

Degree College
Computer Journal
CERTIFICATE

SEMESTER Sem - II UID No. _____

Class AUBSL (C.S) Roll No. 1810 Year 2019 - 20

This is to certify that the work entered in this journal
is the work of Mst. / Ms. Ayush Kumar Mishra

who has worked for the year 2019 - 20 in the Computer
Laboratory.

Teacher In-Charge

Head of Department

Date : _____

Examiner

INDEX

No.	Title	Page No.	Date	Staff Member's Signature
	Sem II			
1.]	Demonstrate the use of different file accessing modes different attributes read methods.	24	28/11/19	late Jm
2.]	Iterators.		5/12/19	Incomplete Dr 19/12/20
3.]	Exception		19/12/19	Jm 16/01
4.]	Reg Exp		9/1/20	Incomplete Dr T6/01/20
5.]	GUI Component (A1B)		16/01/20	Jm 2/1/20
6.]	GUI Components (e) (Message Box, Relief Method)		23/1/20	Jm 6/12
	GUI Component (Transversing (PhotoImage))		6/02/20	Jm
8.]	GUI component(Spinbox, Paned Window, canvas)		6/2/20	late Jm
6.]	Database Connectivity		12/02/20	Jm

Practical no: 1:

Objectives → Demonstrate that use of different file accessing modes, different attributes and methods.

Step 1:- Create a file object using `open` method and use the `write` method and use the `write` access mode followed by writing some contents onto 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 file object for finding the name of the file mode in which it's opened whether the file is still open or close and finally the output of the `softspace` attribute.

```
file obj = open ("abc.txt", "w") # file open (write mode)
file obj.write ("computer science subjects" + "\n") #>>>
file obj.write ("DBMS\n Python\n OS\n")
file obj.close()

file obj = open ("abc.txt", "r") # read mode
# read():
st1 = file obj.read()
print ("the output of read method:", st1)
file obj.close()
>>> ('The output of read method:', 'computer science
subjects\n DBMS\n Python\n OS\n')

# readline():
file obj = open ("abc.txt", "r")
st2 = file obj.readline()
print ("The output of readline method:", st2)
file obj.close()
>>> ('The output of readline method:', 'computer science subject')

# readlines():
file obj = open ("abc.txt", "r")
st3 = file obj.readlines()
print ("The output of readlines method:", st3)
file obj.close()
>>> ('The output of readlines method:', ['computer
science subjects\n', 'DBMS\n', 'Python\n', 'OS\n'])

# file attributes:
a = file obj.name
print ("name of file (name attribute):", a)
>>> ('name of file (name attribute)', 'abc.txt')
```

Step 4:- Now open the file obj in write mode write some another content close subsequently then again open the file obj in 'wt+' mode that is the update mode and write contents.

Step 5:- Open file obj in read mode display the update written contents and close open again in (wt+) 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 file obj again open the file obj in read mode and display the appending output.

Step 7:- Open the file obj in read mode declare a variable and performs file object file tell method 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 subsequently.

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

```

b = file obj - closed
print ("('close') attribute: ", b)
>>> ('close') attribute: 1, 'True'

c = file obj - mode
print ("file mode", c)
>>> c "file mode", 'wt'
>>> C "softspace"
d = file obj - softspace
print ("softspace", d)
>>> C "softspace", 0

# wt+ mode:
file obj = open ("abc.txt",
file obj - write ("DBMC")
file obj - close()

# read mode:
file obj = open ("abc.txt",
s = file obj - read()
print ("Output of read mode!", s)
>>> C "output of read mode!", 'bulk
sbr'.

file obj - write ("lambik
sbr')
file obj - close()

# wt mode:
file obj = open ("abc.txt", "wt")
s1 = file obj - read(s)
print ("output of wt", s1)
file obj - close()
>>> ('output of wt', 'lambik')

# append mode:
file obj = open ("abc.txt", "a")
file obj - write ("data structure")
file obj - close()
file obj = open ("abc.txt", "a")
s2 = file obj - read()
print ("Output of append mode:", s2)
file obj - close()
>>> C "Output of append mode:", 'lambik
data structure'

```

Lambik

```
# tell :  
fileObj = open("abc.txt", "r")  
pos = fileObj.tell()  
print("tell()", pos)  
fileObj.close()  
??('tell() is:', pos)
```

26

```
# seek():  
fileObj = open("abc.txt", "r")  
st = fileObj.seek(0, 0)  
print("seek(0,0) is:", st)  
fileObj.close()  
??('seek(0,0) is:', st)  
  
fileObj = open("abc.txt", "r")  
st2 = fileObj.seek(0, 1)  
print("seek(0,1) is:", st2)  
fileObj.close()  
??('seek(0,1) is:', st2)  
  
# finding lengths 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:', [len(line)])
```

26

```

# item() and next():

28 mytuple1 = ("banana", "orange", "apple")
myitem1 = iter(mytuple1)
print(next(myitem1))
myitem2 = iter(mytuple1)
print(next(myitem2))
myitem3 = iter(mytuple1)
print(next(myitem3))

>> banana
orange
apple

```

for loop:

```

mytuple1 = ("Akhil", "Anil", "Ajay")
for x in mytuple1:
    print(x)

>>> Akhil
Anil
Ajay

```

Square and Cube:

```

def square(x):
    y = x * x
    return y

def cube(x):
    z = x * x * x
    return z

step1 = [square, cube]

```

Aim: Iterators

Practical no: 2

27

Program 1:

Step1:- Create a tuple with elements that we need to iterate using iter and next method the 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 obtain output of square value of the given number in similar fashion declare value to get the value raised to and the same:

Step4:- Call the declare function using function call.

