

No. 8 C  
Date 14/11/2023  
Page No. 1  
Topic File Handling  
Ques. 1

```
# write methods
fileObject = open("abc.txt", "w")
fileObject.write("Subject in " + "CS")
fileObject.write("\n")
fileObject.close()
```

```
# Read methods
fileObject = open("abc.txt", "r")
str1 = fileObject.read()
print("The output of read method is: ", str1)
fileObject.close()
>>> The output of read method is : Subject in CS.
```

Calculus  
Stats  
Python

```
# readline methods
fileObject = open("abc.txt", "r")
str2 = fileObject.readline()
print("The output of readline method is: ", str2)
fileObject.close()
>>> The output of readline method is : Subject in
```

```
# readlines methods
fileObject = open("abc.txt", "r")
str3 = fileObject.readlines()
print("The output of readlines() method is: ", str3)
fileObject.close()
```

```
>>> The output of readlines method is : [ 'Subject in
['\n', '\n', 'Calculus\n', 'Stats\n', 'Python'] ]
```

019

### Practical - 1

Aim - Demonstrate the use of different file accessing modes and different attributes and read methods.

Algorithm:-

Steps:-

1. Create a file object using open method and use the write mode followed by writing some content onto the file and then closing the file.
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 variables.
3. Also, use the fileObject foo finding the home of the file the filename, the file mode in the which it the opened whether the file is still open as close and finally the output of the software atted

## file attributes

020

40]. Now open the fileobject in write mode, write in the some another close subsequently them again open that is the fileobject in 'wt' mode and write content.

50. Open fileobject in read mode and displaying again in 'r' mode that is previously passed and display the output subsequently.

60. Now, open fileobject in append mode open write method, write contents close the fileobject again open the fileobject in read mode and display the append outputs.

```

# file attributes
a = fileobject.name
b = fileobject.close
c = fileobject.mode
print ("The file name is : ", a)
print ("The file mode is : ", b)
print ("file modes: ", c)

>>> The file name is : abc.txt
>>> closed: True
>>> filemode: 'a'

# Write mode
fileobject = open("abc.txt", "w")
fileobject.write ('i')

# Read mode
fileobject = open("abc.txt", "r")
y = fileobject.read()
print ("The output of read mode is : ", y)

>>> The output of read mode is : c

# File output set w+ mode
fileobject = open("abc.txt", "w+")
fileobject.write ("Element")
fileobject.close()

```

It is often necessary to make a model of a system in order to understand it.

51.  $\hat{S}_1 = \text{obj}(\text{category}, \text{minim}$   
point of output of  $\pi^+$ , 51)

foliogrid-clad)

Output of a state

H Appendix mode

```
fileobject = open ("afile.txt", "a")  
fileobject.write ("Hello Streamline")
```

*foliiget = conste*

```
fileobject open("abcr.txt","r")
```

```
52 = f.read(1).read()
```

print ("Output of append  
filled into list",

卷之三

Output of append mode: 'python data structure'

卷之三

"object-field" 1 (61: "PC")

卷之二

Mellor's "Poet"

to finding length of the different line exist within him.

*phyllotaxis* = open (aer. tax), 1121.

period ("backwash," etc.)

Print (line 1)      *Biologicals* (close)

>>> ('output : ', 'college database')

# Open the file object in read mode declare  
a variable and perform file object dot  
method tell & store the output  
variable.

consequently

Use the seek method with the fileobj argument in read and subsequently operating closing subsequently

Open fileobject with read mode and use readlines() and store the result consequently in and print the latter same file showing the length and display the length

~~tax~~

Aior. - Iteration

No. 8

Program 1:-

- Algorithm.
- (1) Define a variable of list data type containing name of fruit.
  - (2) Now iterate over that variable and print each element using next method some conditional loops statement.

Program:- To display odd no. till 6.

- ① Create a class within that define a iter method with an argument and initialize the value and return that values.
- ② Define the next method with an argument the value and conditional statements to check whether the variable in 1st step is smaller than it or not.
- ③ Increase the variable's value by 2 and its at the else to stop iterations.

