

Q3.

```
fileobj = open("abc.txt", "w")
fileobj.write("Science subjects" + "\n")
fileobj.write("Physics\nChemistry\nMaths\n")
fileobj.close()

fileobj = open("abc.txt", "r")
# read()
str1 = fileobj.read()
print("The output of read method:", str1)
fileobj.close()

>>> ('The output of read mode:', 'Science subjects\nPhysics\nChemistry\nMaths\n')

# readline()
fileobj = open("abc.txt", "r")
str2 = fileobj.readline()
print("The output of readline method:", str2)

# readlines()
fileobj = open("abc.txt", "r")
str3 = fileobj.readlines()
print("Output:", str3)
fileobj.close()

>>> ('Output:', ['Science subjects\nPhysics\nChemistry\nMaths'])
```

Shot on OnePlus

By Astley

Step4 :- now open the file obj in write mode, write another content close subsequently then again open the file obj in 'w+' mode that is the update mode & write content.

Step5 :- open file obj in read mode display the updated written content & 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 write contents. Close the file obj again open the file obj in read mode & display the 'append' output.

```
# fileobj - mode
f = fileobj.open("file mode", r)
f = fileobj.open("file mode", "w")
f = fileobj.open("file mode", "r+")
f = fileobj.open("file mode", "a")
f = fileobj.open("file mode", "a+")
# write mode
```

```
fileobj = open("abc.txt", "w")
fileobj.write("DBMS")
fileobj.close()
```

# read mode

```
fileobj = open("abc.txt", "r")
s1 = fileobj.read()
print("Output of r:", s1)
fileobj.close()
```

# append mode

```
fileobj = open("abc.txt", "a")
fileobj.write("data structures")
fileobj.close()
fileobj = open("abc.txt", "r")
s3 = fileobj.read()
print("Output!", s3)
fileobj.close()
```

>> ("Output!", "Ashay data structures")

```
# write mode
fileobj = open("abc.txt", "w")
fileobj.write("DBMS")
fileobj.close()
```

```
# read mode
fileobj = open("abc.txt", "r")
s = fileobj.read()
print("Output:", s)
print("Output of read mode:")
>>> ("Output of read mode:")
>> ("Output of r:", "DBMS")
```

```

SS
# tcl()
fileobj = open("abc.txt", "r")
pos = fileobj.readline()
print("tcl():", pos)
fileobj.close()
>>> tcl(), b

# 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")
st2 = fileobj.seek(0, 1)
print("seek(0,1) is:", st2)
fileobj.close()
>>> seek(0,1) is:, None

```

If finding length of different lines exist within lines

```

fileobj = open("abc.txt", "r")
stod = fileobj.readlines()
print("output : ", stod)
for line in stod:
    print([len(line)])
fileobj.close()
>>> output: ['binary data structures'] >> 20

```

23

step 7 - open the fileobj in read mode declare a variable & perform fileobj.readline() method & store the output consequently in variable.

step 8 - use the seek method with the argument within opening, the fileobj in read mode and closing subsequently.

step 9 - open file obj write read mode also use the readline method & store the output consequently in & print the same for counting the length use the for conditional statement & display the length.

Day 17

AIM :- To demonstrate the use of iterators &  
iterables.

ALGORITHM :- GRANGER

Def a iter method with an argument & initialize  
the value, & return that value.

Def the next method with an argument & compare  
the upper limit by using a conditional  
statement.

QUESTION :- (Q3) -

a) Define a class method with an argument which takes the value of  $y$  and returns the result of  $y + 1$ .

b) Define the next method with an argument to compare the upper limit by using break statement. Increment the value by 1 unless object of the given class passes this object in the `isodd` method.

