Practical - 1.

Objective: Demonstrate the use of different f
different file accessing modes diff
attributes read methods.

Step1: Create a file object using open method and
use the write access mode followed b
writing some contents onto the file and
then closing the file.

Step2: Now open the file in read mode and t
Use read(), readline() and readlines(
Store o/p in variable and finally d
the contents of variable.

Step3: Now use the fileobject for finding the
of the file, the file mode in which it
opened whether the file is still open or
close and finally the o/p of the softs
attribute.

```
fileobj = open("abc.txt", "w").
file obj = write("science subjects"+"\n")
file obj.write("physics in chemistry in Maths in")
file obj.close()
file obj = open("abc.txt", "r").
#read()
str1 = file obj.read()
print("The output of read method:", str1)
file obj.close()
>>> ('The outputs of read method:', 'science subject in
    in chemistry in maths\n')
# read lines()
   fileobj = open("abc.txt", "r").
   str2 = fileobj.readline()
   print("The output of readline method:", str2)
   fileobj.close().
) ('The output of readline method:', 'science subject
readlines()
file obj = open("abc.txt", "r"
str3 = fileobj.readlines().
print["The output of readlines method:", str3]
fileobj.close().
'The output of readlines method:', ['science subj
physics in chemistry (maths]).
'le attributes
= fileobj.name
int("name of file(name attribute):", a)
ame of file(name attribute)', 'abc.txt')
 fileobj.closed
+("cclose) attribute:", b)
closr) attribute:', 'True').
```

Step4: Now open the file obj in write mode write another content close subsequently then again open the file obj in 'wt' mode that is the update mode and write contents.

Step5: Open fileobj in read mode display the update written content and close. Open again in 'r+' mode with parameter passed and display the output subsequently.

Step6: Now open file obj in append mode open write method contents close the file obj again open the fileobj in read and display the 'appending' output.

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

# wt mode.
fileobj=open("abc.txt","wt").
fileobj.write("abhishek").
fileobj.close().

# rt mode.
fileobj=open("abc.txt","rt").
s1 = fileobj.read(5).
print ("output of rt", s1).
fileobj.close().
>>> ['output of rt', 'Abhis']

# append mode.
fileobj=open["abc.txt","a")
fileobj.write["data structure")
fileobj.close().
fileobj=open("abc.txt","r").
s3 = fileobj.read().
print ("output of append mode:", s3).
fileobj.close().
>>> ('output of append mode:', 'Abhishek data str

# write mode
fileobj=open("abc.txt","w")
fileobj.write("DBMS")
fileobj.close().
# read mode
fileobj=open("abc.txt","r")
s=fileobj.read()
print ("output of read mode
>>> ['output of read mode',"s
```

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

# wt mode.
fileobj = open("abc.txt", "wt").
fileobj.write("abhishek").
fileobj.close().

# rt mode.
fileobj = open("abc.txt", "rt").
s1 = fileobj.read(5).
print ("output of rt", s1)
fileobj.close().
) [output of rt', 'Abhis')

- append mode.
'le obj = open["abc.txt", "a")
ile obj.write["data structure")
'le obj.close().
ile obj = open("abc.txt", "r").
3 = file obj.read().
rint ("output of append mode:", s3).
le obj.close().

'output of append mode', ', 'Abhishek data structure'

# write mode
fileobj = open("abc.txt", "w")
fileobj.write("DBMS")
fileobj.close().
# read mode
fileobj = open("abc.txt", "r")
s = fileobj.read()
print ("output of read mode
>>> [output of read mode', 'DBMS')
```

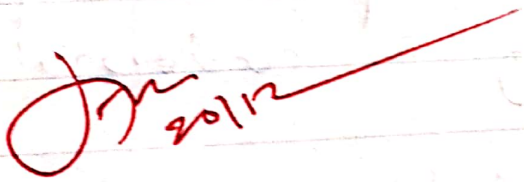
```
# tell()
file obj = open ("abc.txt", "r")
pos = file obj.tell()
print("tell:", pos)
fileobj.close
>>> ('tell():', 0L)

# seek()
file obj = open("abc.txt", "r")
st = fileobj.seek(0,0)
print ('seek(0,0) is:', st)
file obj.close
>> ('seek(0,0) is:', None)
fileobj = # finding length of different lines exist
fileobj = open ("abc.txt", "r")
stat = fileobj.readlines()
print ("output:" stat)
for line in stat:
        print [len(line)).
file obj.close()

('output:', ['Abhishek data structure'])
```

21.

