

Q3

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

```
file obj = 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:", 'Science Subj
```

```
# readlines()
```

```
fileobj = open ("abc.txt", "r")
```

```
str3 = fileobj.readlines()
```

```
print("Output:", str3)
```

```
fileobj.close()
```

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

Shot on OnePlus

By Ashay

Step 4: now open the file obj in write mode, with  
several other content close subsequently  
Then again open the file obj in 'wt' mode  
that is the update mode & write content

Steps: open file obj in read mode display the  
update written content & close. Open  
again in 'r+' mode with parameter  
passed and display the output subseq.

Step 5: 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 'appending'  
output.

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

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

```
# r+ mode
fileobj = open("abc.txt", "r+")
s1 = fileobj.read()
print("Output of r+", s1)
fileobj.close()
>>> ('Output of r+', 'Ashay')
>>> ('Output of r+', 'Ashay')
```

```
# append mode
fileobj = open("abc.txt", "a")
fileobj.write("data structure")
fileobj.close()
fileobj = open("abc.txt", "r")
s3 = fileobj.read()
print("Output!", s3)
fileobj.close()
>>> ('Output!', 'Ashay data structure')
```

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

```
# read mode
fileobj = open("abc.txt", "r")
s = fileobj.read()
print("Output!", s)
>>> ('Output!', 'Ashay')
```

- Step 7 - open the file obj in read mode declare a variable & specify fileobj . tell method & store the output consequently in variable.
- Step 8 - use one seek method with the argument with opening the file obj in read mode and closing subsequently.
- Step 9 - open file obj write read mode also use the readlines method & store the output consequently in & print the same for counting the length use the for conditional statement & display the length.

J. Telli

```

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

# seek()
fileobj = open("abc.txt", "r")
st = fileobj.seek(0,0)
print("seek(0,0) is:", st)
fileobj.close()
>>> seek(0,0) is: 0

fileobj = open("abc.txt", "r")
st1 = fileobj.seek(0,1)
print("seek(0,1) is:", st1)
fileobj.close()
>>> seek(0,1) is: 0

# finding length of different lines exist within lines
fileobj = open("abc.txt", "r")
stat = fileobj.readlines()
print("output : ", stat)
for line in stat:
    print([len(line)])
>>> output: ['binary data structure']
>>> len([len(line)])
>>> 20

```

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

ALGORITHM:- (RANGE)

Def a iterator 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.

ALGORITHM :- ( ODD ) :-

- 1) Def an iter method with an argument & initialize the value & return the value.
- 2) Def the next method with an argument & compare the upper limit by using if condition statement. Increment the value by 2.
- 3) now create an object of the given class & pass this object in the iter method.

SOURCE CODES -

```
class odd:
    def __iter__(self):
        self.num=1
        return self
    def __next__(self):
        if self.num<=20:
            num=self.num
            self.num+=2
            return num
        else:
            raise StopIteration
y=iter(odd())
print(next(y))
```

OUTPUT:-

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

```

class myiter:
    def __iter__(self):
        pow.n = 1
        pow.ni = int(input("Enter a number"))
        pow.nl = int(input("Enter the power limit"))
        return self
    def next(self):
        if pow.n <= pow.nl:
            num = pow.n * pow.n
            pow.n += 1
            return num
        else:
            raise StopIteration
g = iter(myiter())
while True:
    print(next(g))

```

Output:-

Enter a number = 2  
 Enter the power limit = 4

2  
 4  
 8  
 16

### ALGORITHM:- (POWER)

- p Def the iter method with 3 argument : 2 intialize the first argument as 1. Initialize the other two arguments as "Enter the number" & "Maximum limit of power" respectively.
- ⇒ Define the next () with an argument g & complete it by using a conditional statement, increment the value by 1.
- ⇒ Now create the object of the given class & pass this object in the iter method & use the while conditional statement to print.

Dr. Ashay

27/12/19 TS PRACTICAL - 8

Ques :- To demonstrate exception handling

- 1) write a program using the exception block of the `int` variable arithmetic error.
- Step 1 :- Use the `try` block & `except` the `input` using the raw input method & convert it into integer datatype & correctly delineate the block.
- 2) Use the `except` block with the `exception` raw as `ValueError` & display the appropriate message if the suspicious code is part of the `try` block.