SOURCE CODE :-

```
26
class odd:
    def __init__(self):
        self.num = 0
    def isodd(self, num):
        if num < 20:
            num = self.num + 1
            self.num = 2 * num + 1
            return num
        else:
            print("Upper limit reached")
    def next(self, num):
        if num < 20:
            num = self.num + 1
            self.num = 2 * num + 1
            return num
        else:
            print("Upper limit reached")
y = user(odd())
for i in range(1, 10):
    print(next(y))
    print(isodd(y))
```

Output:-

```
1
3
5
7
9
11
13
15
17
19
21
```

18  
 class myclass:  
 def \_\_init\_\_(self):  
 pow = 1  
 pow.n1 = int(input("Enter a number"))  
 pow.n2 = int(input("\nEnter the power limit"))  
 return pow  
 def next\_\_(self):  
 if pow.n1 <= pow.n2:  
 num = pow.n1 \* pow.n1  
 pow.n1 += 1  
 return num  
 else:  
 raise StopIteration  
 y = iter(myclass())  
 while True:  
 print(next(y))

Output:-

Enter a number = 2  
 Enter the power limit = 4

2	4
4	16

Shop on OnePlus  
By Adm

27  
ALGORITHM (POWER)  
 Def the \_\_iter\_\_ method with 3 arguments. Initalise and fix arguments as classfile. The other two arguments as "Enter the number" & "Maximum limit of power" respectively.  
 Define the next() with an argument by compale it. By using conditional statement, increment the value by 1.  
 If value less than or equal to maximum limit, then print the value with end = .  
 Now Create the object of the given class & pass this object in the \_\_iter\_\_ method. If we use the while conditional statement to print

Dr. N. S. N.

23/2/19 5.8 Function - 8

Ques :- To demonstrate exception handling

Q. write a program using the exception block of the native arithmetic errors:

Step 1 :- Use the try block & except block using the raw input method i.e. in the integer datatype give directly decimal one value.

2) Use the except block with the exception as value error & display the appropriate message of the suspicious code in front of the block.

QUESTION

def func1:

```
    a = int(input("Enter a number:"))
    print("Valid number")
    break
except ValueError:
    print("Not a valid number! Try again")
```

INPUT

>>> Enter a number: 17.20

Not a valid number! Try again

Enter a number: 19

valid number!

performing final, basic, testing

QUESTION

def func1:

```
    a = int(input("Enter a number:"))
    print("Valid number")
    break
except ValueError:
    print("Not a valid number! Try again")
```

INPUT

>>> Enter a number: 17.20

Not a valid number! Try again

Enter a number: 19

valid number!

performing final, basic, testing

write a program for accepting the file in a given  
mode & use the environmental error as an  
exception of the given input.

within the try block open the file using the write  
mode and write some content on the file

Use the except block with 10 error & display the  
message regarding missing of the file or incompat-  
ibility of the mode , use the else block to display  
a message that the operation is carried out  
successfully .

Q5:

- a) write a program using the recused list if the list element are empty
- b) Define a function which accepts an array and check using the assert statement that the given list is empty list & if true return the message.
- c) close the function & in the body of function return certain elements in list & take appropriate action.



ShutterOnePlus  
By Akash

SOURCE CODE :-

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

OUTPUT :-

```
list is empty
```

write a program to check the range of the age of the students in a given class. If the age do not fall in given range else the value error exception otherwise return the valid no.

Define a function which will accept the age of the student from the standard input

use the if condition to check whether the input age falls in the range and so return the age else throw the value error exception.

Define the while loop to check is another the boolean expression holds true use the try block to check the age of student e.g terminate the looping condition.

Use except with value error by printing the message not a valid range.

#### PRACTICAL - 4

Ques 1 → Demonstrate the use of regular expression.

**THEORY :-** Regular expression represents a sequence of characteristic which mainly used for finding & replacing the given pattern in a string. This we import re module & can use of regular expression with following functionalities.

- Searching a given string
- Finding a string
- Finding a string into particular substring
- Replacing part of string

Ques 2 → Write a program of regular expression to numeric by alphabetic value from a given string.

Ques 3 → Now apply string & print it find all & display one output.