Step5:- Using for conditional statement specifying the range use the list type casting with map method declare a lambda i.e., anonymous function and print the same.

Step6:- Declare a list num variable and declare some elements then use the map method with help of lambda function give two arguments display the same.

For or In Range (5):

```
Value = list(map(lambda x: x*2), first 11)
print (Value)
```

```
[0,0]
[1,1]
[4,8]
[9,27]
[16,64]
```

map()

```
listnum = [0,4,15,7,19,11,13,15,20,19,25]
listnum = list(map(lambda x: x%5, listnum))
print (listnum)
```

def even(x):

```
if (x%2 == 0):
    return "Even"
```

```
else:
    return "odd"
```

```
list = (map(evens, listnum))
```

odd numbers

class odd:

```
def __iter__(self):
    self.num=1
    return self
```

```
def __next__(self):
    num = self.num
    self.num += 2
    return num
```

```
def __next__(self):
    num = self.num
    self.num += 2
    return num
```

15
myf = odd()
myfier = new myf obj)
myfier . fun (input cin < number : " "))
& int (num cin < number : " "))
cout << myfier ;
if (i > 1);
point (i)

sys enter a number: 15

1

3

5

7

9

11

13

Output:-

>>> f = fact()

>>> x = f.my (f)

>>> x = next (f)

1! = 1

>>> x.next (f)

2! = 2

>>> x.next (f)

3! = 6

23
Step 10: Define a object of a class.

Step 11: Accept one number from the user for which we want to display the odd numbers.

Program for finding factorial of a number using flexible concept in range 1 to 10.

Step 2: Define after (f) with arguments initialize the value and returns the value.

Step 3: Define the next (f) with an arguments and compare the composite then upper limit by using conditional statement.

Step 4: Now calculate an object of the given class & make object in first method.

Practical No: 3

A) Write a program to demonstrate exception handling.

B) Write a program using the exception method to handle arithmetic errors.

Step 1: Use the try block and except the input() function to input the user input method and convert it into the integer datatype and subsequently terminate the block.

Step 2: Use the except block with the exception name as ValueError and display the appropriate message if the suspicious code is the part of the try block.

C) Write a program for accepting the file's name 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 IOError and display the message regarding missing file or the file's inability of the code. Use the else block to display a message that the operation is carried out successfully.

Program

while True:

try:

x = int(input("Enter class"))

break

except ValueError:

print("Enter Numeric Value")

Output:

Enter class 465

Program

try:

f = open("abc.txt", "w")

f.write("Ayush Mishra")

except IOError:

print("Error writing on the file")

else:

print("Operation carried out successfully")

f.close()

Output:

Operation carried out successfully.

```

#1 Program
def accept():
    print("Enter age : ")
    var1 = int(input())
    print("Value entered is ", var1)

```

Output:
12

```

#2 Program
def accept():
    age = int(input("Enter age : "))
    if age >= 20 and age <= 100:
        print("Valid age")
        valid = True
    else:
        print("Not a valid age")
        valid = False
    return valid

```

Output:
Enter age : 4
Not a valid age
Enter age : 18.

3) Write a program using the function to check if the list elements are empty.

Step 1:- Define a function which accepts an argument and check using the len() statement whether the given list is empty list and accordingly return the message.

Step 2:- Close the function in the body of program and define certain elements in list and pass them appropriate.

4) Write a program to check the range of the age of the students in given class and if the age do not fall in the given range else the value is a exception otherwise return the valid no.

~~Step 1:- Define a function which will accept the age of the student from the standard input.~~

~~Step 2:- Use the if condition to check whether the input age falls in the range and so return the age else use the value error exception.~~

Jr. 107

PRACTICAL - 4

Topic : Regular expression

Step 1:- Import re module declare pattern and declare sequence use match function declare arguments if ambiguous matched from point the same otherwise point NOT FOUND!

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

Step 3:- Import re module declare pattern with meta character use the split() and build the output.

Step 4:- Import re module declare string and accordingly declare pattern replace the word with no-space use string without space and print the string.

Step 5:- Import re module declare a sequence use search method for finding subsequently search () with dot operator a group () will show up the matched

methods

import re
pattern = re.compile("EYES")

sequence = "EYES represents Computer Science Stream"
if re.match(pattern, sequence):
 print ("matched pattern found!")

else:
 print ("NOT FOUND!")

>>> matched pattern found!

numerical values (segregation)

import re
pattern = re.compile("\d+")
 string = "Hello 123, howdy 789, 45 hours"
 output = re.findall(pattern, string)
 print (output)
 >>> ['123', '789', '45']

split()

import re
pattern = re.compile("\d+")
 string = "Hello 123, howdy 789, 45 hours"
 output = re.split(pattern, string)
 print (output)
 >>> ['Hello', 'howdy', 'hours']

Step 6:- Import re module declare str with numbers use the conditional statement here we have used up the first conditional statement. use if conditional statement for checking first number is either 8 or 9 and next no. are in range of 0 to 9 and check whether the entered no. are equal to 10. if criteria matches print out all the numbers matched otherwise print failed.