- ④ Create an object of given class and pass the object in the method. Use while conditional loop to print the method.

```
i = ["Apple", "Banana"]
my_fruit = Blu(i)
for k in i:
    print(k)
```

Output:

Apple  
Banana

2.) Code:-

```
class add:
    def __new__(m):
        m + value = 1
        return m
    def - next -(m):
        if m.value < 15:
            m.value + 2
        return m.value
    else:
        raise StopIteration
```

Output

1

3

5

7

9

11

12

15

3.) Code:-

```
class Fact:
    def __init__(m):
        m.value = 1
    def - next -(m):
        return m
```

```
def - next -(m):
    Facto = 1
    for i in range(1, m.value + 1):
        Facto = Facto * (i)
    print("The Factorial of", m.value, "is")
    if (m.value == 7):
        raise StopIteration
```

```
y = inter80(start(1))
while True:
    next(y)
```

**Output:**

The factorial of 1 is 1  
The factorial of 2 is 2  
The factorial of 3 is 6  
The factorial of 4 is 24  
The factorial of 5 is 120  
The factorial of 6 is 720

**q) Code:****Class power:**

```
def __iter__(self):
```

```
pow_num = 1
pow_num *= int(input("Enter number: "))
```

```
return pow
```

```
def __next__(self):
```

```
if pow_num == 1:
```

```
    nu = pow_num * pow_num
```

```
    pow_num *= 1
```

```
    return nu
```

```
else:
```

```
    raise StopIteration
```

**Output:**

```
y = class(power(1))
while True:
    print(next(y))
```

```
32
9
16
```

**Program 3: Factorial of numbers till 10****Algorithm:-**

- ① Create a class within that define a iter method on argument, initialize a value and return it.
- ② Define a next method with an argument.
- ③ Create a variable factorial with value 1 and use for conditional loop to calculate factorial use the value of variable declared in step 1.
- ④ Print the fact's value and increase the value and increase the value of variable in

step 1 by 1

conditional statement to value stop to iteration if the range of variable in step 1

is bigger than 10

⑤ Create a object this class and pass it to show method and use while loop to use next method.

next method.

**Program 4:- Power of numbers taken from user.**

- ① Create a class and define a iter method with a argument, initialize a value and return it and also take a input from user.
- ② Define a next method with an argument.
- ③ If the initialized value is bigger than 5 then the stop iteration or else return the multiplic

argument of itself.

- ④ Create a object of given class and pass it to iter method and use while loop to use

Program 5 - To display the numbers in range 1 to 10 with algorithm.

Create a class and define idet method with an argument, initialize a value & Now define next method. check whether the last assigned value is

```
class Range1 {
    int num = 1;
    def idet():
        if num <= 10:
            num += 1
            return num
        else:
            raise StopIteration
}
def next(g):
    g.idet()
    while True:
        print(next(g))

```

Output:

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

while True:

try:

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

break

except ValueError:

print("Enter numeric value.")

**Ques - Program to demonstrate exception handling.**

- 1] Write a program using the exception method of the `math` module to handle the error.
- Step 1 - Use the try block and except the input using the `raw_input` method and convert it into the integer datatype and subsequently document the block.
- Step 2 - Use the except block with the exception name as value error and display the appropriate message if 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 - Within the try block open the file using the `write` mode and write mode and write some content on the file.
- Step 2 - Use the except block with 10 error and display the message regarding missing file at incompatibility of the mode use the block to display a message that the operation is carried out successfully.

```

f0=open("abctxt","w")
f0.write("Ashish Thakur")
f0.close()
print("File writing in the file")

else:
    print("Operation carried out successfully")
    print("File closed")
    print("File writing in the file")
    print("Operation carried out successfully")

```

```

def add_(n):
    assert len(n) == 0:
    print("list is empty")

```

val 1 = []  
print(add(val 1))

Output:

list is empty

# Program -

def acceptage():

```

age = int(input("Enter age:"))
if age > 30 and age < 16:

```

raise ValueError

return age

valid = False

while not valid:

```

    age = acceptage()
    valid = True

```

```

except ValueError:
    print("Not a valid age")

```

Output:

Enter age = 4

Not a valid age

Enter age : 18

~~Step 1~~

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

Step 1: Define a function which accept an argument and check using the assert statement whether the given list is empty list and accordingly print the message.