Step 7: Open the fileobj in read mode declare a variable and 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 in read mode and closing subsequently

Step 9: Open file obj read mode also use the readline method and store the output consequently in and print the same for counting the length use th for conditional statement and display the length.



Practical – 2

Aim: Demonstrate the use of iteration and iterators.

Q1) Write a program using iterable objects displaying the odd numbers in range 1 to

Algorithm:

Step1: Define an iter() with argument and initialize the value and return that value.

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

Step3: Now create an object of the given cl ~~class~~ and pass the object in the iter method

```
# code:
class odd:
    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
>>> z.next()
5
>>> z.next()
7
>>> z.next()
9
>>> z.next()
11
```

```
# code.
class power:
    def __iter__(self):
        self.num=0
        return self.
    def next(self):
        if self.p<=10:
            num=self.p
            self.p+=1
            po=2**num
            print("2**",self.p-1,"=",po)
            return po
        else:
            raise stop iteration.
>>> p=power()
>>> x=iter(p)
>>> x.next()
    2**0=1.
>>> x.next()
    2**1=2
>>> x.next()
    2**2=4.
>>> x.next()
    2**3=8.
```

Write a program using an [illegible] for calculating the [illegible] of the given number and [illegible] [illegible] [illegible] [illegible] the value calculated should be 1, 2, [illegible]

Algorithm:

Step 1: Define [illegible]( ) with argument and [illegible] initialize value and return the value.

Step 2: Now define [illegible]( ) with an argument and compare the upper limit by using control statement.

Step 3: Now create an object of the given class and pass this object in the [illegible] method.

58

Write a program using iterable concept to find factorial of number in range 1 to 10

Algorithm:

Step1: Define a iter() with argument a, initialize 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 and pass the object to iter method.

CODE

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

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

Write a program using iterable conc. to display mult of 2 in 1 to 10.

Step1: Define a iters() with argument and initialize 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 and pass that's object in the iter

code

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

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

Practical-4.

Aim :- Demonstrate the use of regular expr

Theory :- Regular expression represents the sequence of characters which is mainly used for finding and replacing the give pattern in a string and for this we import remodule and common usage of regular expression

Algorithm :-

Step1 :- Import re module

Step2 :- Define a string variable and a str to bworked upon.

Step3 :- \d is used for matching all decimal digit whereas is used to match non-decd

Step 1- Import re module and apply and str

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

Step3 :- Now use if condn. statement for use t know whether to match is found or not

```
String = 'hello1234 abc 5678'
result = re.findall("\d+", String)
print(result)
print(result[1])
```

op:-
['1234', '5678']
['hello', 'abc']

② import re.

```
string = "Python is an imp. language"
result = re.search("\APython", string)
    print(result)
if result:
    print("Match found")
else:
    print("match not found").
```

op:-
<re.match object; span=(0,6),
match = 'Python'>
match found

③ import re.

```
String = "Python is imp"
result1 = re.findall("\w+", String)
result = re.findall("w", String)
print(result1)
print(result2)
```

['Python', 'is', 'imp']
['Python', 'is', 'imp']

④ import re

```
i=["987654 3210", "8765326"
"7654321098", "6543210986" for
```

elements in i:
print("correct mob no")
print(result.group()).
else:
print("incorrect mobile")

op:- correct mobile no.
987654 3210.
Correct mobile no.

incorrect mobile no.
incorrect mobile no.

4) step1:- import re module and apply string a module no.

step2:- Now use the conditional statement to find if the number starts with 8 or 9 and the total number should length of 10 use match() module to find the match in given string.

step3:- Use if conditional statement to know whether we have a match or not if we have group() to display the o/p and if we dont display correct no.