Step 7:- Import re module declare a string use the module with findall() for finding the numbers in the string and declare the same

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

Step 9:- Import re module enter a string use pattern to display only two elements of the particular string use findall() declare two variables with initial value as zero use for condition and subsequently use the if condition statement check whether condition satisfy add up the val else increment the value and display the value subsequently.

```

# no-space:
import re
string = 'abc def ghi'
pattern = '([l-z]+) ([l-z]+)'
replace = ''
replaced = re.replace(string, replace, pattern)
print(replaced)

# group():
import re
sequence = 'python is an interesting language'
v = re.search(r'(\w+) (\w+)', sequence)
print(v)
m1 = v.group(0)
print(m1)

# validating the given set of phone numbers
import re
list1 = ['8004567892', '9145678210', '7865432982',
         '9867543201']
for value in list1:
    if re.match(r'^[8-9]\d{9} [0-9]\d{9}$'):
        value = len(value) == 10:
            print("Criteria matched for even number!")
    else:
        print("Criteria failed!")

```

>>> criteria matched for cell number

criteria matched for cell number

criteria failed!

34

criteria matched for cell number.

vowels

import re

str1 = 'plant is life, overall'

output = re.findall(r'\b[aeiouAEIOU]\w+', str1)

print(output)

>>> ['is', 'overall']

host & domain

import re

seq = 'abc•tcs@edu.com, xyz@gmail.com'

pattern = r'\b[\w]+\.\w+\b

output = re.findall(pattern, seq)

print(output)

>>> ['abc•tcs', 'edu.com', 'xyz', 'gmail.com']

counting of first 2 letters:

import re

s = 'mst•a, ms•b, ms•c, mst•t'

p = r'\b[\ms\mst\ms\b]+

o = re.findall(p, s)

print(o)

m = 0

f = 0

for v in o:

if v == 'ms':

F = F + 1

Xath
Orang

```
else:  
    m=m+1  
    print("No of males is:", m)  
    print("No of females is:", f)  
    res = [ 'm1', 'm2', 'm3', 'm4' ]  
  
if m == 2:  
    print("No of males is : 2")  
    print("No of females is : 2")  
    exit()
```

PRACTICAL - 5 (UIB)

Aim:- GUI Components

Step 1:- Use the tkinter library for importing the feature of text widget.

Step 2:- Create a variable from text method and position it onto the parent window.

Step 3:- Use the pack method along with the object created from the text method. Output →

Step 4:- Use the mainloop method for triggering of corresponding events.

Step 5:- Use the tkinter library for importing the feature of text widget.

Step 6:- Create a variable from text method or. Position it onto parent window.

Step 7:- Use the pack method along with the object created from the text method and use the parameter.

Step 8:- side = LEFT
side = LEFT
side = TOP
side = TOP

padx = 20
padx = 30
padx = 40
padx = 50

* TEXT

CODE →

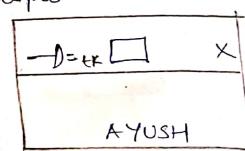
```
from tkinter import *
```

```
root = Tk()
```

```
d = Label (root, text = "AYUSH")
```

```
d.pack ()
```

```
root.mainloop ()
```



LABEL

CODE →

```
from tkinter import *
```

```
root = Tk()
```

```
d1 = Label (root, text = "python")
```

```
d1.pack (root, text = "ayush", bg = "purple", fg = "
```

```
d1.pack (side = TOP, ipadx = 40)
```

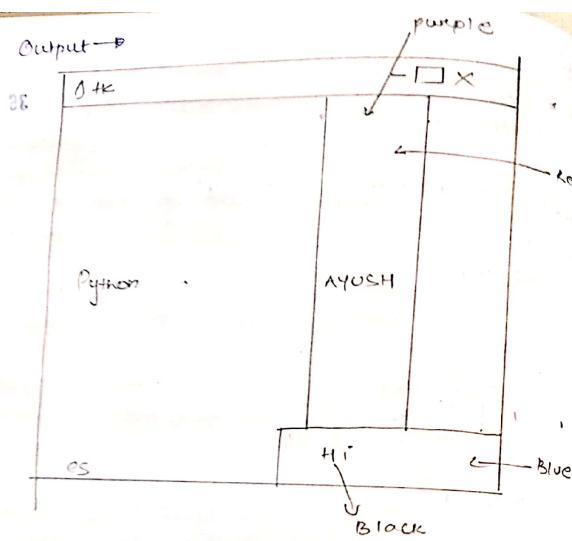
```
m1 = Label (root, text = "CS")
```

```
m1.pack (side = LEFT, padx = 20)
```

```
m2 = Label (text = "shell", bg = "blue", fg = "
```

```
m2.pack (side = LEFT, ipadx = 50)
```

```
root.mainloop ()
```



Step 9: → Use the readloop method for triggering of corresponding event.

Step 10: → Now repeat the above steps with the label method which takes the following arguments:

- (i) Text attribute which defines string
- (ii) Bg (Background) color
- (iii) Fg (foreground) color
- (iv) Name of the parent window
- (v) Use the .pack method with the relevant padding attribute.