Step 2: Use the function and in the body of program and define certain element in list and take input appropriate value.

4) Write a program to check the range of the age in the students in given class and if the age do not fall in given range then the value will exception otherwise return the valid number.

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 do return the age else set the value None exception.

Step 3 - Define the while loop to check whether the expression holds true . use the try block to except the age of student and formulate the looping condition.

import re  
pattern = 'Hello' # pattern  
sequence = 'Finds sequence'; found 'Hello'  
if re.match(pattern, sequence):  
 print("Matched pattern")

```
else:  
    print("NOT FOUND")
```

>>> matched pattern found by  
# numerical value (separation)

import re  
pattern = '123'

string = 'Hello123, nowdy789, 15nowdn'

output = re.findall(pattern, string)

print(output)

>>> ['123', '789', '15']

```
# split()
```

import re  
pattern = '123'

string = 'Hello123, nowdy789, 15nowdn'

output = re.split(pattern, string)

print(output)

>>> ['Hello', 'nowdy', '15nowdn']

Topic: Regular Expression:

Step 1:- Import re module declare pattern and sequence use match method match with declare argument if arguments matched than print the same otherwise print pattern 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.

Steps:- Import re module declare pattern with meta character use the split() and print the output.

Step 3:- import re module declare pattern replace the accordingly declare pattern replace the blank space with no-space. Use sub() with 3 arguments and print the string without space.

Step 5 - Import re module declare a sequence use search method for finding subsequent use the group() with dot operator search() give memory location using a It will show up the matched string q.

# no-space:

```
import re
string = 'abc def ghi'
pattern = r'\st'
replace = ''
v1 = re.sub(pattern, replace, string)
print(v1)
```

>>> abcdefghi

# group()

```
import re
sequence = 'python is an interesting language'
v = re.search('A python', sequence)
print(v)
v1 = v.group()
print(v1)
```

>>> <\_sre.SRE\_Match object at 0x0281DF00>  
python

# verifying the given set of phone number

```
import re
list1 = [18004567891, 91115673210, 7865432981,
         9876543201]
for value in list1:
    if re.match(r'[8-9]\d{13}[0-9]\d{9}'):
        value or len(value) == 10):
            print("Criteria matched for cell number!")
    else:
        print("Criteria failed!")
```

Criteria matched for cell number -

```

# vowels
import re
s1 = 'plant is life overall'
output = re.findall(r'[aeiouAEIOU]w+', s1)
print(output)
>>> ['P', 'l', 'e', 'o', 'v', 'e', 'r', 'a', 'l', 'l']

# host & domain
import re
seq = 'abc.cs@edu.com, xyz@gmail.com'
pattern = r'([w.]+)@([w.]+)'
output = re.findall(pattern, seq)
print(output)
>>> ['abc.cs@edu.com', 'xyz@gmail.com']

# counting of first letters
import re
s = 'mr.a,ms.b,ms.c,mr.t'
p = r'([ms])|([mr])'
o = re.findall(p, s)
print(o)
m = 0
f = 0
for v in o:
    if (v == 'ms'):
        f = f + 1
    else:
        m = m + 1
print("No. of males is:", m)
print("No. of females is:", f)
>>> [('mr', 'ms'), ('ms', 'mr')]
('No. of males is: 1, 2')
('No. of females is: 1, 2')

```

- 031
- Step7- Import re module declare a string use the module with.findall() for finding the vowels in the string and declare the same.
  - Step8- Import re module declare the host and domain name declare pattern for separating the host and domain name use the.findall() and print the output respectively.
  - Step9- Import re module enter a string use pattern display only two elements of the particular string use.findall() declare two variable with initial value as zero use for condition and subsequently use the if condition check whether condition satisfy add upto the else statement increment value and display the values subsequently.

