declared function using function call

```
# iter() and next()
my tuple 1 = ("banana","orange","apple")
my iter 1 = iter (mytuple)
print (next (myiter1))
myiter2 = iter (my tuple)
print (next (myiter2))
myiter 3 = iter (my tuple)
print (next (my iter 3))
>>> banana
    orange
    apple

# for loop
mytuple 1 = ("kevin", "stuart", "bob")
for x in mytuple1
  print (x)
>>> kevin
    stuart
    bob

# square and cube
def square (x):
  y = x * x
  return y
def cube (X):
  Z = X * X * X
  return Z
unit 1 = [square, cube]
```

```
for r range(5):
    value = list(map(lambda x: x(r), functn))
    print [value]
```

8. >>> [0,0]
[1,1]
[4,8]
[9,07]

9. [16,64]

```
# map[ ]
list num=[0,4,5,7,9,11,13,15,20,19,25]
list num=list(map(lambda x: x%5, list num))
print (list num)
def even (x):
    if (x%2==0)
        return"EVEN"
    else:
        return"ODD"
list (map (even, list num))
```

```
#odd numbers
class odd:
    def__iter__(self):
        self.num1
        return self
    def__next__(self):
        num.self=num
        num.self+=2
    def__next__(self):
        num=self.num
        self.num+=2
        return num
```


Step 5:- Using for conditional statement specifying and ra use the list type casting with map method declare a 'lambda' i.e annonyms function and print the sam

Step 6:- Declare a list num variable and declare some element then use the map method with help of lamb function give two argument display the output

Step 7:- define a function given with a parameter then using conditional statement do check whether number is even, odd & return respectively

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

Step 9: Now use the next() and define the logic for d odd values

Step 10: define an object of a class

Step 11: Accept an number from the user till which use wants to display the odd number

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

» Enter a number: 15

1
3
5
7
9
11
13

20/12/17

```
# arithematic error
while True
    try:
        x = int (input("Enter class")
        break
    except ValueError
        print ("Enter numeric char:")
```

Output:

Enter class tycs
Enter numeric char:
Enter class :13

```
# environment error
try:
    fo = open("abc.txt","w")
    fo.write ("Aachal")
except IOError
    print ("error writing on the file")
else:
    print ("operation carried out successful")
    fo.close()
```

Output:
Operation carried out successful

Practical - 03

Aim: To demonstrate exceplion handling

1. Write a program using the exceplion method of the nature Arithmatic error

Step1: Use the try block and except the input using the input method and convert it into the integers datat and subsequent terminate the block
Step2: Use the except block with the exception name as va error and display the appropriate message if the sus code is part of the try block

2. Write a program for accepting the file in a given mode and use the environment error as an e for given input

Step1: Within the try block open the file using th write mode and write some content on the fil

Step2: Use the except block with IO error and displa message regarding missing of the file or incom at the mode use the else block to display to display a message that the operation is ca out respectfully

3. WAP using the assert () to check if the last element are empty

Step1: Define a function which accepts an argument and check using the assert statement whether the test is empty list and accordingly return the message

Step2: Close the function and in the body of program and define certain element in the list and take some appropriate action

4. WAP to check the range of the age at the student in given class and if the age do not fall in given range else the value error exception otherwise return the valid no.

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

Step4: Use exept with valueError and print the message not a valid range


```
# assert error

def assert_(n):
    assert (len(n) == 0)
    print ("list is empty")
var1 = [ ]
print (assert_(var1))
```

Output:

list is empty
None

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

Output:

Enter your age :18

```
# code 1
import re
string = "Hello 1234 abc 4567"
result = re.findall ("\d", string)
result = re.findall ("\d", string)
print (result)
print (result)
```

```
# Output
>>> ['1234','4567']
>>> ['Hello', 'abc']
```

```
# code 2
import re
string = "Python is an important language")
result = re.search ["\A python", string]
print (result)
if result:
    print (' match found")
else:
    print ("match not found")
```

```
# output:
>>> rematch object span = (0,6)
        match = Python
>> match not found
```


Practical No. 4

Aim: Demonstrate the use of regular expression

Theory :- Regular expression represents the sequence of character which is mainly used for finding & replacing the given pattern in a string & for this we import remodules and common usage of regular expression
It involves following function

