PRACTICAL NO 1

Aim: To demonstrate the use of different file accessing modes, different files, different attributes and read methods

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.

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.

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

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

Step 5: Open fileobj in read mode and display the updated written contents and close. Open again in 'r+' mode with parameter passed and display the output subsequently.

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

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

# W+ mode
fileobj = open ("abc.txt", "w+")
fileobj.write ("loukik sir")
fileobj.close()

# r+ mode
fileobj = open ("abc.txt", "r+")
S1 = fileobj.read (6)
print ("Output of r+", S1)
fileobj.close()
>>> ('output of r+', 'loukik')

# 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:', 'loukik sir 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!", S)
>>> ('Output of read mode!',
     'loukik sir')
```

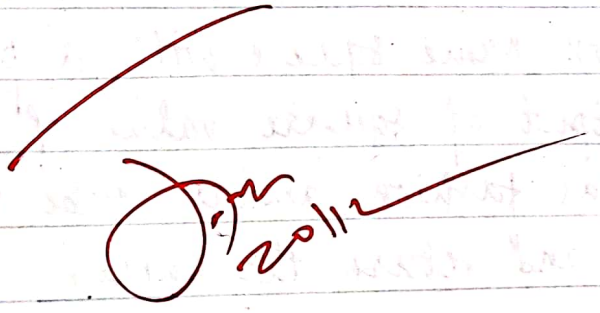
```
# tell()
fileobj = open("abc.txt", "r")
pos = fileobj.tell()
print("tell():", pos)
fileobj.close
>>> ('tell():', 0L)

# seek()
fileobj = open("abc.txt", "r")
st = fileobj.seek(0,0)
print("seek(0,0) is :", st)
fileobj.close()
>>> ("seek(0,0) is :'None)
fileobj = open("abc.txt", "r")
st1 = fileobj.seek(0,1)
print("seek(0,1) is :", st1)
fileobj.close()
>>> ("seek(0,1) is :", None)
fileobj = open("abc.txt", "r")
st2 = fileobj.seek(0,2)
print("seek(0,2) is :", st2)
fileobj.close()
>>> ("seek(0,2) is :", None)
# Finding length of different lines exist within lines
fileobj = open("abc.txt", "r")
stat = fileobj.readlines()
print("output:" stat)
for line in stat:
    print(len(line))
fileobj.close()
>>> ('output:', ['loukik sir data structure'])
```

Step 7: Open the fileobj in read mode, declare a variable and perform fileobject dot tellmethod and store the output consequently in variable.

Step 8: Use the seek method with the arguments with opening the file obj in read mode and closing argument.

Step 9: Open fileobj 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 conditional statement and display the length.



ES PRACTICAL NO 2

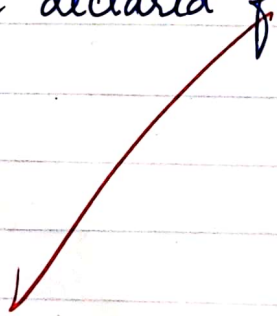
Objective : Iterators

Step 1 : 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.

Step 2 : The similar output can be obtained by using for conditional statement. An iterable variable is to be declared in for loop which will iterate.

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

Step 4 : Call the declared function using function call.



```
# iter() and next()
mytuple1 = ("banana", "orange", "apple")
myiter1 = iter(mytuple)
print(next(myiter1))
myiter2 = iter(mytuple)
print(next(myiter2))
myiter3 = iter(mytuple)
print(next(myiter3))
>>> banana
    orange
    apple
# for loop
mytuple1 = ("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
unit1 = [square, cube]
```

```
for r in range (s):
    value = list (map (lambda x: x(r), funct1))
    print (value)
```

```
>>> [0, 0]
    [1, 1]
    [4, 8]
    [9, 07]
    [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 num 1
        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 range use the list type casting with map method declare a 'lambda' i.e anonymous function and print the same.

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

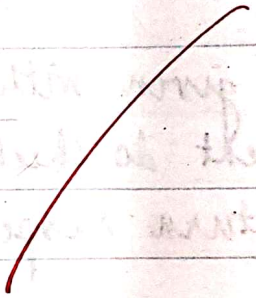
Step 7 : Define a function given with a parameter then using conditional statement do check whether the number is even, odd and return respectively.