SOURCE CODE :-

28

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

OUTPUT :-

```
>>> Enter a number: 17.2
Not a valid number! Try again
Enter a number: 17
Valid number!
```

PERFECTED THE BOILER PLATE

Shot on OnePlus  
By Ashay

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 incompatibility of the mode ,use the else block to display a message that the operation is carried out successfully .

Q.S:

- 3) write a program using the classed list if the list element are empty.
- 4) Define a function which accepts an argument and check using the assert statement whether the given list is empty list & accordingly return the message.
- 5) close the function & in the body of program define certain elements in list & take an appropriate action.

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 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 also use the value error exception.

Define the while loop to check whether the boolean expression holds true use the try block to check the age of student eg terminate the looping condition.

Use except with value error & print the message not a valid range

#### PRACTICAL - 4

- 9/1/19 18...
- Aim :- Demonstrate the use of regular expression.
- Theory :- Regular expression represents the sequence of characteristic which is mainly used for finding & replacing the given pattern in a string. To this we import re module & common usage of regular expression includes following functionalities.
- Searching a given string.
  - Finding a string.
  - Replacing part of string for another.
  - Finding a string into another.
  - Substring.
  - Replacing part of string for another.
- D) Write a program of regular expression regarding numeric & alphabetic value from a given string.
- Q) Now apply string & pattern to find all () & display the output.
- 2) \d is used for matching all decimal digit whereas \w is used to match non-decimal digits.

# Import re  
32  
string = "Hello 1234 abc 456"  
result = re.findall ("(\d)", string)  
result = re.findall ("(\w)", string)  
print (result)  
print (result)  
# Output  
[('1234', '456')]  
[('Hello', 'xyz')]

write 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 use if conditional statement for user to know whether match is found or not.

88

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

Algorithm :-

- 1) Import re module & apply a string of mobile number.
- 2) Now use for conditional statement to find if number starts with 8 or 9 & the total length should be of length 10. we match() inside for statement to find the match in given string.
- 3) we use if conditional statement to know 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 an intended language.  
result1 = re.findall("look", string)  
result2 = re.findall("wo", string)  
print(result1)  
print(result2)
```

>>> Output

>>>

35

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

algorithm

step 1: Import re module  
step 2: Import re module & apply a string:  
we find all () to extract a word from given string.

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

now display the output.

Jn M



Shot on OnePlus  
By Asha

Q8

- 1) write a regular expression for extracting last word from a string.
- 2) Algorithm :-
- 3) Import the module & apply a string.
- 4) use findall() in which use "w\w" as parameter to find first word of string & "w\w" all parameter to find last string.
- 5) Display the output.

# code

```
import re
string = "python is an intended language"
result 1 = re.findall("w\w", string)
result 2 = re.findall("w\w", string)
```

36

# Output

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



Shot on OnePlus

By Ashay

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 to use 12/22/2023  
12/24/23

58

- 1) write a code for extracting the information.
  - 2) Username from email-id
  - 3) Hostname from email-id
  - 4) Both username & hostname from email-id
- Algorithm
- 1) Import re module & supply string.
  - 2) Use.findall() to find username & hostname both from email-id.
  - 3) Use "\w+" for username & "[\w+\.\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+\.\w+]", string)

print(result1)
print(result2)
print(result3)
```

OUTPUT

```
>>> ['abc']
('abc',)
>>> ['tcs . edu']
('tcs . edu',)
>>> ['abc', 'tcs . edu']
('abc', 'tcs . edu')
('tcs . edu',)
```

HIM :- 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 text method.

Use the mainloop() for triggering of the corresponding above mention events.

Use the tkinter library for importing the features of the text message window.

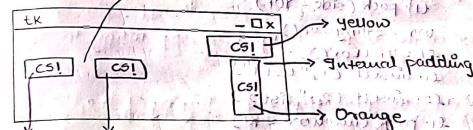
Create a variable using the from the text method and position it on the parent window.