Ques 4 → Id is used for matching all decimal numbers whereas is used to match non decimal digits.

Import re

```
string = "Hello1234abc456"
result = re.findall("10", string)
print(result)
```

```
result = re.findall("(\\w)", string)
print(result)
```

```
result = re.findall("[1-4]", string)
print(result)
```

32

wire a regular expression for finding the matching at the beginning of given sequence:

algorithm:

import re module & apply a string:

use search with "IA python" & string as two parameters.

now display the output.

now we if conditional statement for user to know whether match is found or not.

Q.8.

- 3) Write a regular expression to check whether given mobile no. start with 8 or 9 & total length 10.

Algorithm :-

- 1) Import re module & apply a string of number.
- 2) Now use if conditional statement to find number starts with 8 or 9 & the total length should be of length 10. we match() function of statement to find the match in given string.
- 3) We use if conditional statement to check, we have to match or not if we have we group () to display the output & if we can display incorrect mobile no.

18...  
# Code  
import re  
string = "Python is no intended language"  
result1 = re.findall("look", string)  
result2 = re.findall("wo", string)  
print(result1)  
print(result2)

>>> Output

>>>

35  
use a regular expression for extracting a word from given string along with space character in between the word. It subsequently extract the word without space between.

Algorithm :-

Step 1:- Import re module.  
Step 2:- Applying a string.

we "look" to extract word along with space &  
we "wo" to extract word without space.

now display the output.

- 26
- » write a regular expression for extracting first word from a string
- Algorithm :-
- 1) Import the module & apply re module
  - 2) Use findall() in which use "re" as parameter to find first word of the sentence "with all parameter to find desired string".
  - 3) Display the output

code :-

```
input str
string = "python is an interpreted language"
result 1 = re.findall("^\w+", string)
result 2 = re.findall("^\w+\s+\w+", string)
print(result 1)
print(result 2)
```

Output :-

```
>>> ['python']
>>> ['python', 'language']
```

Write a regular expression for extracting the date in format dd-mm-yyyy by using the.findall() over the string has following format:

VISHNU 24-12-2019

Algorithm

Import re module & apply a string.

We use.findall method given 12-12-2019

12-12-2019

58

- Q) Write a re for extracting strings from email-id  
① Username from email-id  
② Hostname from email-id  
③ Both username & hostname from email

#### Algorithm

- 1) import re module & supply string
- 2) use findall() to find username & hostname from email-id.
- 3) use "\w+" for username and "\w+\.\w+" for host name & use "[\w+]" for both as parameter in findall()
- 4) display the output

# code

import re

```
string = "abc @ tcs . edu"
result1 = re.findall("\w+", string)
result2 = re.findall("\w+\.\w+", string)
result3 = re.findall("[\w%]+", string)
```

```
print(result1)
```

```
print(result2)
```

```
print(result3)
```

#### Output

```
>>> ['abc']
['abc']
>>> ['tcs . edu']
['tcs . edu']
>>> ['abc @ tcs . edu']
['abc @ tcs . edu']
>>> [ 'abc' , ' @ ' , 'tcs' , ' . ' , 'edu' ]
['abc', '@', 'tcs', '.', 'edu']
>>> [ 'abc' , ' @ ' , 'tcs' , ' . ' , 'edu' ]
['abc', '@', 'tcs', '.', 'edu']
>>> [ 'abc' , ' @ ' , 'tcs' , ' . ' , 'edu' ]
['abc', '@', 'tcs', '.', 'edu']
```

## Q1 :- GUI components:

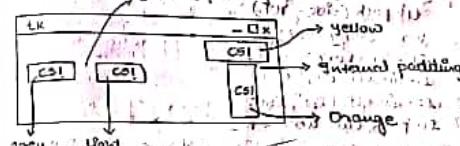
- Step 1 : Use the tkinter library for importing the features of the text widget.
- Create an object using the Tk() .
- Create a variable using the widget label & use the get method.
- Use the mainloop() for triggering of the corresponding above mentioned events.
- Use the tkinter library for importing the features of the text message window.
- Create a variable using the from\_ the text method and position it on the parent window.