Algorithm :-

Step1: Now apply string & pattern in find all () and displ the output

Step2: \d is used for matching all decimal digits whe it is used to match non - decimal digit

- find the match string at the begining of given sequen

Step1: Import remodule and apply a string

Step2: Use search() with "\A Python" & string as a paramet

Step3: Now display the output

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

- Check whether the given mobile number with 8 or 9 and the total length is 10

Step1: Import the re module and apply a string of mob

Step 2: Now use for conditional statement to find if s starts with 8 or 9

Step 3: Use if condition statement to know whether we a match or not if we have use group() to the output and if we don't display incorrect mobil

- Extracting a word from given string along with characters in between subsequence and word with space charater

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 and "\wt" to extract wordwithout space

Step 4: Now display the output

Code 3:

```
import re
li ['9869729127','8540827373','8896822668','9082428253')
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 number
    9896271972
    Correct mobile number
    8869286286
```

code 4:

```
import re
string = "Python is important"
result1 = re.findall ("\wt",string)
result 2= re.findall ("\wt,string)
print (result1)
print (result 2)
```

code 5:

```
import re:
  string = "Python is important"
  result = re.findall ("n\w+", string)
  result1 = re.findall ("\w+$", string)
  print (result)
  print (result1)

# output
>>> ["Python"]
>>> ["Important"]
```


Write a regular expression for extracting first and last word

Algorithm.

Step 1: Import re module and apply a string

Step 2: Use findall () in which use "\wt" as one parame to find first word of string then are "\w+$" as parame to find last word of string

Step 3: Display the output

Practical No - 5

Aim:- GUI components

Step 1: Use the tkinter library for import the features of the text widget

Step 2: Create an object using the TK()

Step 3: Create a variable using the widget label and use the text method

Step 4: Use the mainloop () for of the corresponding about mention events

2:

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

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

Step 3: Use the pack () along with the object created from the text () and use the parameter

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


```
# Creation of parent window
from tkinter import *
root = TK()
l = label (root , text = "python")
l.pack ()
root , mainloop ()
```

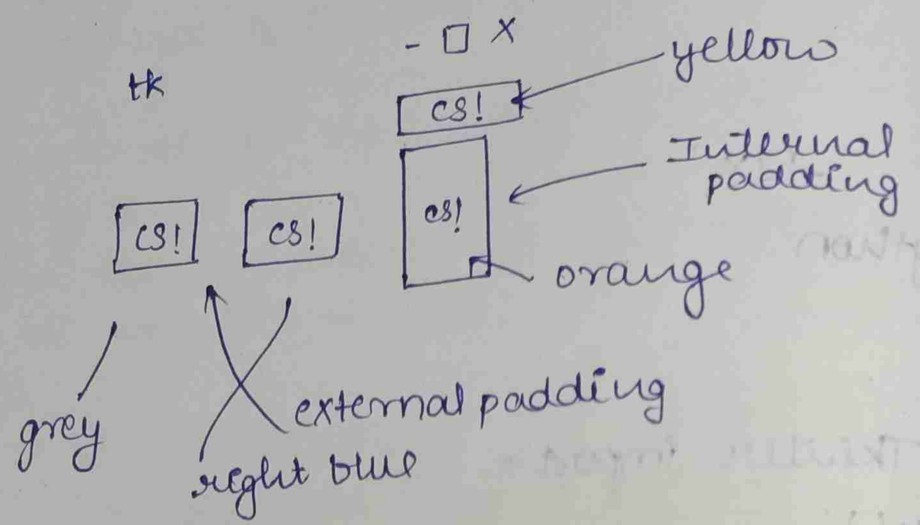
Output:

tk -□x

python

```
# 2:
from Tkinter import *
root TK()
l = label (root , text = "python")
l. pack ( )
l1 = label (root, text = "CS!", fg="grey")
    fg = "flack", fout = "10")
l2 = label (root, text = 'CS!", fg = "light blue",
    fg = " flack", font "20")
l2. pack (side = LEFT, pack up = "30")
l3. label (root = text = "CS!", fg = "yellow",
    fg = "flack", font = "10")
l3. pack (side = TOP, ipadx = "40"
```

```
lg = label (root, text = "cs!", fg = "orange"
      fg = "flack", font = "0")
lg.pack (side = TOP. ipady = 50)
root.mainloop ()
```