Step 8 : Define a class and within that define the iter() method which will initialize the first element within the contain object.

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

Step 10: Define an object of a class.

Step 11: Accept an number from the user till which use wan
to display the odd numbers.



```
myobj = odd()
myiter = 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
```

20/11/19

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

Output:

```
Enter class fyes
Enter numeric char
Enter class : 13
```

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

Output:

Operation carried out successful

PRACTICAL NO 3

Aim: To demonstrate exception handling

1] Write a program using the exception handling method of the nature Arithmetic error.

Step 1: Use the try block and except the input using the raw input method and converts it into the integers datatype and subsequent terminate the block.

Step 2: Use the except block with the exception name as value error and display the appropriate message it the suspicious 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 exception for the given input.

Step 1 : Write the try block open the file using the write mode and write some content on the file.

Step 2: Use the except block with IO error and display the message regarding missing of the file or incompatibility of the mode. Use the else block to display a message that the operation is carried out respectfully.

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

Step 1: Define a function which accepts an argument and using the assert statement whether the given list is empty list and accordingly return the message

Step 2: Close the function and in the body of program define certain elements in the list and take some appropriate actions.

4] WAP to check the range of the age of the students in the given class and if the students in the given class age do not fall in given range else the value error exception otherwise return the valid number

Step 3: 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.

Step 4: Use except with ValueError and print the message not a valid range.

```
# assert error
  def assert_(n)
      assert ( len (n) == 0)
      print (" list is empty")
  Var l = [ ]
  print ( assert_ (var l))
```

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 Value Error:
        print ("not valid age")

Output:
Enter your age : 18

code

```
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']
```

PRACTICAL NO 4

Aim: Demonstrate the use of regular expression.

Theory: Regular expression represents the sequence of characters which is mainly used for finding and replacing the given pattern in a string and for this use import remodule and common usage of regular expression. It involves following functionalities.

8] Write a regular expression segregating numeric and alphabetic values from a given string

Algorithm:

Step 1: Now apply string and pattern in findall() and display the output.

Step 2: \d is used for matching all decimal digits whereas is used to match non-decimal digits.

2] Write a regular expression for finding the match string the beginning of given sequence.

Algorithm:

Step 1: Import re.module and apply a string.

Step 2: Use search() with "\A Python" and string as two parameters

Step 3: Now display the output.

Step 4: Now use if conditional statement for user to know whether the match is found or not.

```
# 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
```

(18)

Code

```
import re
li = ["9876542310", "8765932109", "96543210981", "6543210981"]
for elements in li:
    result = re.match("[8-9]-{1}[0-9]{9}", element)
if result:
 print("correct mobile no")
 print("result.group(1)")
else:
  print("Oncorrect mobile no")
```

Output

```
>>> Correct mobile no
    9876T5543210
    Correct mobile no
    8765463210|
    Incorrect mobile no
```

3] Write a regular expression to check whether the given mobile number starts with 8 or 9 & the total length is 10!

Algorithm:

Step1: Import the 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 & the total number should be length of 10. Use match() inside for the 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.

4] Write a regular expression for extracting a word from given string along with space characters in between the words and subsequently extract the word without space character.

Algorithm:

Step 1: Import re module and apply a string.

Step 2: Use findall( ) to extract a word from given string.

Step 3: Use "\w+" to extract word along with space and use "\w+" to extract word without space.

Step 4: Now display the output.

code

```
import re
string = "Python is important"
result1 = re.findadd ("\w*", string)
result2 = re.findadd ("\w+", string)
print (result1)
print (result2)
```

output

```
>>> ["Python", "is", "important"]
    ['Python', 'is', 'important']
```

```
# code 5
import re
string = "Python is important"
result= re.findall ["n\w+", string ]
result1 = re.findall ["\w+$", string]
print [result1]
print ( result1)

# Output
>>> [' Python']
>>> [' Important']

# code 6
import re
string = " Sakshi 201  04-03-2020"
result = re.findall ["\d{2}-\d{2} -\d{4}", string]
print [result]

# output
>>> ['04-03-2020']
```

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

Algorithm:

Step 1: Import re module and apply a string.