Radiobutton
With making use of the control variable and
button widget for selection of the given
option.

Step 1: → use the widget to impart steering
option

Step 2: → Define a function which tells the user at
the given selection need of the multiple option
available.

Step 3: → Use the configuration method along with
the label object and add the variable as an
argument within the method.

Step 4: → Now define the parent window and
define the option using control Variable.

Step 5: → Now Create an Object from the
Radiobutton method which will take the following
arguments ↳

- Positioning on Parent window
- Defining the text Variable [1, 2, 3, 4]
- Define the Variable argument.
- Corresponding Value and trigger the
given function.

Radiobutton

Code →

from tkinter import *

root = Tk()

def set():

selection = " You selected Selected the Option " + str(c)
l = Label(root, config(labtext="selected", text="selection", justify="LEFT")

o1 = Radiobutton(root, text="option 1", variable=v, value=1)

v = IntVar()

o1 = Radiobutton(root, text="option 2", variable=v, value=2, command=set)

o1 = Radiobutton(root,

text="option 3", variable=v, value=3, command=set)

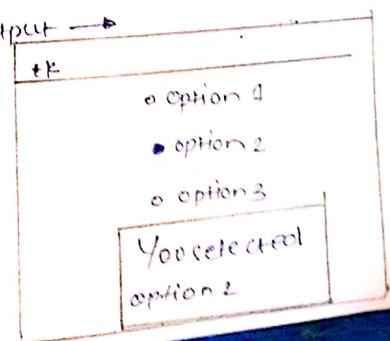
o1 = Radiobutton(root,

text="option 4", variable=v, value=4, command=set)

o1.pack()

root.mainloop()

Output →



```

Code ->
#1 Frame object
#2 Frame window import
f1 = Frame(window, width=100, height=200)
f1.pack()
frame1 = Frame(f1)
frame1.pack(side=LEFT)

frame1.pack()
leftframe = frame1.pack(side=TOP)
leftframe.pack(side=LEFT)

b1 = Button(leftframe, text="Select", activebackground="red", bg="blue")
b1.pack()

b2 = Button(leftframe, text="Modify", activebackground="blue", bg="pink")
b2.pack()

b3 = Button(leftframe, text="Add", activebackground="yellow", bg="pink")
b3.pack()

b4 = Button(leftframe, text="Exit", activebackground="green", bg="black")
b4.pack()

for b in range(4):
    b.pack(side=BOTTOM)

```

Step 6:- Pack method for the corresponding objects so created and specify the attribute as an anchor attribute.

Step 7:- Now define the label object from the corresponding method and place it on parent window.

Subsequently use pack method for this window and make use of the mainloop method.

Step 8:- Import the relevant method from tkinter library.

Step 9:- Define the Object corresponding to the object window and define the size of parent window in terms of no. of pixels.

Step 10:- Now define the frame object from the method and place it onto the parent window.

Step 11:- Create another frame object the leftframe and put it onto the Parent window on its left side.

Step 12:- Similarly define the right frame and subsequently define the button object placed onto the right frame with the attribute as text activebg and bg.

Step 13:- Now use the pack method along with the side attribute.

15
Step 15:- Similarly create the button object corresponding to modify option and put it onto frame object with side equal to right attribute set.

Step 15:- Add another button and put it on right frame object and teach it as EXIT

Step 16:- Use the Pack method for all the objects and finally use the mainloop method.

Output →

+K - □ X

blue → Select → black
blue → Modify → purple
red → Add → yellow
black → EXIT → green

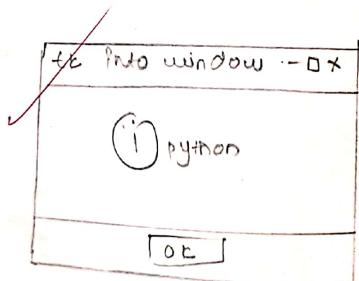
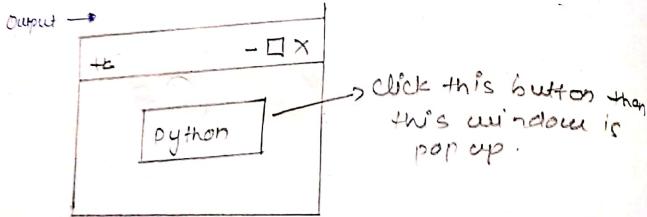
```

3) message box
# importing tkinter
from tkinter import *
import tkinter.messagebox
root = Tk()

def function():
    tkMessageBox.showinfo("Info window", "python")
    b1 = Button(root, text="python", command=function)

b1.pack()
root.mainloop()

```



Practical 5(CC)

41

Aim:-> Gui Components (Message Box)

Step 1:-> Import the relevant methods from tkinter library

Step 2:-> Import tkMessageBox

Step 3:-> Define a Parent window object along with the Parent Window.

Step 4:-> Define a function which will use tkMessageBox with showinfo method along with info window attribute

Step 5:-> Define a button with Parent window object along with the command attribute

Step 6:-> Place the button widget onto the Parent window and finally call mainloop() for triggering of the events caused above.