- 1) Use the pack() along with the object from the root c) & use the parameters
- 1) side = LEFT , padx = 20
  - 2) side = LEFT , pady = 30
  - 3) side = TOP , ipadx = 40
  - 4) side = TOP , ipady = 50
- 2) Use the mainloop() for the triggering of corresponding events.
- 3) Now repeat above steps with the label which takes the following arguments:-
- 1) Name of the parent window.
  - 2) Text attribute which def. the string.
  - 3) The background color.
  - 4) The foreground fg & then use the with a relevant padding attributes.

```
l1 = Label (root, text = "cst", bg="orange", fg="black",
            font = 10)
```

```
l1.pack (side = TOP, pady = 50)
```

```
root.mainloop ()
```

OUTPUT :- 

grey background  
blue border  
white internal padding  
orange border

## PRACTICAL - SB

AIM :- GUI components

#1 : import tkinter  
import tkinter.messagebox  
import random  
from tkinter import \*  
from tkinter import messagebox  
from random import randint  
from tkinter import font

import tkinter as tk  
from tkinter import \*  
from tkinter import messagebox  
from random import randint  
from tkinter import font

support the relevant methods from the tkinter library create an object with the parent window.

use the parent window object along with the geometry function for specifying pixel size of the parent window.

now define a function which tells the user about the given selection made from multiple option available.

now define the parentwindow and define the option with control variable.

use the listbox() and insert option on the parent window along with the pack() which specifying anchor attribute.

Create an object from radiobutton which will take following arguments; parentwindow object, test variable which will take the values option no. 1, 2, 3; variable argument, corresponding value & trigger the function declared.

⑤ Finally make use of the mainloop() along with friend object.

# 2:

① Import relevant method from the tkinter  
module.

② Create a friend object called parenting to the  
parent window.

③ Use the geometry() for sizing of the window.

④ Create an object and use the scrollbars()

⑤ Use the pack() along with the scrollbar  
object with side and fill attributes.

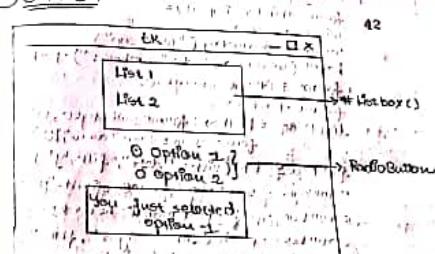
⑥ Use the mainloop with the friend object

⑦ In the scrollbars, use left and right  
arrow keys to scroll up and down the  
content of the scrollbars.

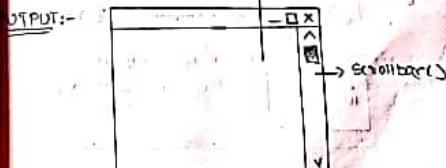
⑧ Finally make use of the mainloop() along  
with friend object.

Shot on OnePlus  
By Asim

## OUTPUT



Scrollbar()  
from tkinter import \*  
root = Tk()  
root.geometry("500x500")  
s = scrollbar()  
s.pack(side="right", fill="y")  
root.mainloop()



#3 :-

import the relevant libraries from the tkinter. Create a corresponding object of the parent window.

use the geometry manager with the pixel size, (680x500) or any other suitable pixel value

use the label widget along with the parent object created and use the pack().

use the frame widget along the parent object created by user the pack().

use the listbox method along with the attributes like width, height, font etc. To create a listbox methods object use pack() for the same.

use the scrollbar() with an object use the attribute of vertical; then configure the same with object created from the scrollbar() & use pack().

trigger the event using mainloop().