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

Step 3: Display the output

6] Write a regular expression for extracting the data in format dd.mm.y by using the findall[] where the string has the following format
SAKSHI 201 04.03.2020

Algorithm:

Step 1: Import re module and apply string

Step 2: Use findall method and use '\d 23-\d{2}-\d{4} as
parameter

Step 3: Now display the output

PRACTICAL NO 5

Aim: GUI components

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

Step 2: Create an object using the TK()

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

Step 4: Use the mainloop() for the corresponding about m
events.

#2:

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

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

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

1. Side = LEFT, padx = 20
2. Side = LEFT, pady = 30

```
# ... window for
from tkinter import *
root = Tk ( )
l = label ( root, text = "Python")
l.pack ( )
root.mainloop ( )
```

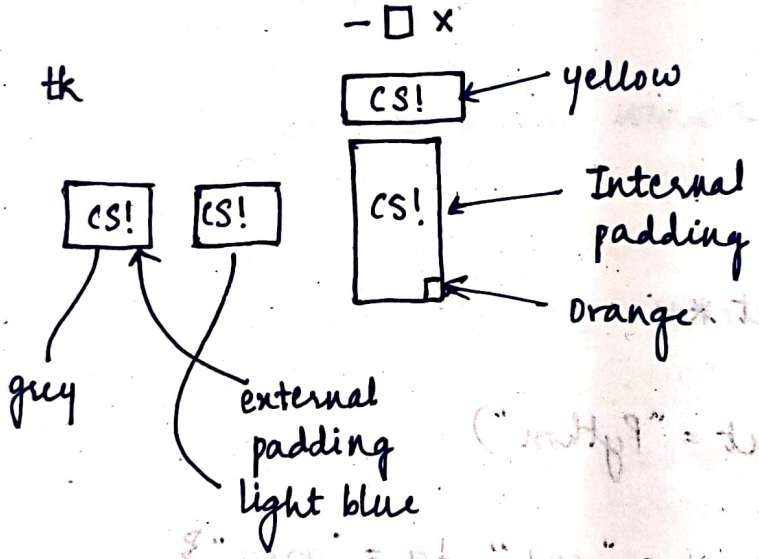
Output :

tk — □ X,
python

#2

```
from Tkinter import *
root. Tk ( )
l= label ( root, text = "Python")
l. pack ( )
l1 = label ( root, text = "CSI", fg = "grey"}
   fg = "black", font="10")
l2 = label ( root, text = "CSI", fg = "light blue",}
   fg = "black", font = "20")
l2. pack ( side = LEFT, pack up = "30")
l3 = label ( root, text = "CSI", fg = "yellow",
   fg = "black", font = "10")
l3. pack ( side = TOP, ipadx = "w")
```

```
lg = Label(root, text="CS!", fg="orange"
           fg="black", font="0")
lg.pack(side=TOP, ipady=50)
root.mainloop()
```



Step 4: 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 colours (fg)
4. The foreground and than use the pack ( ) with 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.

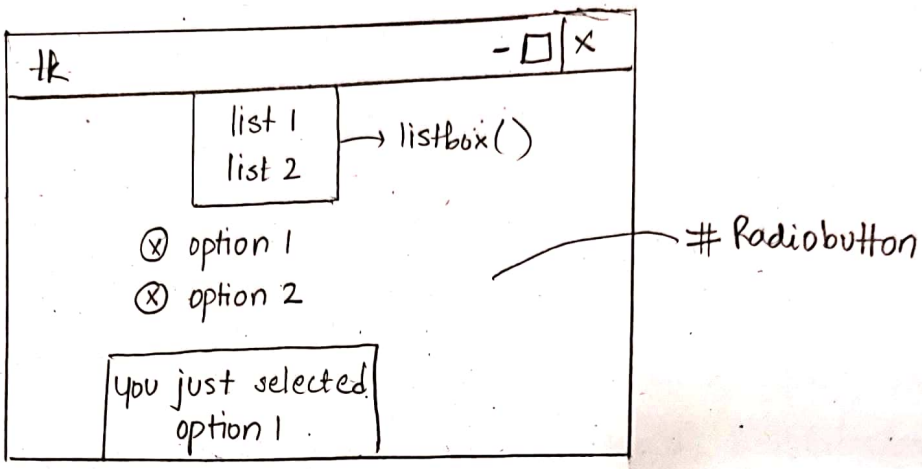
Step 3: Now define a function which tell the user about the given selection made from multiple option of variable.