5)

→ ① Use findall() to extract word along with space and use "\w" to extract word without space.

Jan 17(7)

(5) import re
string = "python is important"
result1 = re.findall ("\w+", string)
result2 = re.findall ("\w+ A", string)
print (result1)
print (result2)

o/p:-
['python', 'is', 'important']
['import']

(6) import re.
string = "Nikki 20-24-12-2020"
result = re.findall ('\d{2}-\d{2}-\d{4}
print (result)

o/p
['24-12-2019']

Practical-5

Topic - GUI components

Step1: Use the tkinter library for importing 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 mainloop() for triggering of the corresponding above mention events.

#2:

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

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

Step3: Use the pack() along 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

Step4: Use the mainloop() for the triggering of the corresponding events.

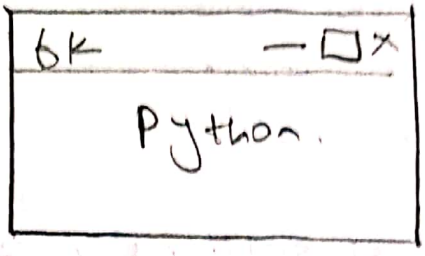
Step5: No repeat above steps with the label() which takes the following arguments

1) Name of the parent window.
2) Text attribute which defines the string.
3) The background color (bg) 4) The background fg and the use the pack() with a relevant padding attribute

creation of parent window

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

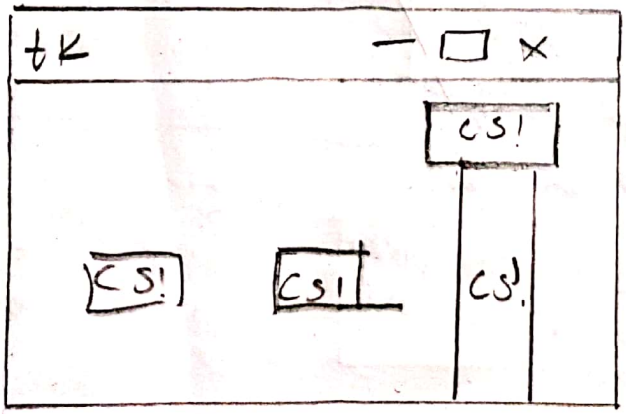
o/p:-



#2:

```
from tkinter import *
root Tk()
l=label(root, text = "python")
l.pack()
l1 = label(root, text = 'cs', bg = "grey", fg = "black"
           font = "10").
l1.pack(side = LEFT, padx = 20)
l2.label(root, text = "cs", bg = "light blue", fg = "black".
           font = "20").
l2.pack(side = LEFT, pady = "30").
l3.label root, text = ("cs", bg = "yellow", fg = "black"
                       font = "40").
l3.pack(side = Top, ipad x = 40)
l4 = Label(root, text = "cs"), bg = "orange". fg = "black"
           font = "10").
14.pack(side = Top, ipad y = 50)
root.mainloop().
```

o/p:-



Practical - 5(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 parent window.

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

Step4: Now define the parent window and define the option with control variable.

Step5: Use the listbox() and insert options on the parent window along with the pack() with specifying anchor attribute.

Step 6: Create an object from radiobutton which will take following arguments parent window object, text variable which will take the values option 1, 2, 3..... Variable argument, corresponding value and trigger the function declared.