- 1.4
- 1) Import relevant methods from tkinter library
  - 2) Define the object corresponding the parent window & define the size of parent window in terms of no. of pixels
  - 3) Now define the frame object from the window class & add it to the parent window.
  - 4) Create another frame object named as left frame and put it on the parent window on its LEFT side.
  - 5) Similarly define the RIGHT frame and add it to the parent window with the attribute active background & foreground.
  - 6) Now add the pack() along with the side attribute.
  - 7) Similarly create the button object corresponding to the modify operation put it into frame on side = "right".

```

1.5
from tkinter import *
window = Tk()
window.geometry("600x400")
frame = Frame(window)
frame.pack()
leftframe = Frame(window)
leftframe.pack(side = "left")
rightframe = Frame(window)
rightframe.pack(side = "right")
b1 = Button(frame, text = "Modify", activebackground =
            "yellow", bg = "black")
b1 = Button(frame, text = "Select", activebackground =
            "red", bg = "blue")
b3 = Button(frame, text = "Add", activebackground =
            "blue", bg = "red")
b4 = Button(frame, text = "Exit", activebackground =
            "red", bg = "green")
b1.pack(side = "left", padx = 20)
b2.pack(side = "right", padx = 20)
b3.pack(side = "bottom", pady = 20)
b4.pack(side = "top")

```

45

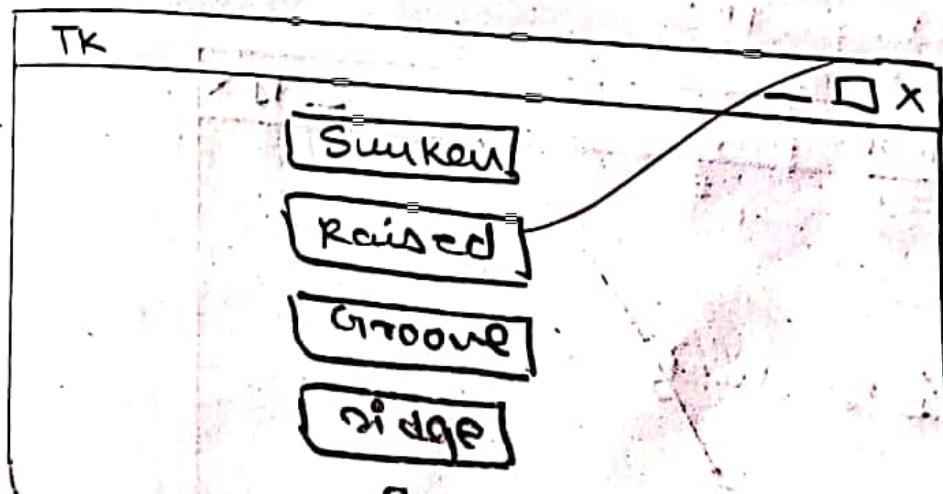
Create another button object & place it on the right frame & label the button as ADDHILL

ADD another button & put it on the top of frame and label it as EXIT if you want to exit

use the pack() simultaneously for all the objects & finally use the mainloop()

# code

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



→ Showwarning ( )

- ① Define a function which will use the showwarning() from the messagebox library.
- ② The attributes which a given method takes will specify the string one related to the message displayed (1) corresponding to the message.
- ③ Now create an object from the object by place it on the parent window with the title of the button attribute to execute the relevant fun.
- ④ Terminate the paragraph by using the mainloop()

from tkinter import \*

top = Toplevel()

messagebox

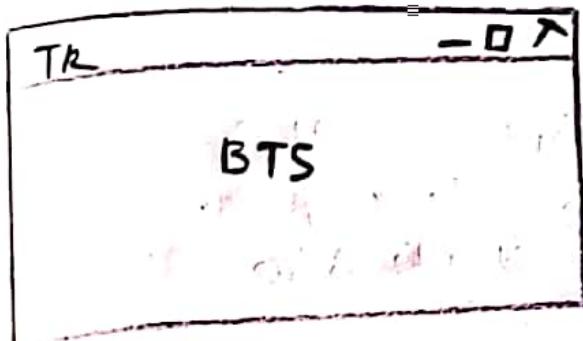
def msgb():

messagebox.showinfo("Ashley", "Great Being")

b1 = Button(top, text="BTS", command=msgb)

b1.pack()

top.mainloop()



askyesno()

Define a function, you will use the askyesno derived from the messagebox library.