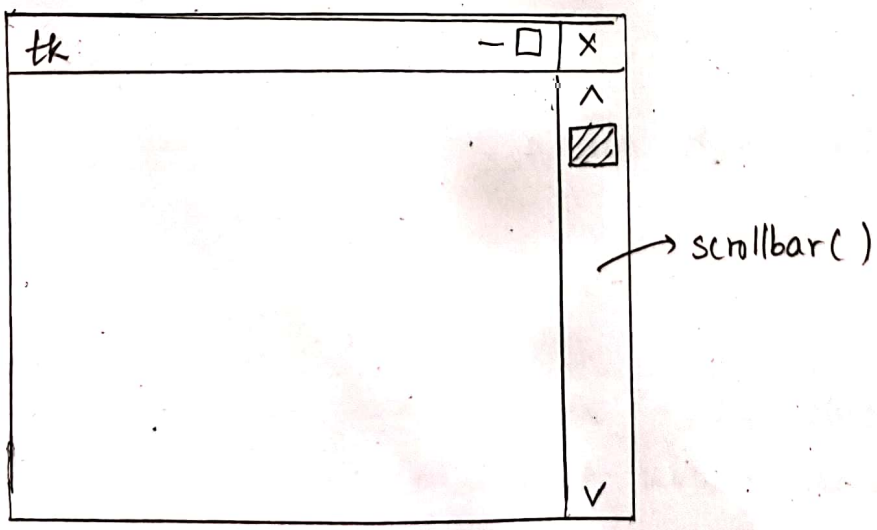
Step 4: Now define the parent window and define the option with control variable.

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

Step 7: Now call the pack () for radio object as 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 window

Step 4: Create an object use the scrollbox ( )

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

Step 6: Use the mainloop with parent object.

#3

Step 1: Import the relevant libraries from the tkinter method

Step 2: Create an corresponding object of the parent window

Step 3: Use the geometry manager with pixel size (650×500) or any other suitable pixel values.

Step 4: Use the label widget along with the pointer object created and subsequently use the pack method.

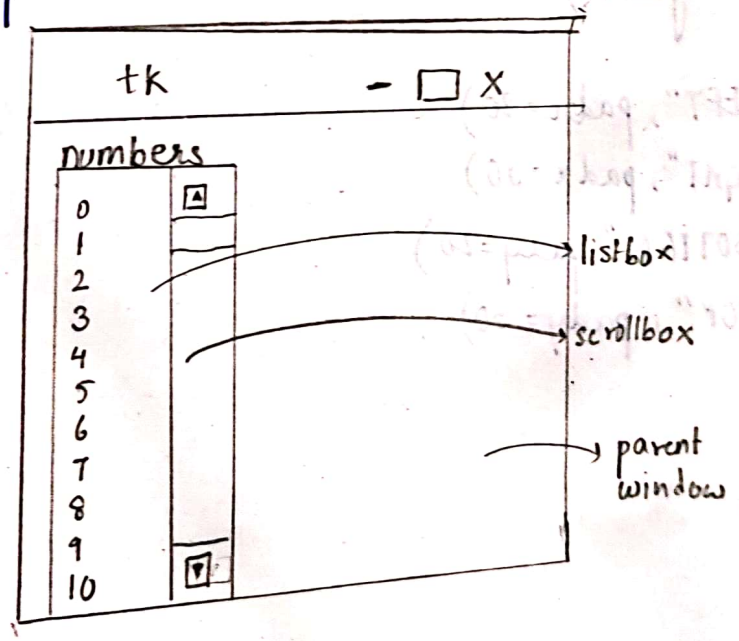
Step 5: Use the frame widget along with the pointer object created and use the pack method.

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 attributes of vertical then configure the same with object create 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 = list box (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 ("650X500")
frame = Frame ( window )
frame.pack ( )
left frame = frame ( window )
left.frame.pack ( scale = "left")
right frame = 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 ( frame, 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

Step 1: Import relevant method from thinter library

Step 2: Define the object corresponding to parent window & define the size of parent window in terms of no of pixels.

Step 3: Now define the frame object from the method & place it on the parent window.

Step 4: Create another frame object termed as the left frame and put it on the parent window on its LEFT side.

Step 5: Similarly define the RIGHT frame and subsequently define the button object placed onto the given frame.