18

- Step 1 → Import the relevant methods from the tkinter library along with Parent window object declared.
- Step 2 → Use Parent window object along with window function for window size.
- Step 3 → Define a function main, declare Parent object and use config(), grid(), minsize() methods used as buttons and use pack() and rowspan() simultaneously.
- Step 4 → Finally define the function second and use the absolute accordingly.
- Step 5 → Define another function button along with Parent object and declare button with attributes like (text, fg, bg, command=PAUSED, font=font) - changing widget.
- Step 6 → Finally caused the mainloop() for event driven programming.

19 Multiple windows

If different button (button 1)

from Tkinter import *

root = TEC()

root.minsize(300,300)

def main():

top = TEC()

top.config(bg="black")

top.title("Home")

top.minsize(300,300)

b1 = Label(top, text="SAN FRANCISCO")

In Place of present : In Golden Gate

Pavilion 'In Learnted States In Chinatown'

In soft Tower")

b1.pack

b1 = Button(top, text="next", command=second)

b1.pack(side=RIGHT)

b2 = Button(top, text="exit", command=quit)

b2.pack(side=LEFT)

root.mainloop()

```
def second():
    top2 = Tk()
    top2.config(bg="orange")
    top2.title("About us!")
    top2.minsize(300, 300)
    l=Label(top2, text="Created by : Aayush Verma")
    l.pack()
    In for more details contact to our officia
    account"
    b1=Button(top2, text="prev", command=prev)
    b3=Button(top2, text="next", command=next)
    b3.pack(side=LEFT)
    b2=Button(top2, text="exit", command=terminate)
    b2.pack(side=RIGHT)
    top2.mainloop()
```

```
def button():
    top3 = Tk()
    top3.geometry("300x300")
    b1=Button(top3, text="flat button", relief=FLAT)
    b1.pack()
    b2=Button(top3, text="groove button", relief=GROOVE)
    b2.pack()
    b3=Button(top3, text="raised button", relief=RIDGE)
    b3.pack()
    b4=Button(top3, text="sunken button", relief=SUNKEN)
    b4.pack()
```

```
b5 = Button (top3, text=" Judge button", relief=RIDGE)
top3.mainloop()

def terminate():
    b = quit()

b5 = Button (root, text=" TOUR DETAILS", command=main)
b5.pack()

b6 = Button (root, text=" BUTTON DETAILS", command=button)
b6.pack()

root.mainloop()
```

John

```

from tkinter import *
root = Tk()
def main():
    w100 = Tk()
    w100.config(bg = "pink")
    w100.title("main")
    w100.geometry("200x200")
    l1 = Label(w100, text = "Also known as calcium")
    l1.pack()
    l1_in = Label(w100, text = "In")
    l1_in.pack()
    l1_out = Label(w100, text = "- Sources are Egg Yolk, cheese,")
    l1_out.pack()
    l1_out_in = Label(w100, text = "cheese etc")
    l1_out_in.pack()
    l1_out_out = Label(w100, text = "etc")
    l1_out_out.pack()
    l2 = Label(w100, text = "Vitamin D")
    l2.pack()
    w100.mainloop()

def se():
    w2 = Tk()
    w2.config(bg = "green")
    w2.title("2")
    w2.geometry("400x200")
    l2 = Label(w2, text = "Vitamin D")
    l2.pack()
    w2.mainloop()

def se2():
    w3 = Tk()
    w3.title("3")
    w3.geometry("400x200")
    l3 = Label(w3, text = "Also known as Tocopherol")
    l3.pack()
    l3_in = Label(w3, text = "In")
    l3_in.pack()
    l3_out = Label(w3, text = "- Sources are Almonds & Peanuts etc")
    l3_out.pack()

```

* Transversing and making use of geometry layout manager methods.

Step 1: Define a function and create a Object of the given window by using the three methods namely config, title and minsize

Step 2: Create a button Object and use the text and command attribute for triggering the given event and bind method along with internal and external padding specified similarly create another button Object which will allow application

Step 3: Define second function corresponding to second window with attributes config, title, minsize for the window object and define one button Object which will shift the focus onto the third window.

Step 4: Create third window object and in this narrate two button object for moving on to first window for restarting the process and second button for terminating.

Step 5: Define a function for termination and call the quit method and finally call the root function command and trigger method.

b3 = back()

b3 = Button (root, text = "back", command = main)

b3.pack (side = LEFT)

46

b4 = Button (root, text = "terminate", command = ter)

b4.pack (side = BOTTOM)

root.mainloop()