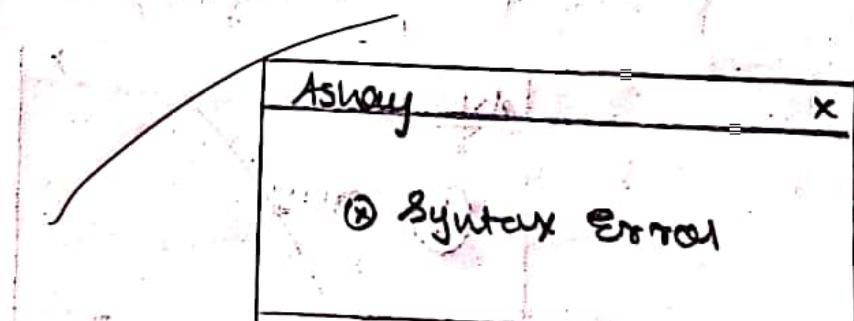
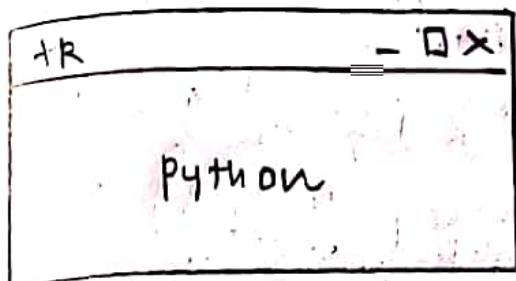
and attributes which a given method takes of the window (1) corresponding to the message displayed.

Now create an obj. from the button method & place it to the parent window (1) of the button object specified & finally, use one command attribute to execute function.

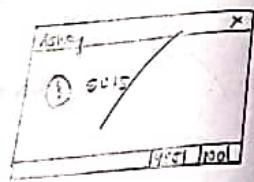
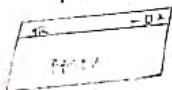
- ① Terminate the program by mainloop.

Final

```
from tkinter import *  
top = Toplevel()  
msgbox()  
def msgbox():  
    messagebox.showerror("Ashay", "Syntax error")  
b1 = Button(top, text="python", command=msgbox)  
b1.pack()  
top.mainloop()
```



```
from tkinter import *  
top = Tk()  
messagebox*  
def msg():  
    messagebox.showinfo("Alert", "SUC")  
bt = Button(top, text="pycharm", command=msg)  
bt.pack()  
top.mainloop()
```

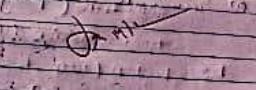


### QUESTION (1)

81

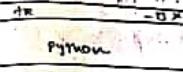
define a function which will take two arguments  
from the messagebox library  
and calculate, which argument is greater than specify  
and 2 string?  
① related to subseq displayed

and create an object from the button method  
if place it on the press ipudow of CAM 3  
generate the program by using mainloop()



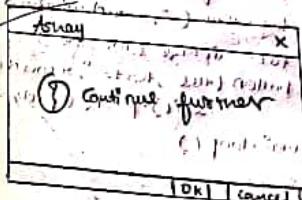
- 18
- askforcancel()
- ① Define a function which will ask user two questions and messagebox "library".
  - ② The attribute which gives method click...contains the 2 string.
  - ③ Related to the title -
  - ④ Corresponding to the messagebox displayed.
  - ⑤ Now create an obj of this class button method place it onto the parent window.
  - ⑥ Terminate the program by calling mainloop.

```
from tkinter import *
top = Tk()
messagebox = 52
def msgbox():
    messagebox = Toplevel()
    messagebox.title("Library")
    b1 = Button(messagebox, text="OK", command=ok)
    b1.pack()
    b2 = Button(messagebox, text="Cancel", command=cancel)
    b2.pack()
    top.mainloop()
    messagebox.destroy()
    return
```



6) Python - message box

```
def ok():
    print("OK Clicked")
    messagebox.destroy()
    top.mainloop()
def cancel():
    print("Cancel Clicked")
    messagebox.destroy()
    top.mainloop()
    return
```



7) Python - message box

5D]:

Write a program to move from one window to another window with the help of button widget.