- 3) Use the pack() along with the object created from the text class & use the parameters:
    - 1) side = LEFT , padx = 20
    - 2) side = LEFT , pady = 30
    - 3) side = TOP , ipadx = 40
    - 4) side = TOP , ipady = 50
  - 4) Use the mainloop() for the triggering of corresponding events.
  - 5) 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 pack() with a relevant padding attributes.

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

```
l4. pack Cside = TOP , pady = 50 )  
root . mainloop ( )
```

Output :- External padding  $(TOP = 20\%)$  down side



grey blue light blue = blue; yellow = yellow; white = white; blue = blue; brown = brown; "knicks" = outfit, wear =

*Chrysanthemum* (Chrysanthemum) *discolor* (L.) *Thunb.* var. *discolor* (L.) *Thunb.*

19. *Leptopeltis* sp. (Copeia) (1962)



# Shot on OnePlus

By Ashay

## PRACTICAL - 5B

AIM :- GUI components

#1: import the relevant methods from the tkinter library create an object with the parent window

use the parent window object along with the geometry function declaring specific 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 , text variable which will take the values option no. 1,2,3; variable argument, corresponding value & trigger the function declared .

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

#2:

  - 1) Import relevant method from the tkinter library.
  - 2) Create a parent object corresponding to the packed windows.
  - 3) Use the geometry() for laying of the windows.
  - 4) Create an object and use the scrollbar()
  - 5) Use the pack() along with the scrollbar object with side and fill attributes.
  - 6) Use the mainloop with the parent object.

## OUTPUT

42

Tkinter window

Listbox()  
List 1  
List 2

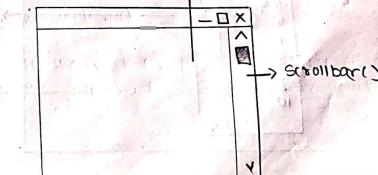
Option 1  
Option 2

You just selected

Scrollbar

mainloop()

## OUTPUT



#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 execute a listbox methods object use pack () for the same .

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

Trigger the event using mainloop .

- 84.
- 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 method & place it on to the parent window.
  - 4) Create another frame object termed as the left frame and put it on the parent window on its LEFT side.
  - 5) Similarly define the RIGHT frame and subsequently define frame with the attribute (as details of background & foreground)
  - 6) Now use the pack() along with the side attribute to similarly locate the button object corresponding to the MODIFY operation put it into frame object on side = "right".

44.

```

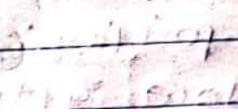
from tkinter import *
window = Tk()
window.geometry("600x500")
frame = frame(window)
frame.pack()
leftframe = frame(window)
leftframe.pack(side="left")
rightframe = frame(window)
rightframe.pack(side="right")
b2 = Button(frame, text="Modify", activebackground=
            "yellow", fg="black")
b1 = Button(frame, text="Select", activebackground=
            "red", fg="blue")
b3 = Button(frame, text="Add", activebackground=
            "blue", fg="red")
b4 = Button(frame, text="Exit", activebackground=
            "red", fg="green")
b1.pack(side="LEFT", padx=20)
b2.pack(side="right", padx=30)
b3.pack(side="bottom", pady=20)
b4.pack(side="top")

```

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

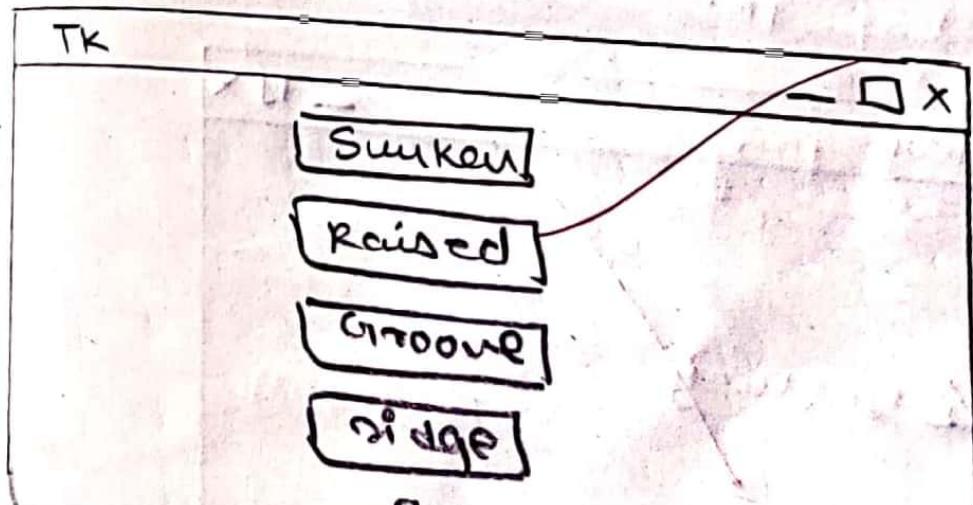
ADD another button & put it on the top of frame and label it as EXIT