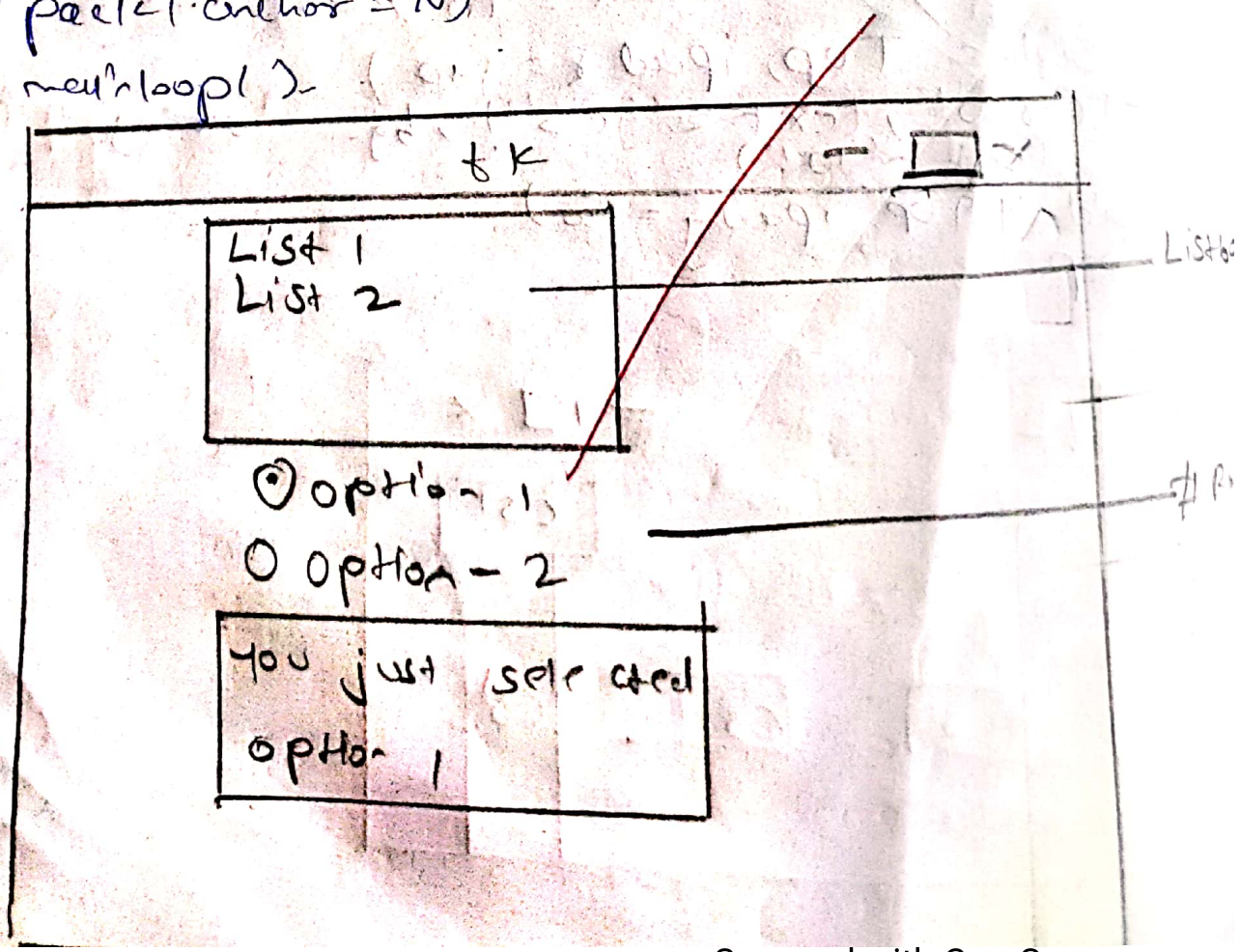
Step7: Now call the pack() for radio object so created and specify the argument using another attribute.

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

#1:

Radio button

```
from Tkinter import *
root = Tk()
root.geometry("500x500")
def select():
    selection = "you just selected" + str(var.get())
    l1 = Label(text=selection, bg="white", fg=
    l1.pack(side=TOP)
var = StringVar()
l1 = Listbox()
l1.insert(1, "List1")
l2.insert(2, "List2")
l1.pack(anchor=N)
r1 = Radiobutton(root, text="option1", variable=v
                 value="option1", command=se
r1.pack(anchor=N)
r2 = Radiobutton(root, text="option2", variable=
                 value="option2", command=sele
r2.pack(anchor=N)
root.mainloop()
```



#2

Step1 ~~#2~~:- Import relevant methods from the 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 scroll bar().