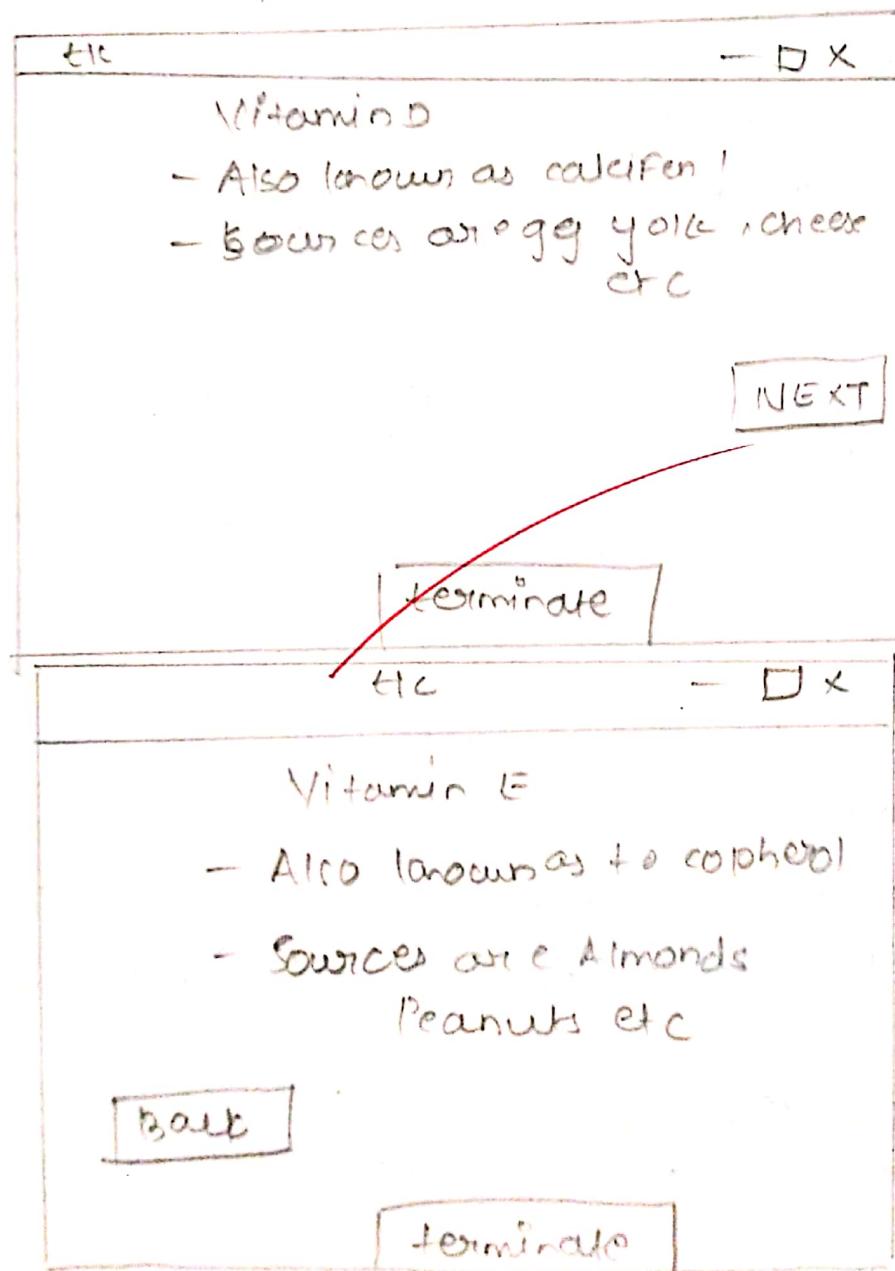
def ter ()

quit ()

b5 = Button (root , text = "know your Vitamin's",
command = main)

b5.pack ()

root.mainloop()



```

from tkinter import *
root = Tk()
root.title("python")
root.minsize(1000, 900)
root.config(bg="black")
root = Frame(root, bg="pink", height="400",
            width=280)
leftframe = Frame(root,
                  width=120)
leftframe.grid(row=0, column=0)
rightframe = Frame(root, bg="light green", height="400",
                    width=120)
rightframe.grid(row=0, column=1)
label1 = Label(leftframe, text="Photo 0", height=2, width=20)
label1.grid(row=0, column=0)
image1 = PhotoImage(file="dance.gif")
image1 = subsample(1, 2)
image2 = PhotoImage(file="dance.gif")
image2 = subsample(3, 4)
label2 = Label(rightframe, image=image2)
label2.grid(row=0, column=0)
label3 = Label(leftframe, image=image1)
label3.grid(row=1, column=0)
label4 = Label(leftframe, text="Personal Info", height=2, width=20)
label4.grid(row=2, column=0)
label4.config(bg="red")
def name():
    print("Name: ")
def has():
    pass

```

④ Displaying the image.

Algorithm: →

Step 1: → Create an object corresponding to the Parent window and use the following 3 methods • Title • Minsize • Config.

Step 2: → Create a leftframe object from the frame method and place it onto the parent window with the height, width and the bg specified. Subsequently use the grid method with the row, column, padx, pady specified.

Step 3: → Now create a rightframe object from the frame method with the width, height specified and the row and the column value should be specified.

Step 4: → Create a label object from the label method and place it onto the leftframe with text attribute denoting the original image with relief attribute used as RAISED and subsequently use grid method with row, column value specified or with some external padding values.

Step 5: → Now use the photoimage method with the file attribute specified.

Step 7 → use the set sample method with the object of the image and give the my coordinates values.

Step 8 → use the label method and position it onto the leftframe and placing the image after the label and use the grid method for sampling and use the first row for positioning in the first row.