Step4: Use the mainloop() for the triggering of the corresponding event

Step 5: Now repeat above steps with label() which take the following arguments
1. Name of the parent window
2. Text attributes which defines the string
3. The background color (fg)
4. The foreground and than use the pack() u
a relevant padding attributes

Practical 5 (B)

Aim: GUI components

#1

Step 1: Import the relevant method from the tkinter library create an object with the parent window

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

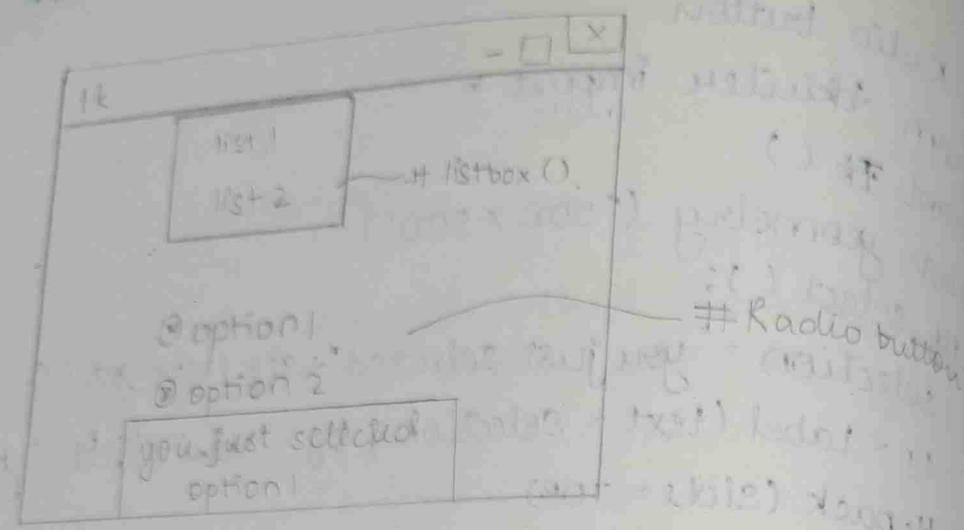
Step 3: Now define a function which tell the user about the given selection made from multiple option of variable

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

Step 5: Use the list box() and insert the option on the parent window along with the pack() specifying anchor attributes

Step 6: Create an object from radio object which will take following argument

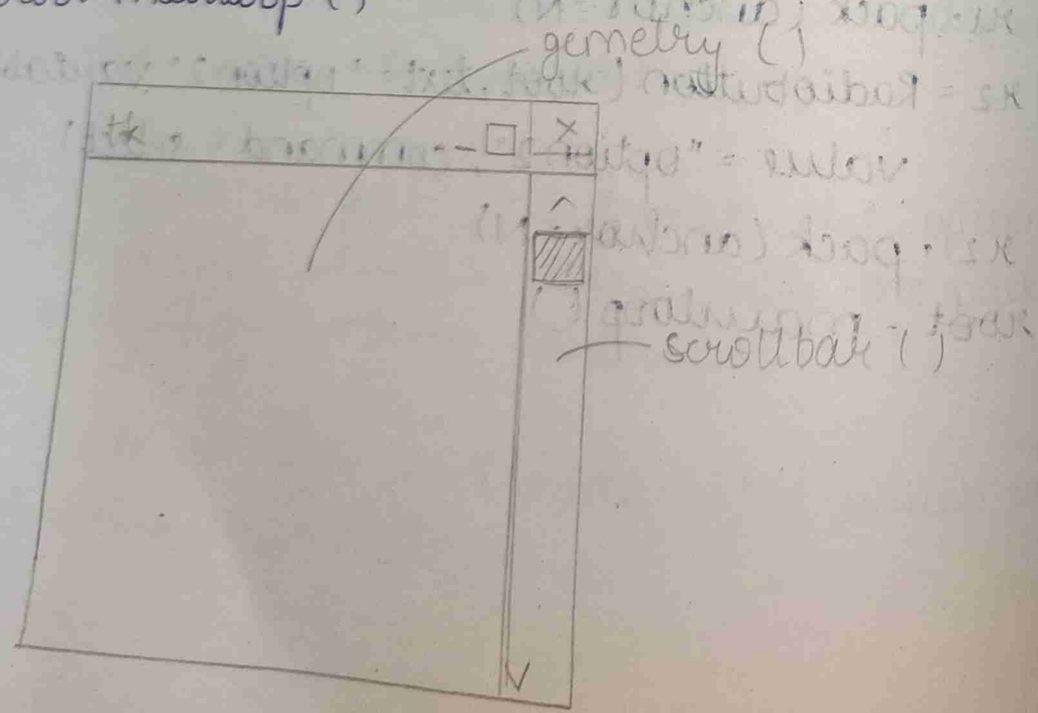

```
# Radio button
from tkinter import *
root Tk()
root. geometry ("500 x 500")
def select();
    selection = "you just selected"+str(var.get())
    L1 = label (text = selection, bg = "white", fg = "green")
    H. pack (side = TOP)
var . string var()
    L1 = list box()
    L1 . insert (1,"list1")
    L2 . insert (2,"list2")
    l1. pack (anchor = N)
      r1 = Radiobutton (root, text = "option1", variable = v
          value = "option1", command = select)
      r1. pack (anchor = N)
      r2 = Radiobutton (root, text = "option 2" variable
          value = "option 2", command = select)
      r2 . pack (anchor = N)
root = mainloop()
```