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

from tkinter import \*

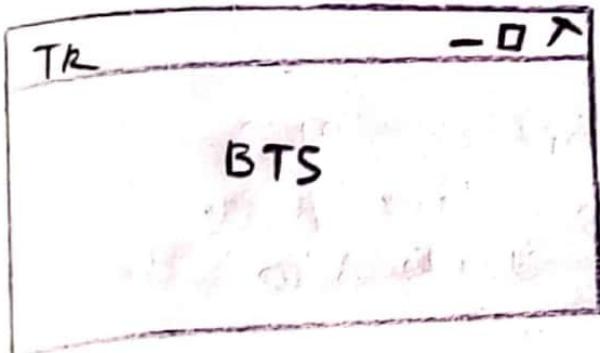
top = Toplevel()  
messagebox.showinfo("Ashley", "Great Being")

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

button.pack()

top.mainloop()

48



`askyesno()`

Define a function which will use the `askyesno` derived from the messagebox library.

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

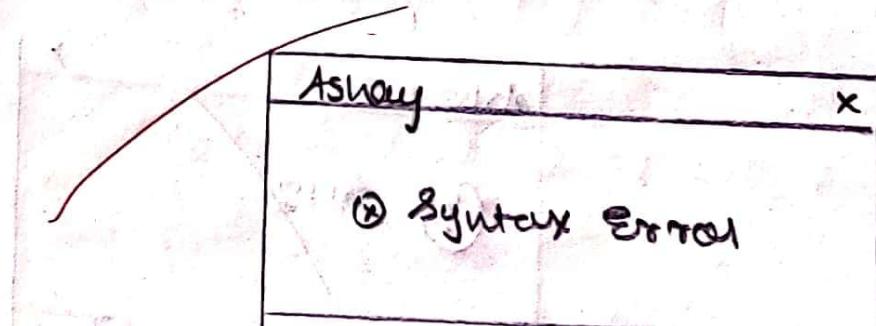
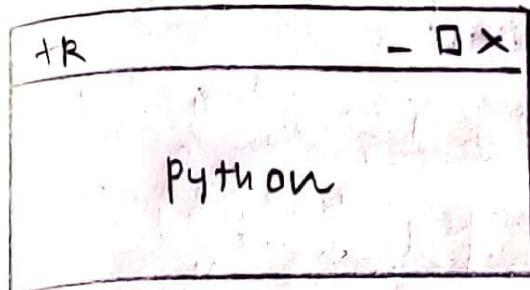
Now create an obj. of from the button method & place it to the parent method with little of the button object specified & finally use the command attribute to execute function.

④ Terminate the program by mainloop.

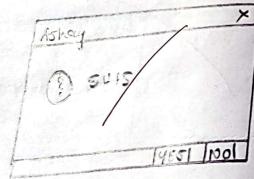
*Final*

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

50



```
from tkinter import *  
top = Tk()  
msgbox()  
def msgbox():  
    messagebox.askquestion("Ashay", "SUS")  
b1 = Button(top, text="python")  
b1.pack()  
top.mainloop()
```



### askquestion()

51

Define a function which will use the askquestion() from the messagebox library.  
and attributes, which a given method take will specify  
① related to sub msg displayed  
Note create an object from the button method  
if place it on the parent window  
Terminate the program by calling mainloop().