~~Note~~  
~~Jan~~

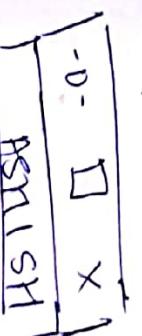
Code:-

```

from tkinter import *
root = Tk()
l = Label(root, text="YASHVI")
l.pack()
root.mainloop()

```

Output:-



Code:-

```

from tkinter import *
root = Tk()
l = Label(root, text="Python")
l.pack(side=LEFT, ipady=40)
l = Label(root, text="Java")
l.pack(side=TOP, ipady=40)

```

```

m = Label(root, text="C")
m.pack(side=LEFT, padx=20)
m2 = Label(text="Shell", bg="blue", fg="black")
m2.pack(side=LEFT, ipadx=50)

```

- Step 8:- side=LEFT, ipady=40  
 Side=LEFT, padx=20  
 Side=TOP, ipadx=50  
 Side=TOP, ipady=50.
- Step 1:- Use the tkinter library for importing the feature of text widgets.
- Step 2:- Create a variable from text method and place it on parent window.
- Step 3:- Use the pack method along with the object created from text method.
- Step 4:- Use the mainloop method for triggering of corresponding events.
- Step 5:- Use the tkinter library for importing the feature of text widgets.
- Step 6:- Create an variable from text method and position it onto parent window.
- Step 7:- Use the pack method along with the object created from text method and use the parameter parameter.

Step 8:- side=LEFT, ipady=40  
 Side=LEFT, padx=20  
 Side=TOP, ipadx=50  
 Side=TOP, ipady=50.

Step 9:- Use the main loop method for triggering of corresponding events.

Step 10:- Now repeat the above step with the label method with take the following arguments:

1. Text attribute which define string.
2. Bg (background) color
3. Fg (fore ground) color
4. Name of the parent window.
5. VBC paint method with the relevant padding attribute

var set of Radiobutton

Step 1: Import relevant method from the tkinter library

Step 2: Define a function and define a variable which will keep track of the option selected

Step 3: Now the config method along with the label object and use the attribute for displaying the deletion mode

Step 4: Now define the parent window attribute the function definition & define the control variable

Step 5: Now define an object corresponding to the radio button with following attribute.

- ① Parent window
- ② Text attribute
- ③ Variable
- ④ Value
- ⑤ Commands.

Step 6: Likewise define the five different radiobutton and use the pack method simultaneously by defining the anchor attribute

Step 7: Now define the label object and pack it on the parent window.

1. From tkinter import \*

root = Tk()  
var = IntVar()  
R = Radiobutton(root, text = "option No 1", variable = var, value = 1)

R1 = Radiobutton(root, text = "option 2", variable = var, value = 2)

label = Label(root)

R1.pack(anchor = "w")  
label.pack()  
root.mainloop()

You selected option 1!

tk - □ ×

option 1

You selected option 1!

from tkinter import \*  
import tkmessagebox

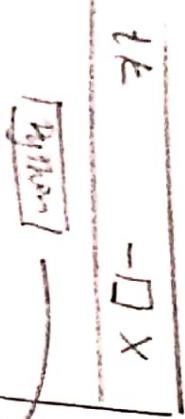
def function():

tk.messagebox.showinfo("info window", "Python")

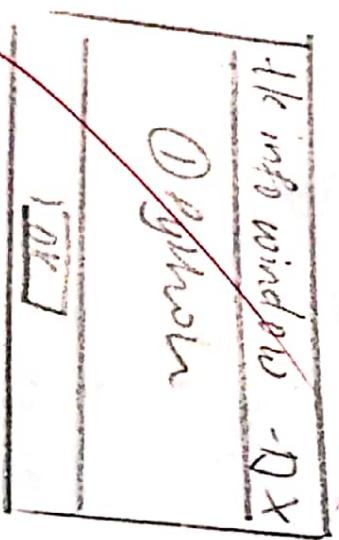
python = "python"

b1 = Button(root, text=python, command=function)

Output :-



On clicking this window appears.