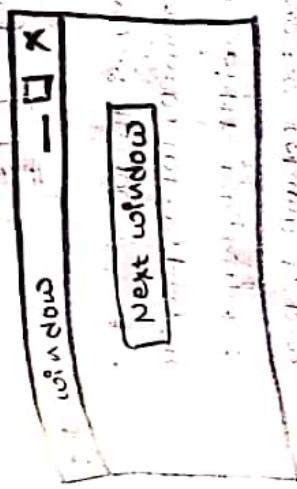
Define a function and create a parent window object and use the config · title · min size ·

Now define a button object and place it on to the parent window with a suitable title & use the command attribute to call the next function using the grid method specifying an external padding.

Now define a function corresponding to second window and create another parent window obj with the method config · title & min & again place the button obj calling the next function

Similarly create the fun & use the button widget and finally create a fun which will terminate the aggregate function using quit method().

### Computer monitor



Front view of monitor screen



Back view of monitor



Side view of monitor

Computer monitor is a device which displays the output of a computer.

It consists of a screen, a frame, a base, a handle, and a power cord.

It is used to display the output of a computer.

It is used to display the output of a computer.

It is used to display the output of a computer.

```

final class Main {
    static void main(String[] args) {
        JFrame frame = new JFrame("Pygame");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setSize(800, 600);
        frame.setLayout(new GridLayout(2, 2));
        JButton button = new JButton("Pygame");
        JButton button2 = new JButton("Black");
        JButton button3 = new JButton("Image");
        JButton button4 = new JButton("Clock");
        frame.add(button, "North", 0, 0);
        frame.add(button2, "South", 0, 1);
        frame.add(button3, "East", 1, 0);
        frame.add(button4, "West", 1, 1);
        frame.setVisible(true);
    }
}

class Pygame extends JPanel {
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawImage(pygameImage, 0, 0, null);
    }
}

class Black extends JPanel {
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.setColor(Color.BLACK);
        g.fillRect(0, 0, 800, 600);
    }
}

class Image extends JPanel {
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawImage(pygameImage, 0, 0, null);
    }
}

class Clock extends JPanel {
    public void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawString("Time: " + System.currentTimeMillis(), 100, 100);
    }
}

```

Shot on OnePlus  
N100

- 57
- write a program to insert an image in the frame window using the other widget.
  - create the parent window object & use the method titleConfig and the widget which is object.
  - create an object from the frame methods & it is the parent window object with the width, height and its color specified & use the grid() along with the row & column attribute & (0,0) with some external padding.
  - similarly create the right frame object from the frame class with same & col attribute matching their values (0,1).
  - use the label() to the parent window object corresponding to left frame & set its relief attribute specific by setting col value (0,0) in the grid().
  - similarly create the label for right frame & use one title of the row col value (0,1).
  - use the photo() with the file attribute specify by subsequently sub example() for specifying the image obj.

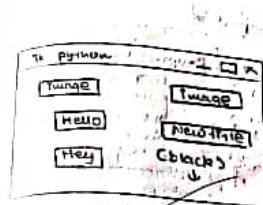
72

4) Now use the label() using the `label` function and one unique attribute of the `label` class value specified in the grid. (3) Similarly release use using label() function similarly release use using `color` attribute and `size` of `line` obj. Note the `line` attribute can be same by color with value `(0,0)`.

5) Now define a `gui` using the `grid` statement which may caused an defining the `grid`.