2

scroll bar ()

```
from tkinter import *
root = Tk()
root.geometry ("500X 500")
s= scrollbar
s.pack (scale=" right", fill="y")
root.mainloop ()
```



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

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

2

Step 1: Import relevant method from the tkinter binary

Step 2: Create a parent object corresponding to the parent window

Step 3: Use the geometry () for laying of the wind

Step 4: Create an object use the scrollbar ()

Step 5: Use the pack () along with the scrollbar object scan and fill attributes

Step 6: Use the mainloop with parent object

#3.

Step 1: Import the relevant libraries from the tk method

Step 2: Create an corresponding object of the parent window

Step 3: Use the geometry manager with pixel size (680x500) or any other suitable pixel value

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

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

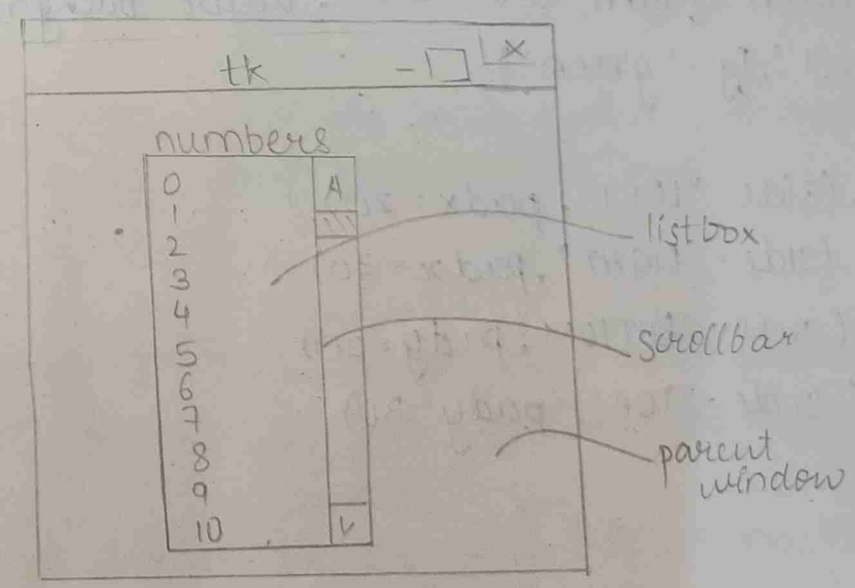
Step 6: Use the listbox method along with the attributes like width, height, font do create a list for box method object use pack() for the same

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

Step 8:- Trigger, the event using mainloop()