Step5:- Use the pack() along with the scroll bar() object with side and fill attributes

Step6:- Use the mainloop with the parent object

#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 (0) 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 use the pack method

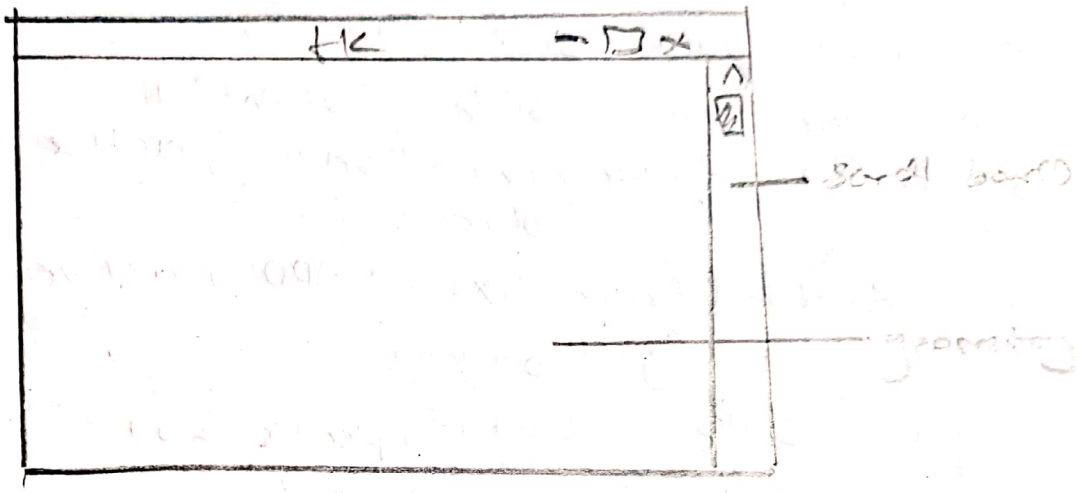
Step6:- Use the listbox method along with the attributes like width, height font. Do create a listbox method's object. Use pack() for the same

Step7:- Use the scrollbar() with an object use the attribute of vertical. Then configure the same with object created from the scroll bar() and use pack()

Step8:- Trigger the

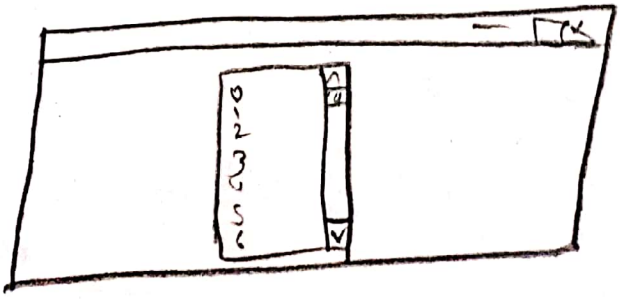
#2
Scroll bar()

```
from tkinter import *
root = TK()
root.geometry ("500 x 500")
S = Scroll bar ()
S.pack (side = "right", fill = "y" )
root.mainloop().
```



#3
Using frame widget

```
from tkinter import *
window = TK()
window.geometry ("600 x 500")
label (window, text = "number :").pack()
frame = Fram (window)
frame.pack ()
list nodes = Listbox(frame, width=20, height=20,
                     font=("Times new Roman", 10))
list nodes.pack (side = "left", fill = "y")
Scroll bar = scroll bar (frame, orient = "vertical")
Scroll bar.config (command = listnodes.yview)
Scroll bar.pack (side = "right", fill = "y")
for x in range(100):
    listnodes.insert (END, str (x))
window.mainloop().
```



#4:

Step1:- Import relevant methods from tkinter library.

Step2: Define the object corresponding to parent window and define the size of parent window in terms of 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 from the method and place it on the parent window on LEFT SIDE

Step5:- Similarly, define the RIGHT frame and subseq define the button object placed on to the given frame with the attribute as text, active background and foreground.