Message box

→ Step 1:- Import relevant method from tkinter library.

Step 2:- Import tk message box

Step 3:- Define a parent window object along with a parent window.

Step 4:- Define a function which will use tkmessage box with showinfo method along with its window attribute.

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

Step 6:- Place the button widget onto the parent window and finally call mainloop().

triggering of the event called above

Relief style:-

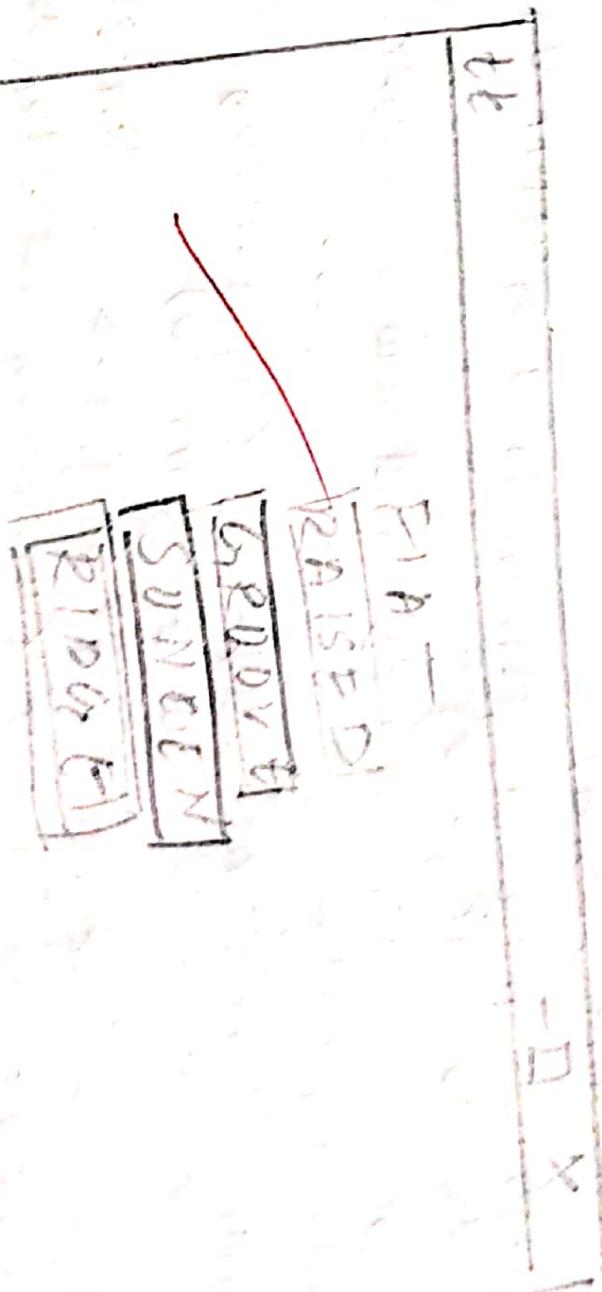
Step 1:- Use the button objects with the following attributes:-

1. The parent window
2. Text attribute
3. Relief

Step 2:- Use the corresponding pack method for the respective button objects and trigger the corresponding event.

Step 3:- Finally use the mainloop method.

Output:-



```
from tkinter import *
root = Tk()
root.title("Tk")
b1=Button(root, text="FLAT", relief=FLAT)
b1.pack()
b1=Button(root, text="RAISED", relief=RAISED)
b1.pack()
b2=Button(root, text="SUNKEN", relief=SUNKEN)
b2.pack()
b3=Button(root, text="RIDGE", relief=RIDGE)
b3.pack()
b4=Button(root, text="GROOVE", relief=GROOVE)
b4.pack()
b5=Button(root, text="sunken", relief=SUNKEN)
b5.pack()
root.mainloop()
```

"... render image "