Using frame widgets

```
from tkinter widgets
  window = Tk()
window.geometry("680x500")
label(window, text = "numbers;")
frame =(Frame window)
  frame.pack()
list nodes = listbox(frame, width = 20, height = 10,
        font = (" Times New Roman", 10
  list nodes.pack (side = "left", fill = "y")
  scrollbar = scrollbar(frame = "vertical")
  scrollbar.config (command = lIstNodes.yview)
  scrollbar.pack (side = "right", fill = "y")
  for an range (100):
    List nodes insert (END, str (x))
  window mainloop
```



```
from tkinter import
window = Tk()
window.geometry ("680 x 500")
frame = Frame (window)
frame.pack()
list frame = frame (window)
left frame.pack (scale = "left")
rightframe = frame (window)
right frame.pack (scale = "right")
f1 = Button (frame, txt = "select", active background
    = "red", fg = "blue")
f2 = Button (frame, txt = "modify", active background
    = "yellow", fg = "black")
f3 = Button (frame, txt = "ADD", active background
    = "blue", fg = "red")
f4 = Button (fram, txt = "EXIT", active background
    "red", fg = "green")

f1.pack (side = "LEFT", padx = 20)
f2.pack (side = "RIGHT", padx = 30)
f3.pack (side = "BOTTOM", pady = 20)
f4.pack (side = "TOP", pady = 30)
```


#4

Step1: Import relevant method from tkinter library

Step2: Define the object corresponding to parent window & define the size of parent window in term of no. of pixels

Step3: Now defines the frame object from the method & place it on the parent window

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

Step5: Similarly define the RIGHT frame and subsequently defi the button object placed onto the given frame

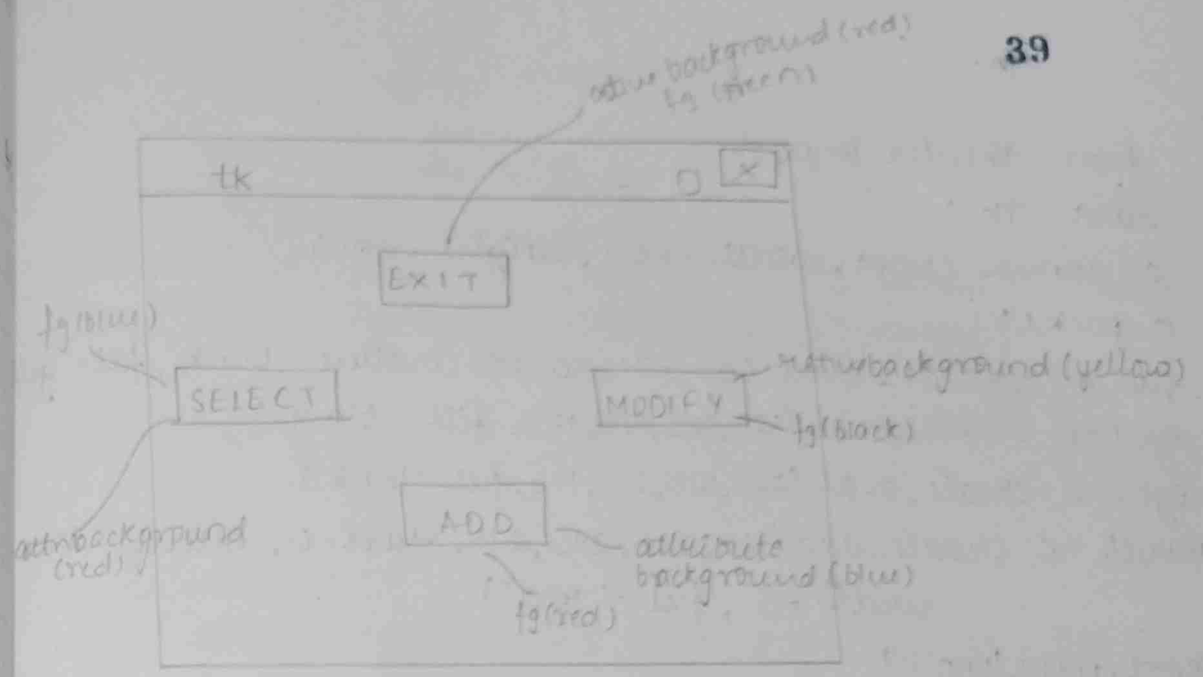
Step6: Now use the pack() along with the scale attr

Step7: Similarly create the buttons object corresponding to the (modify operations put it into frame object side = "RIGHT")

Step8: Create another button object and place it on th RIGHT frame & label the button as ADD