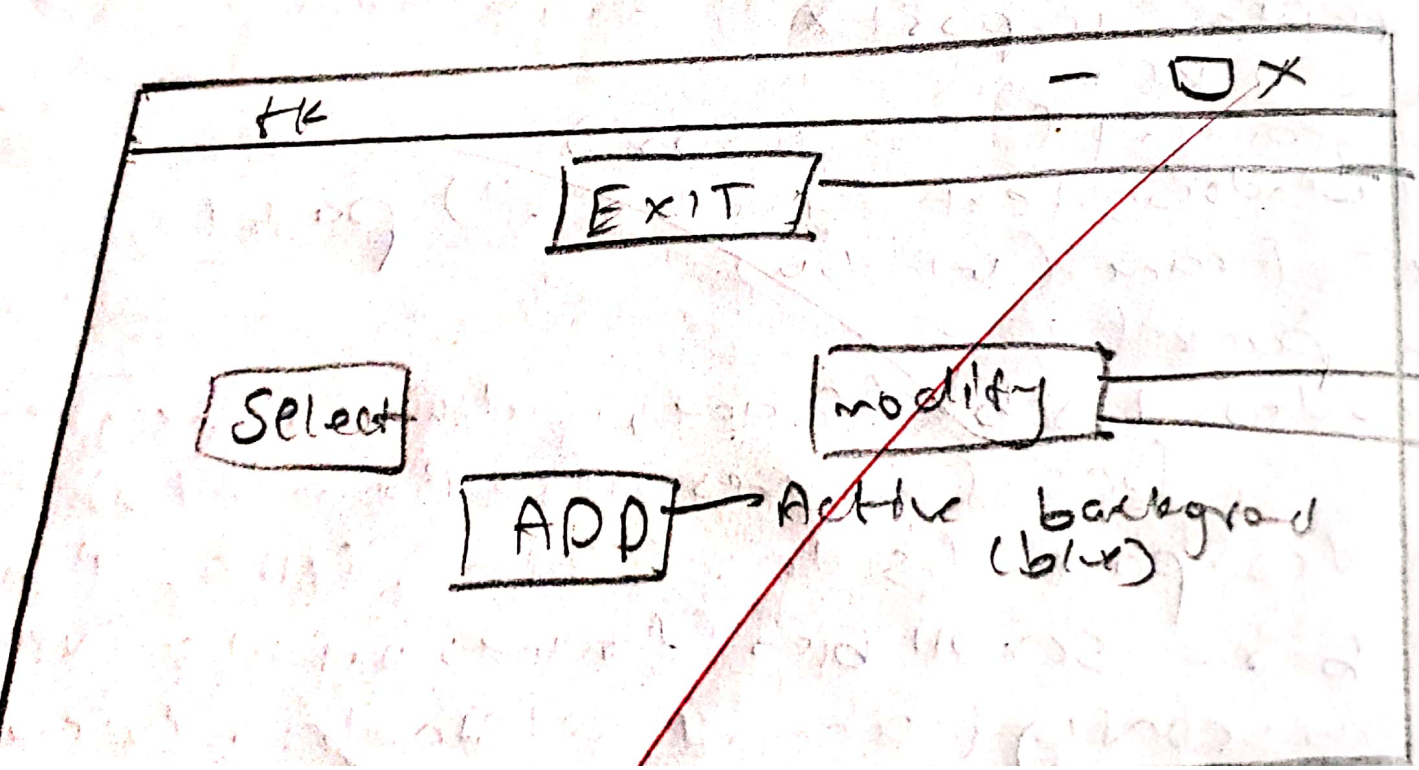
Step6:- Now use pack() along with the side attribute

Step7: Similarly create the button objects corresponding to the MODIFY operation put it into frame object on side = "right"

Step8:- create another button object and place it onto the right frame and label it as EXIT.