Jan 11



Shot on OnePlus  
By Ashay

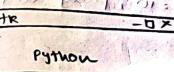
18

askokcancel

- ④ Define a function which will use the askokcancel from the messagebox library.
- ⑤ The attribute which gives method title with specifying the 2 string.
  - ⑥ related to the title
  - ⑦ corresponding to the messagebox displayed.
- ⑧ Now create an obj from the button method & place it onto the parent window.
- ⑨ Terminate the program by calling mainsleep.

from tkinter import \*  
top = Tk()  
messagebox  
def msgb():  
 messagebox.askokcancel("Ashey", "Continue")  
b1 = button(top, text="Python", command=msgb)  
b1.pack()  
top.mainloop()

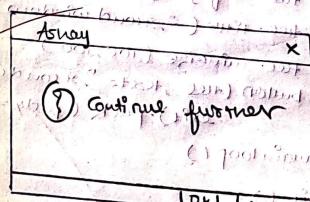
52



(1) Python - 100%

(2) 100% - 100%

(3) 100% - 100%



(1) 100% - 100%

(2) 100% - 100%

(3) 100% - 100%

(4) 100% - 100%

(5) 100% - 100%

(6) 100% - 100%

(7) 100% - 100%

(8) 100% - 100%

(9) 100% - 100%

(10) 100% - 100%

(11) 100% - 100%

(12) 100% - 100%

(13) 100% - 100%

(14) 100% - 100%

(15) 100% - 100%

(16) 100% - 100%

(17) 100% - 100%

(18) 100% - 100%

(19) 100% - 100%

(20) 100% - 100%

(21) 100% - 100%

(22) 100% - 100%

(23) 100% - 100%

(24) 100% - 100%

(25) 100% - 100%

(26) 100% - 100%

(27) 100% - 100%

(28) 100% - 100%

(29) 100% - 100%

(30) 100% - 100%

(31) 100% - 100%

(32) 100% - 100%

(33) 100% - 100%

(34) 100% - 100%

(35) 100% - 100%

(36) 100% - 100%

(37) 100% - 100%

(38) 100% - 100%

(39) 100% - 100%

(40) 100% - 100%

(41) 100% - 100%

(42) 100% - 100%

(43) 100% - 100%

(44) 100% - 100%

(45) 100% - 100%

(46) 100% - 100%

(47) 100% - 100%

(48) 100% - 100%

(49) 100% - 100%

(50) 100% - 100%

(51) 100% - 100%

(52) 100% - 100%

(53) 100% - 100%

(54) 100% - 100%

(55) 100% - 100%

(56) 100% - 100%

(57) 100% - 100%

(58) 100% - 100%

(59) 100% - 100%

(60) 100% - 100%

(61) 100% - 100%

(62) 100% - 100%

(63) 100% - 100%

(64) 100% - 100%

(65) 100% - 100%

(66) 100% - 100%

(67) 100% - 100%

(68) 100% - 100%

(69) 100% - 100%

(70) 100% - 100%

(71) 100% - 100%

(72) 100% - 100%

(73) 100% - 100%

(74) 100% - 100%

(75) 100% - 100%

(76) 100% - 100%

(77) 100% - 100%

(78) 100% - 100%

(79) 100% - 100%

(80) 100% - 100%

(81) 100% - 100%

(82) 100% - 100%

(83) 100% - 100%

(84) 100% - 100%

(85) 100% - 100%

(86) 100% - 100%

(87) 100% - 100%

(88) 100% - 100%

(89) 100% - 100%

(90) 100% - 100%

(91) 100% - 100%

(92) 100% - 100%

(93) 100% - 100%

(94) 100% - 100%

(95) 100% - 100%

(96) 100% - 100%

(97) 100% - 100%

(98) 100% - 100%

(99) 100% - 100%

(100) 100% - 100%

(101) 100% - 100%

(102) 100% - 100%

(103) 100% - 100%

(104) 100% - 100%

(105) 100% - 100%

(106) 100% - 100%

(107) 100% - 100%

(108) 100% - 100%

(109) 100% - 100%

(110) 100% - 100%

(111) 100% - 100%

(112) 100% - 100%

(113) 100% - 100%

(114) 100% - 100%

(115) 100% - 100%

(116) 100% - 100%

(117) 100% - 100%

(118) 100% - 100%

(119) 100% - 100%

(120) 100% - 100%

(121) 100% - 100%

(122) 100% - 100%

(123) 100% - 100%

(124) 100% - 100%

(125) 100% - 100%

(126) 100% - 100%

(127) 100% - 100%

(128) 100% - 100%

(129) 100% - 100%

(130) 100% - 100%

(131) 100% - 100%

(132) 100% - 100%

(133) 100% - 100%

(134) 100% - 100%

(135) 100% - 100%

(136) 100% - 100%

(137) 100% - 100%

(138) 100% - 100%

(139) 100% - 100%

(140) 100% - 100%

(141) 100% - 100%

(142) 100% - 100%

(143) 100% - 100%

(144) 100% - 100%

(145) 100% - 100%

(146) 100% - 100%

(147) 100% - 100%

(148) 100% - 100%

(149) 100% - 100%

(150) 100% - 100%

(151) 100% - 100%

(152) 100% - 100%

(153) 100% - 100%

(154) 100% - 100%

(155) 100% - 100%

(156) 100% - 100%