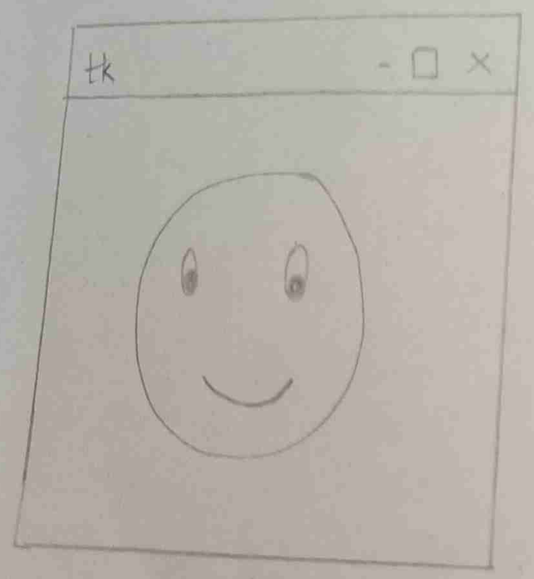
Step9: Add another button and fix is on the top frame and label it as EXIT

Step10: Use the pack() simultaneously for all object and finally use the mainloop()



```
from tkinter import
root = Tk()
c = canvas (root, width = 500 , height = 500)
c.pack()
face = c. create_oval (50,50,350,350 , outline = "black")
eye 1 = c. create_oval (125,125,175,175, fill = "black")
eye 2 = c. create_oval (225,125,275,175, fill = "black")
mouth = c. create_arc (125,225,275,275, start = 0, extent
                width = 5 , fill = "red")
root.mainloop()
```

output



Practical No: 6

Aim :- Demonstrate the use of GUI by creating a human face and converting celsius into fahrenheit

Q1. Write a program to draw human face using GUI

Algorithm :

Step1: Import relevant method from tkinter library

Step2: Create an object corresponding to the parent window

Step3: Create an object from canvas() & place onto parent window along with height & width

Step4: Now use pack() for positioning of widget onto parent window

Step 5: Now create an object face & use object create_oval() with co-ordinates 50,50,350,350 & outline = "black", fill = as attribute to create face

Step6: Now create eye 1 object & again use object create_ov with appropriate co-ordinates along with fill as attrib to create left eye

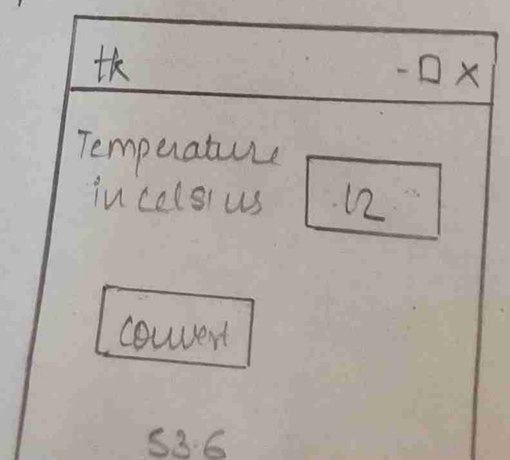
Step 7: Now repeat same step 6 to create right eye

Step 8: Create an object mouth & use object create_arc() with appropriate co-ordinates, start = 0, extent = -180 & fill = "red", width = "5" as attribute to create mouth

Step 9: Finally, use mainloop()

```
from tkinter import *
window = Tk()
farhenheit = DoubleVar()
farhenheit.set (32.0)
def convert (celsius):
    farhenheit.set ((9.015.0)* celsius+32)
l1 = Label (window, text = " Temperatures in celsius
l1.grid (row=0, column=0)
e = Entry (window, text variable = celsius)
e.grid (row=0, column=1)
celsius = IntVar()
l2 = Label (window, text variable = farhenheit)
l2.grid (row=2, column=0, columnspan=2)
B = Button (window, text = " calculate ", command=lambda
                convert (celsius.set())
B.grid (row=1, column=0, columnspan=2)
~~main~~
window mainloop ()
```

#output



Q2 Write a program to convert celsius into farhenheit

Algorithm:

Step1: Import all the relevant method in the tkinter library

Step2: Create object corresponding to the parent window from Tk

Step3: Now create an object l2 using label & place it onto parent window & use text attribute as enter a no

Step4: Initilize celsius as integer using IntVar()

Step5: Create another object & use entry widget to enter the input & place it onto the parent window

Step6: Now use grid() for positioning the object onto parent window with text variable attribute

Step7: Now again use label() along with text variable using attribute to display

Step8: Finally use mainloop()