037

```
root = Tk()
root.title("a Python")
root.geometry("1000x900")
leftframe = Frame(root, bg="black", height="400", width="400")
leftframe.config(bg="black")
leftframe.grid(row=0, column=0, rowspan=0, columnspan=0)
rightframe = Frame(root, bg="light green", height="500", width="500")
rightframe.grid(row=0, column=1, rowspan=0, columnspan=0)
label1 = Label(leftframe, text="A Photo", 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(5, 2)
label2 = Label(leftframe, image=image1).grid(row=0, column=0,
                                             padx=20, pady=10)
label3 = Label(rightframe, image=image2).grid(row=0, column=1,
                                             padx=20, pady=10)
```

Step 2:- Create a leftframe 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 rows, column, padx, pady specified.

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

It shows - Create a label object from the label method and it onto the leftframe with text attribute denoted by the original image and relief attribute denoted by RAISED value and subsequently we used the Subsample method with row, column value specified as (5, 2) with same external padding value.

Step 5:- Now use the Image method with the file at specified.

Step 6:- Use the sub sample method with the object of the image and give the x,y co-ordinate of values.

Step 7:- Use the label method and position it onto the left frame and placing the image after the sampling and use the grid method for the positioning in the first row.

Step 8:- Create another label object positioning it onto rightframe and specifying the image and bg attribute with row and column attribute specify it as (0,0).

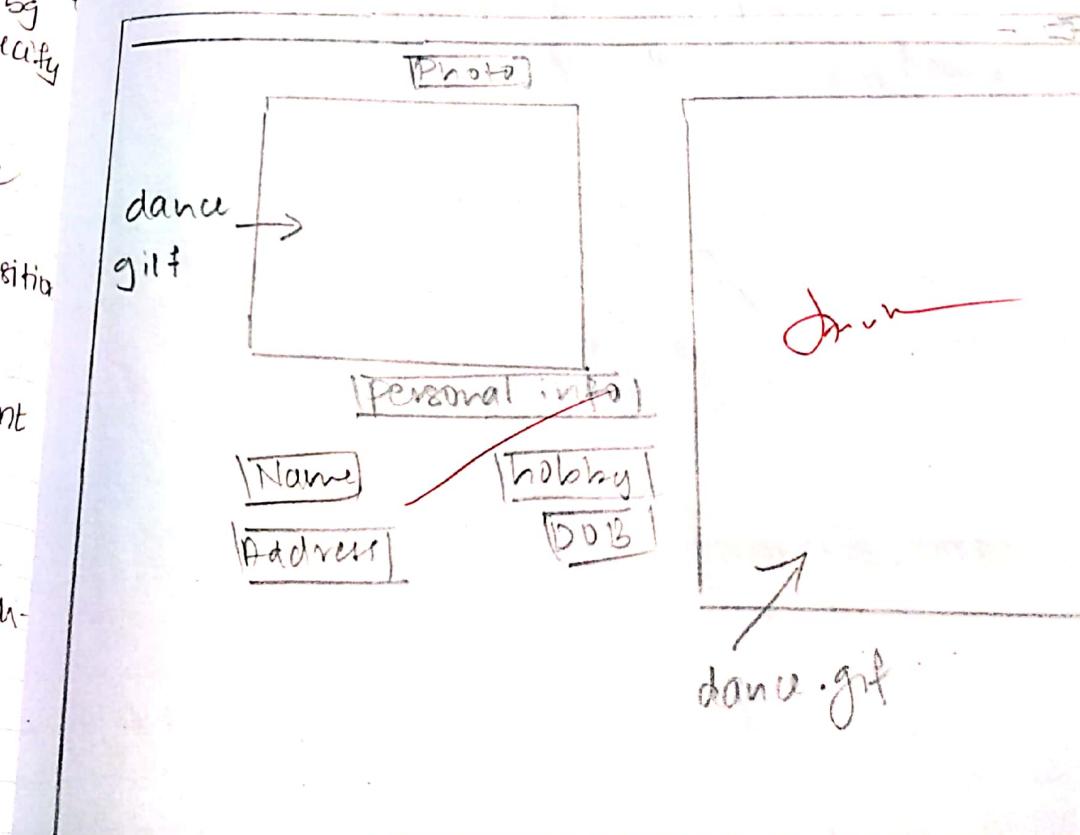
Step 9:- Now create a toolbar object from the frame method and position it onto the leftframe with the height and width specified and position it onto the second row.