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

```
# GUI:
from tkinter import *
window = Tk()
window.geometry("680x500")
frame = Frame(window)
frame.pack()
left frame = Frame(window)
left frame.pack(side = "LEFT")
right frame.pack(side = "right")
b1 = Button(frame, text = "select", active background=
            fg = ("blue")).
b2 = Button(frame, text = "ADD", active background=
            fg = ("black")).
b1.pack(side = "LEFT", padx = 20)
b2.pack(side = "right", padx = 30)
```



Practical - 6

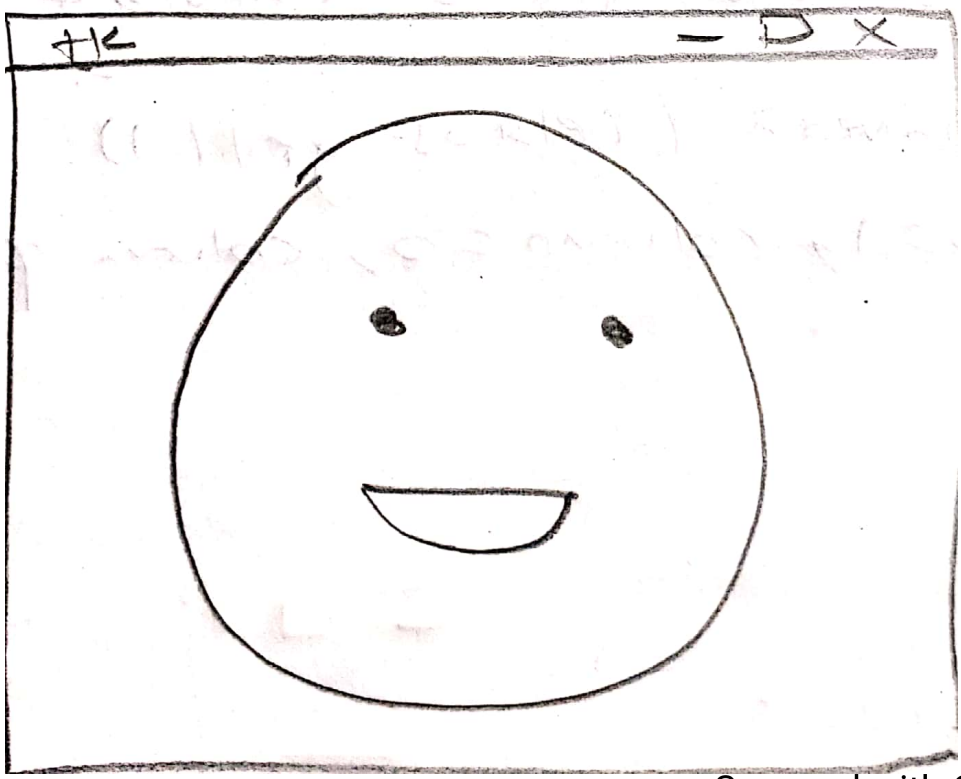
Aim: Demonstrate the use of GUI by creating human face and converting celsius into f...

Q1) Write a program to draw human face ...
GUI

Algorithm:

1: Import relevant methods from tkinter library
2: Create an obj. corresponding to the parent ... from tk()
3: Create an object from canvas() & place in ... window along with height and width
4: Now use pack() for pasting of widget on ... parent window along with height and ...
5: Now repeat same step 6 to create ...
6: Finally use the mainloop()

```
Enter import*
<()
as(root, width= 500, height =
c()
.create_oval (50,50,350,
reate - oval (125,125,175,17
reate - oval (225, 125, 275,
create_oval (125,125,275, 17
                    extent=1
ainloop
```



12) Write a prog to convert into fahrenheit using GUI

Algorithm:

1. Import all the relevant methods in the tkinter library
2. Create object corresponding to the parent window tkinter Tk()
3. Now initialize fahrenheit as doubleVar() and set it to 320
4. Now define a function convert with arrangement celsius to convert celsius into fahrenheit using. set()
5. Now use grid() for position the object onto the parent window.
6. Initialize celsius as integer using intVAR
7. Now again use label() along with text variable using attribute to display output and use grid() for positioning.
8. Finally use mainloop():

code

```
from tkinter import *
window = Tk()
fa = DoubleVar()
fahrenheit = Set.set(32.0)
def convert (celsius):
    fa.set(69.0*15.0) * (celsius
l1 = label(window, text = "Temperature in
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 = fa
l2 = grid(row=2, column=0, colum
B = button(window, text = "calculate
                convert((celsius.get()))
B.grid(row=1, column=0, colum
mainloop()
```

output