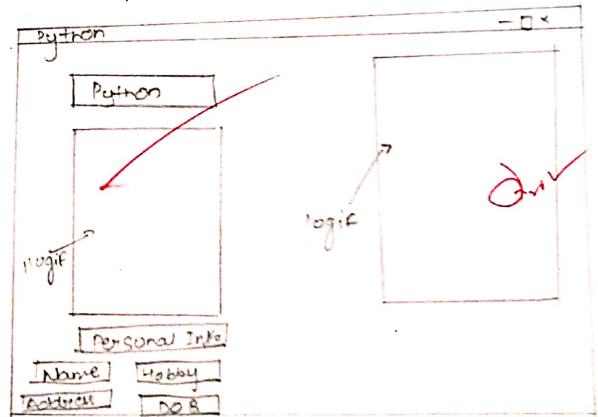
Step 9 → Create another label object positioning it onto the leftframe and specifying the image and background attribute with row and column attribute specifying as (0,0)

Step 10 → Now Create a toolbar Object from the frame method and Position it onto the leftframe with the height and width specified and position it onto the second row.

Step 11 → Now define the toolbar function for different tool bar options provided in the leftframe

Step 12 → From the label method position the text on the toolbar use the relief attribute and cause packing grid value and incorporate the internal padding as well.

```
print("Hobby: ")  
def add():  
    print("Address: Mumba")  
def del():  
    print("DOB: 24/08/1982")  
Button(toolbar, text="Name", height=1, width=16, command=  
"name").grid (row=1, column=0)  
Button(toolbar, text="Hobby", height=1, width=16,  
command=hob).grid (row=1, column=0)  
Button(toolbar, text="Address", height=2, width=16,  
command=add).grid (row=2, column=0)  
Button(toolbar, text="DOB", height=1, width=16,  
command=del).grid (row=2, column=1)  
root.mainloop()
```



Step 12 → Create the label method position it on the toolbar
with the next title as previous information and
Position it on same row but next column.

Step 13 → Now make use of ~~breakup~~ method.

Done.

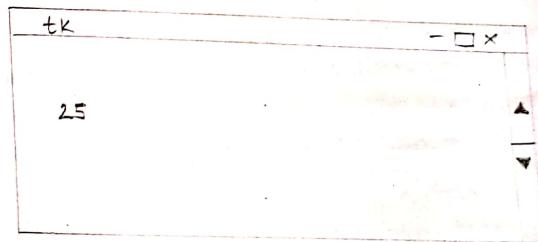
- Q) write a program to make use of spinbox
widget
- Step1:- Use of the tkinter library to import the relevant method.
- Step2:- Create the parent window Object
- Step3:- Create an object from the spinbox method and place it onto the parent window with the option specified.
- Step4:- Now use the pack method to make the object visible onto the parent window and run Mainloop:

50

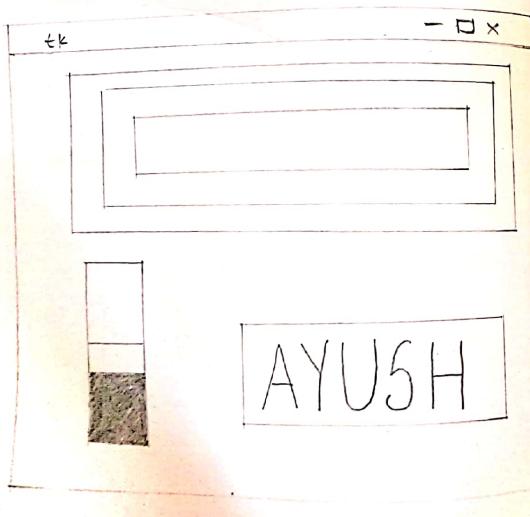
```

from tkinter import *
master = Tk()
s = Spinbox(master, from_=-10, to=25)
s.pack()
master.mainloop()

```



68 From tkinter import *
 root = Tk()
 m = Panedwindow(bg = "green", orient = VERTICAL,
 height = 100, width = 200)
 m.pack(fill = BOTH, expand = 1)
 e = Entry(m, bd = 10)
 m.add(e)
 p = Panedwindow(m, orient = HORIZONTAL)
 m.add(p)
 b = Button(p, text = "AYUSH")
 p.add(b)
 root.mainloop()



③ Paned window

- (i) Create an object from the Paned window method & use the Pack method to make this object visible.
- (ii) Now create an object from the entry widget and place it onto the paned window & use the add method.
- (iii) Similarly create an object of the paned window & add it onto the existing window.
- (iv) Create an object from the Scale method & place it onto the preceding paned window and use the add method accordingly.
- (v) Create a button widget and place it onto the Paned window & define functionality along with the button event.
- (vi) Use the Pack method & mainloop method for the corresponding events to be triggered.

16.

③ Canvas

Step1:- Create an object from the canvas widget by using the attribute height, width, background colour and the parent window Object

Step2:- Use the corresponding method for drawing the simple geometrical shapes like arc, oval & line and specify the co-ordinate values.

Step3:- Similarly use the create_line and create_oval methods along with the coordinate values and fill the attribute for specifying the colour.

Step4:- Finally use the pack & mainloop method.