(157) 100% - 100%

(158) 100% - 100%

(159) 100% - 100%

(160) 100% - 100%

(161) 100% - 100%

(162) 100% - 100%

(163) 100% - 100%

(164) 100% - 100%

(165) 100% - 100%

(166) 100% - 100%

(167) 100% - 100%

(168) 100% - 100%

(169) 100% - 100%

(170) 100% - 100%

(171) 100% - 100%

(172) 100% - 100%

(173) 100% - 100%

(174) 100% - 100%

(175) 100% - 100%

(176) 100% - 100%

(177) 100% - 100%

(178) 100% - 100%

(179) 100% - 100%

(180) 100% - 100%

(181) 100% - 100%

(182) 100% - 100%

(183) 100% - 100%

(184) 100% - 100%

(185) 100% - 100%

(186) 100% - 100%

(187) 100% - 100%

(188) 100% - 100%

(189) 100% - 100%

(190) 100% - 100%

(191) 100% - 100%

(192) 100% - 100%

(193) 100% - 100%

(194) 100% - 100%

(195) 100% - 100%

(196) 100% - 100%

(197) 100% - 100%

(198) 100% - 100%

(199) 100% - 100%

(200) 100% - 100%

(201) 100% - 100%

(202) 100% - 100%

(203) 100% - 100%

(204) 100% - 100%

(205) 100% - 100%

(206) 100% - 100%

(207) 100% - 100%

(208) 100% - 100%

(209) 100% - 100%

(210) 100% - 100%

(211) 100% - 100%

(212) 100% - 100%

(213) 100% - 100%

(214) 100% - 100%

(215) 100% - 100%

(216) 100% - 100%

(217) 100% - 100%

(218) 100% - 100%

(219) 100% - 100%

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 · minsize ·

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 & minsize again place the button obj calling the next function

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

C. H. MANN'S  
CLOTHING

A whiteboard diagram illustrating a window frame. The frame is represented by a rectangle divided into four quadrants. The top-left quadrant contains the word "window". The top-right quadrant contains the word "next window". The bottom-left quadrant contains the letter "X" above a horizontal line with a square symbol. The bottom-right quadrant contains the letter "T".

~~Ques: What is the difference between a primary and a secondary market?~~

John Chapman  
1870

Die Vierjahresperiode ist durch die vier Jahreszeiten bestimmt.

```

from tkinter import *
root = Tk()
root.title("python")
root.configure(bg = "black")
root.minsize(200,200)
root.maxsize(200,200)
leftframe = Frame(root, width=100, height=200, bg="white")
leftframe.grid(row=0, col=0, padx=10, pady=10)
label (leftframe, text = "hello", relief=RAISED).grid(row=0, column=0, padx=10, pady=10)
label (rightframe, text = "hey", relief=SUNKEN)
image = PhotoImage(file = "earth.gif")
on_image = image.subsample(3,3)
label (leftframe, img = on_image).grid(row=0, column=0, padx=10, pady=10)
label (rightframe, img = on_image).grid(row=0, column=1, padx=10, pady=10)
def abc():
    print ("new title")
toolbox = Frame(leftframe, width=100, height=100),
        .grid (row=2, column=0, padx=10, pady=10)

```

Shot on OnePlus  
By Ashay

- 55
- write a program to insert an image in the frame widget using the canvas widget.
  - Create the parent window object & use the method title.config and the window with this object.
  - Create an object from the frame method & it to the parent window object with the width, height and bg color specifies & use the grid() along with the row & column attribute as (0,0) with some external padding.
  - Similarly create the right frame object from the frame() with scope & col attribute making the values (0,1).
  - use the label() to the parent window object corresponding to left frame reset & relief attribute specific & col value (0,0) in the grid().
  - Similarly create the label for right frame & use the title & the row col value (0,1).
  - use the photo() with the file attribute specify & subsequently subsample() for specifying the image obj.