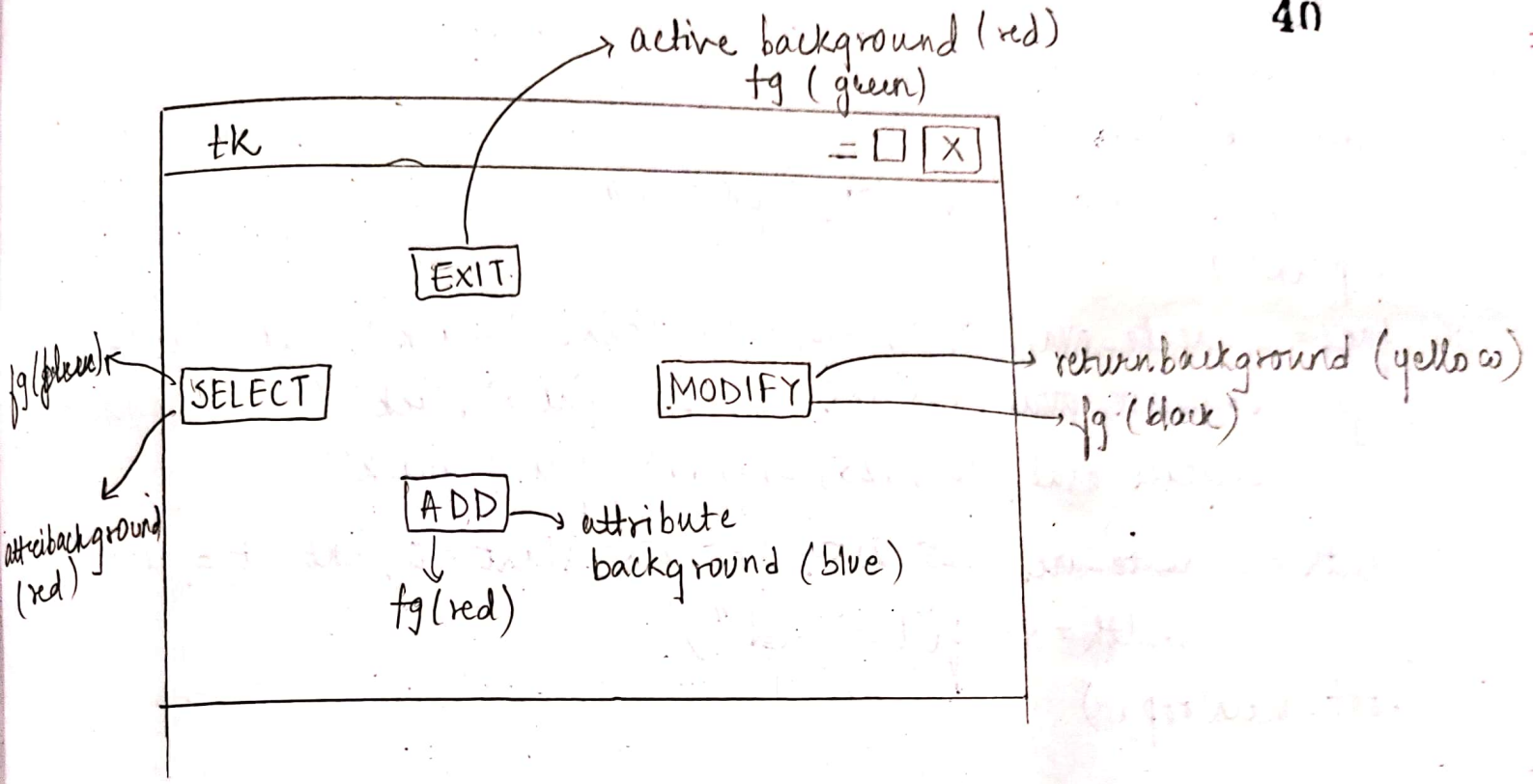
Step 6: Now use the pack() along with the scale attributes.

Step 7: Similarly create the buttons object corresponding to the modify operations put it into frame object on side = "RIGHT"

Step 8: Create another button object and place it on the RIGHT frame and label the button as ADD.

Step 9: Add another button and fit it on the top and label it as EXIT.

Step 10: Use the pack() simultaneously for all the object and finally use the mainloop().