Folios + kinder import *
root = Tk()

c = Canvas (root, bg = "green")

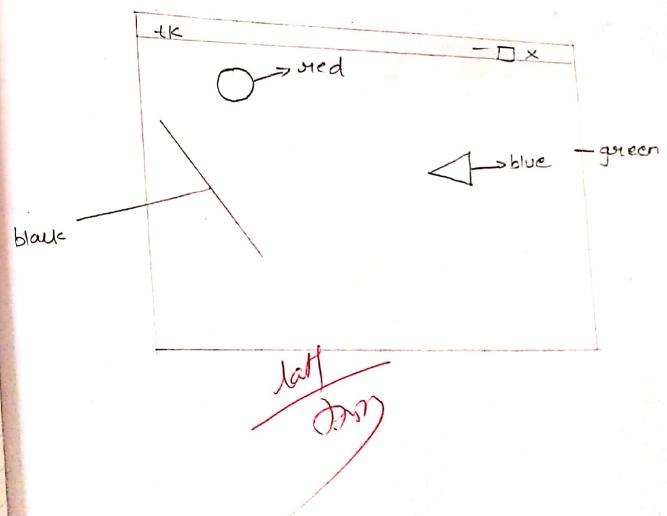
oval = c.create_oval (40, 20, 30, 40, fill = "red")

arc = c.create_arc (40, 180, 90, 300, start = 0, extent = 30, fill = "blue")

line = c.create_line (50, 60, 70, 90, fill = "black")

c.pack()

root.mainloop ()



Database Connectivity

```

56
>>>import dbm
>>>db=dbm.open("database", flag='c', create=True)
>>>db["name"] = "name"
>>>if db["name"] != None:
    print("database not empty //match")
else:
    print("database empty!" //Not-match)

>>>match
>>>db.close()

```

Step 1 → Import the (dbm) dbms library & use the open() for creating the database by specifying the name of the database along with the corresponding flag.

Step 2 → Use the object S created for accessing the given website & corresponding regular name for the website

Step 3 → Check whether the given url address matches with the regular name of the page is not equal none then display the message that particular found/match or else not found / unmatched.

Step 4 → use the close() to terminate database library.

58

Step1:- Import corresponding libraries to make particular connection to and SQLite.

Step2:- Now Create the Connection Object using SQLite's library and connect() for creating new database.

Step3:- Now Create cursor Object using the cursor() and __e from the Connection Object created.

Step4:- Now use the execute() for creating the table with the column name and respective datatype.

Step5:- Now with cursor-object use the insert statement for entering the values corresponding to different fields, corresponding the datatype.

Step6:- Use the commit() to complete the transact using the connection object.

Step7:- Use the execute Statement along with cursor-object accessing the values from the database using the select from where clause.

Step8:- Finally use the fetch() or fetchall() for displaying the values from the table using the cursor-object.

59

```
import os,sqlite3
conn = sqlite3.connect("employee.db")
cur = conn.cursor()
cur.execute("Create table dos(dName char,
                                dAge no int)
cur.execute("Insert into dos values
('Ayush', 1820), ('Yash', 1809)")
conn.commit()
cur.execute("select * from dos")
print cur.fetchall()
conn.close()
```

Output :-

[('Ayush', 1820), ('Yash', 1809)]

1/2
8/10

→ Execute () & drop table syntax part terminating
the database and finally use the close();

```
import java.sql.*;  
  
public class Database {  
    public static void main(String[] args) {  
        try {  
            // Step 1: Load the JDBC driver  
            Class.forName("com.mysql.cj.jdbc.Driver");  
  
            // Step 2: Create a connection  
            Connection conn = DriverManager.getConnection(  
                "jdbc:mysql://localhost:3306/test",  
                "root", "password");  
  
            // Step 3: Create a statement  
            Statement stmt = conn.createStatement();  
  
            // Step 4: Execute the SQL query  
            String sql = "DROP TABLE IF EXISTS employees;  
            stmt.executeUpdate(sql);  
  
            // Step 5: Close the resources  
            stmt.close();  
            conn.close();  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    }  
}
```

PROJECT

AGE CALCULATOR

Code →

```

import datetime
import tkinter as tk
window = tk.Tk()
window.geometry("620x480")
window.title("Age Calculator App")
name = tk.Label(text = "Name")
name.grid(column=0, row=1)
nameEntry = tk.Entry()
nameEntry.grid(column=1, row=1)
year = tk.Label(text = "Year")
year.grid(column=0, row=2)
yearEntry = tk.Entry()
yearEntry.grid(column=1, row=2)
month = tk.Label(text = "Month")
month.grid(column=0, row=3)
monthEntry = tk.Entry()
monthEntry.grid(column=1, row=3)
date = tk.Label(text = "Day")
date.grid(column=0, row=4)
dateEntry = tk.Entry()
dateEntry.grid(column=1, row=4)
def getInput():
    name = nameEntry.get()
    monkey = Person(name, datetime.datetime(int(yearEntry.get()), int(monthEntry.get()), int(dateEntry.get())))
    textArea = tk.Text(master = window, height = 10, width = 50)
    textArea.pack()
    textArea.insert(tk.END, monkey.info())

```

Output →

56

Age Calculator App			
Name	ayush	Date	
Year	2002	Month	
Month	09	Day	
Day	05	Calculate Age	
Heyy ayush!!!. You are 19 years old			

answer = "Heyy %monkey%!!!. You are %age% years old!!!".
 format(monkey=name, age=monkey.age))

textArea.insert(END, answer)

button = Button(window, text="Calculate Age",
 command=getInput, bg="pink")

button.grid(column=1, row=5)

class Person:

def __init__(self, name, birthdate):
 self.name = name
 self.birthdate = birthdate

def age(self):

today = datetime.date.today()

age = today.year - self.birthdate.year

return age

window.mainloop()