Step 10:- Now define the various junction for different tool bar option provided in the leftframe

Step 11:- From the label method position the text on the toolbar use the relief attribute and corresponding grid value and incomplete the internal padding as well:

038

```
def add():
    print("Address: Mumbai")
def dob():
    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=dob).grid(row=1, column=1)
Button(toolbar, text="Address", height=1, width=16, command=add).grid(row=2, column=0)
Button(toolbar, text="DOB", height=1, width=16, command=dob).grid(row=2, column=1)
root.mainloop()
```



## E Spin box

Step 1:- Use the parent window widget.

Step 2:- Create an object from the spinbox and place it onto the parent window and the option specified.

Step 3:- Now, use the pack method to make the object visible onto the parent window the call the mainloop method.

from tkinter import \*

root = Tk()

S = Spinbox(root, from\_=0 to=10)

S.pack(anchor=N)

root.mainloop()

040

# Paned window

from tkinter import \*

root = Tk()

P = Panedwindow()

P.pack(expand=1, fill='both')

c = Entry(P, bd=1, bg='black')

P.add(c)

P1 = Panedwindow(P, orient=VERTICAL)

P.add(P1)

top = Scale(P1, orient=VERTICAL)

P1.add(top)

root.mainloop()

Output:-



Paned window:-

Step 1:- Create an object from the paned window method and use the pack method with the attribute fill and expand.

Step 2:- Create an object from the label method and put it onto the paned window with the attribute and use the add method to enter the new object.

Step 3:- Similarly, create a second paned window & it onto and add it onto the first paned window with orientation specified

Step 4:- Now create another label object and put onto the second paned window object onto the second pane

Step 5:- Trigger mainloop.

#1:-

```
>>> import dbm
>>> db = dbm.open("database", flag = 'c', mode = 'w')
>>> db["name"] = "name"
>>> if db["name"] != None:
    print("database not empty! //match!")
else:
    print("database empty! //Not Match!")
>>> match
>>> db.close()
```

Practical - b.

Aim: Database connectivity

#2:-

~~Step:- Import the (DBM) dbm library & use the open() for creating the database by specifying the name of the database along with the corresponding tag.~~

~~Step2:- Use the object so created for accessing the given website & corresponding regular name from the websites.~~

~~Steps:- Check whether the given url address match with the regular name of the page is it not equal to none then display the message "not particular found" otherwise~~

~~Step 4:- Use the close() to terminate database library.~~

#2

step 1:- Import corresponding library, OS & the  
make database library

SQLite -3

>

Step 2:- Now create the connection object & the content  
SQLite-3 library of the database.  
for executing will be  
created.

conn = conn.cursor()

Output:-  
conn.execute('create table dos(Names char  
char execute('insert into dos values ('Zau  
1313), ('Makar', 1815))  
conn.commit()  
cur.execute('select \* from dos')  
print(cur.fetchall())  
conn.close()

Step 3:- Now use the execute() for creating  
table with the column name & setting  
the datatype.

Step 4:- Now with userin-object use the insert  
Statement for entering the Values  
corresponding to different field  
corresponding the datatype.

Step 5:- Now with userin-object use the insert  
Statement for entering the Values  
corresponding to different field  
corresponding the datatype.

Step 6:- Use the transaction commit() to complete  
the transaction using the connection  
object.

Note  
Ans

t('Zauri', 1813), ('Makar', 1815)]

045

Step 5:- Finally use the fetch() or fetchall() for displaying the value from the table using the cursor-object:

Step 9:- Execute() & drop table syntax for terminating the database & finally use the close()