(15)

```
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(225, 125, 275, 175, fill = "black")
mouth = c.create_arc(125, 225, 275, 275, start = 0, extent = -180,
                     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 farenheit.

8.1] Write a program to draw human face using GUI

Algorithm:

Step1: Import relevant methods from tkinter library

Step2: Create an object corresponding to the parent window from Tk().

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.

Step5: Now create an object face & use object create_oval() with coordinates 50, 50, 350, 350 & outline = "black", fill = "yellow" as attribute to create face.

Step6: Now create eye1 object & again use object create_oval() with appropriate coordinates along with fill as attribute to create left eye.

Step7: Now repeat same step 6 to create right eye.

Step8: Create an object mouth and use object create_arc() with appropriate co-ordinates, start = 0, extent = 180 & fill = "red", width = 5 as attribute to create mouth.

Step9: Finally use the mainloop()

9.2] Write a program to convert celsius into farenheit using GUI.

Algorithm:

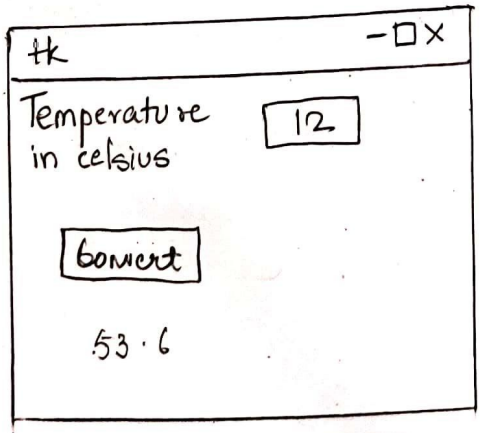
Step1: Import all the relevant methods in the tkinter library.
Step2: Create object corresponding to the parent window from Tk()
Step3: Now initialize farenheit as DoubleVar() & set it to 320.
Step4: Now define a function 'convert' with argument celsius to convert celsius into farenheit using set().
Step5: Now create an object l2 using label() & place it onto parent window & use text attribute as enter a no.
Step6: Now use grid() for position the object onto the parent window.
Step7: Initialize celsius as integer using IntVar()
Step8: Create another object & use entry widget to enter the input & place it onto the parent window.
Step9: Now use grid() for positioning the object onto parent window with text variable attribute.
Step10: Now again use label() along with text variable using attribute to display output & use grid() for positioning.
Step11: Finally use mainloop().

code

```
from tkinter import *
window = Tk()
fahrenheit = DoubleVar()
fahrenheit.set(32.0)
def convert(celsius):
    fahrenheit.set((9.015) * celsius + 32)
l1 = Label(window, text = "Temperature 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)
window mainloop()
```

output



proj.py - C:\Users\Pranav\Downloads\proj.py (3.6.2)

File Edit Format Run Options Window Help

```
import tkinter
from tkinter import *
import random
from tkinter import messagebox

answers=["india","canada","japan","china","sydney","egypt","california","switzerland","thailand","chicago"]
words=["dinai","ancdaa","anajp","hncai","ydnsye","gpeyt","ilrofinaca","wsztdinearl","lahitdan","oahcgic"]

num=random.randrange(0,10,1)


def first():
    global words,answers,num
    label.config(text=words[num])

def checkans():
    global words,answers,num
    var=entry.get()
    if var == answers[num]:
        messagebox.showinfo("Wow!","It's the correct answer")
        reset()
    else:
        messagebox.showerror("Error","wrong answer")

def reset():
    global words,answers,num
    num=random.randrange(0,10,1)
    label.config(text=words[num])
```

```
root=tkinter.Tk()
root.geometry("350x400+400+150")
root.title("Jumbling Words!")
root.config(background="black")

label=Label(root,font=("Verdana",18),bg="black",fg="white")
label.pack(pady=30,ipadx=10,ipady=10)


select=StringVar()
entry=Entry(root,font=("Verdana",15),textvariable=select)
entry.pack(ipadx=5,ipady=5)

button=Button(root,text="check",font=("Comic sans ms",16),width=16,bg="grey",fg="magenta",relief=RIDGE,command=checkans)
button.pack(pady=40)

res_button=Button(root,text="Reset",font=("Comic sans ms",16),width=16,bg="grey",fg="light green",relief=RIDGE,command=reset)
res_button.pack()



first()

root.mainloop()


print("Sakshi Kangane")
```