48  
7) Now use the label() using the left attribute and the image attribute of the robot obj value specified in the grid.  
Similarly create the long label() using right frame obj with the long attribute the right frame obj with the long attribute some by color with value (0,0).

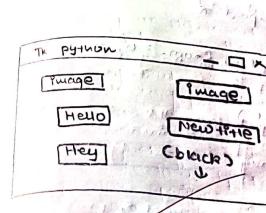
8) Now define a fn using the print statement which shall be called on defining the screen.

9) Create an object from frame() named tk and place it on the leftframe with value as (5,0).

10) Now create the button obj & place it on the toolbox object with the text & command attribute specify.

11) Now create the button obj & place it on the toolbox object with the text & command attribute specify.

12) Now create the button obj & place it on the toolbox object with the text & command attribute specify.



Jai

100

100

100

100

100

100

100

100

100

100

Shot on OnePlus  
By Arijay

spinbox :-

constructor

use the tkinter library to import the relevant method int() for the input from user.

create into the parent window object

but creation from

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 is created by option command~~

## Practical - 5E

### \* Paned Window

- 1) Create an object from the paned window and use the pack() to make this obj visible.
- 2) Now create an object from the entry widget & place it onto the paned window and use the add method. Similarly create an object of a paned window.
- 3) Create an object from the label widget & place it onto the preceding paned windows & use 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 Ashay

### SOURCE CODE:

```
from tkinter import  
w1 = PanedWindow()  
w1.pack(fill=BOTH, expand=1)  
e = Entry(w1, text="left pane", bd=10)  
m2 = PanedWindow(w1, orient=VERTICAL)  
w1.add(m2)  
top = Scale(m2, orient=HORIZONTAL)  
w2.add(top)  
bottom = Button(w2, text="bottom pane", command=top)  
w2.add(bottom)  
w2.pack()  
w1.mainloop()
```

### OUTPUT:

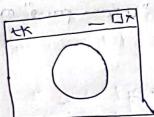


### # Source code

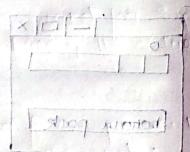
#### 8.8 FOR OVAL

```
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



BEGINNING  
Canvas widget

59

#### # Canvas Widget

- 1) Create an object from the canvas widget by using the attribute master, width, height, bg color & the parent window object.
- 2) Use the corresponding method for drawing the simple geometrical shapes like arc, oval, line and specify the co-ordinate values.
- 3) Similarly use the create\_oval() to create oval along with the coordinate values of the fill attribute for specifying the color.
- 4) Finally use the pack & mainloop().

• Like other regular window problems, starting with mainloop().  
• After our pack() of canvas frame.

• Then pack() underneath the main window frame (a).  
• Now another window part in "b".

E.P.

## PRACTICAL - 6

- Database Connectivity.
- o Imported the relevant libraries for the database connection and the import `sqlite3` function library.
  - o Now create an object for making the connection for the given database.
  - o Further create an object corresponding to the cursor object for execution of the different query statements.
  - o Use the cursor object so created for implemented the structure of the database & the values with the database.
  - o Use the execute method for implementation to select clause for fitting the info.
  - o Now use the fetch all method along with cursor object for display the values on the screen.

Shot on OnePlus  
By Ashay

### Source Code

```
import os,sqlite3  
conn = sqlite3.connect("student.db")  
cur = conn.cursor()  
cursor.execute("create table info (Roll int, Name  
text) sqlite3.cursor object at 0x02F62F0>  
cursor.execute("insert into info values (01, "Ashay", "0-08"  
( 02, "Om", 05-06-1999"))  
<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" )