6) Create an object from `frame` (1) `button` toolbox (5) place it on the left frame with `value` as `(2,0)`.

7) Now create one `button` obj (5) place it on the toolbox object with the `text` of `command` similarly specify.



Spinbox :-

Abstract Inherit

use the spinner library to import the relevant method ~~int i=0; i<10; i++~~

create a new parent window object

create an object from the spinbox method and place it on to the parent window with the option specified

now use the pack method to make the object visible onto the parent window & called the mainloop method

~~the object must be option scrollable~~

## 78 Practical - 5e]

### \* Paned Window

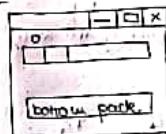
- 1) Create an object from the paned window and use the pack () to make this object visible.
- 2) Now create an object from the entry window & place it onto the paned window and use the add method. Similarly create an object of label window.
- 3) Create an object from the widget & place it onto the paned window by using the add method accordingly.
- 4) Create a button widget & place it onto the paned window defined a functionality along with the button widget.
- 5) Use the pack () & mainloop () for the corresponding event to trigger.

Shot on OnePlus  
By Aman

### -- SOURCE CODE --

```
from tkinter import
w1 = Panedwindow()
w1.pack(fill=BOTH, expand=1)
w1.pack(fill=BOTH, expand=1)
e = Entry(w1, text="left pane", bd=5)
e.pack(side=LEFT, expand=1)
m2 = Panedwindow(w1, orient=VERTICAL)
w1.add(m2)
m2.pack(expand=1)
top = Scale(m2, orient=HORIZONTAL)
top.pack()
m2.add(top)
bottom = Button(m2, text="bottom pane", command=top)
m2.add(bottom)
m2.pack()
m2.mainloop()
```

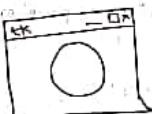
### -- OUTPUT --



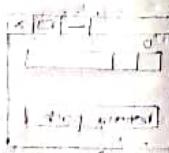
### # Source Code

```
FOR OVAL  
from tkinter import*  
master = Tk()  
c = Canvas(master, bg = "white", height = 600, width = 400)  
c.pack()  
c.create_oval(50, 50, 400, 400)  
mainloop()
```

### Output



### # for



Shot on OnePlus  
By Ashay

### # Canvas Widget

- 1 Create an object from the canvas widget by using the attribute `Canvas(master, bg="color", height="height", width="width")`.
- 2 Use the corresponding methods for drawing the simple geometrical shapes like `create_oval` etc. and specify the co-ordinate values.
- 3 Similarly, use the `create_rectangle` along with the co-ordinates values of the fill attribute for specifying the color of the rectangle.
- 4 Finally use the `pack()` function.

The following is the python program with its output:

For further details refer to the notes on Tkinter.

## QUESTION - 6:

### Database Connectivity

- a) imported the relevant libraries for the database connection and the import `sqlite3` function.
- b) now create an object for opening the connection for the given database.
- c) further create an object corresponding to the execution of the different query like `insert`, `update`, `delete` etc.
- d) use the cursor object so created for inserting the structure of the database & the values in database.
- e) use the execute method for implementing the select clause for fetching the info.
- f) also use the fetch all method along with the object for displaying the values on the screen.



Shishu eGuruPlus

By AMITY

### Source Code

```
import os,sqlite3  
conn = sqlite3.connect("student.db")  
cur = conn.cursor()  
cursor.execute("Create table info (ENO int, Name  
text, Estdate int); cursor object at 0x00F62F0>  
cursor.execute("Insert into info values (0, "Ashay", "0-08-  
(08)/04/06-1999"))  
<sqlite3::cursor object at 0x02F62F20>  
cursor.execute("select DOB from info")  
<sqlite3::cursor object at 0x02F62F20>  
conn.commit()  
cursor.execute("select Name from student")  
print(cursor.fetchall())  
cursor.close()
```

### OUTPUT

```
( "Ashay", "Male", "1852" )
```