

PRACTICAL - 2

AIM:- Demonstrate the use of different file accessing modes, different attribute read methods.

step 1: Create a file object using open method and use the write access mode followed by writing some contents onto the file and then closing the file.

step 2: Now open the file in read mode and then use read(), readline() and readlines() and store the output in variable and finally display the content of variable.

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

Step 4: Now open the fileobj in read mode and write same content again subsequently. Their again open the fileobj in read mode that is the update mode and write content.

Step 5: Open file obj in read mode and display the update written content and close open again in + mode with parameter passed and display the output subsequently.

Step 6: Now open file obj in append mode open with methods write contents again open the fileobj in read mode and display the appending outputs.

Output:- (Source code) :-

```
# Demonstrate Writing Write () :-  
fileobj = open ("xyz.txt", "w")  
fileobj.write ("Hello world is a basic sentence  
used in programming language.\n")  
fileobj.write ("It is used to check the print  
output is correctly working or not.")  
  
file.close()  
  
# Demonstrating READ () :-  
# Demonstrating READ () :-  
file obj = open ("xyz.txt", "r")  
line 1 = file obj.read()  
fileobj.close()  
  
# Demonstrating APPEND () :-  
file obj = open ("xyz.txt", "a")  
file obj.write ("I am student of Shaktee College  
of Science and Commerce.\n")  
fileobj.close()  
  
# Demonstrating Read + (j) :-  
file obj = open ("xyz.txt", "r+")  
fileobj.write ("I am a college star checker\n")  
line 2 = file obj.read()  
print ("file contents are: " + "\n", line, line2)  
  
# Demonstrating Tell () :-  
position = fileobj.tell()  
print ("file current position is: ", position)  
  
# Demonstrating Seek () :-  
file obj.seek (40, 0)  
pos, position 2 = fileobj.tell()  
line 3 = fileobj.read ()  
print ("the current position is: ", position)  
print ("the output is: ", line 3)
```

Output:-

file content are:

Hello world is a basic sentence used in programming language.
it is used to check the print output is correctly working or not.

will be used as a basic sentence used in programming language.
it is used to check the print output is correctly working or not.
i am a student of shakun college of science and commerce.

The current year is 187
The user's position is 40
OUTPUT is :- programming language.
it is used to check the print output is correctly working or not.
i am a student of shakun college of science and commerce.

No.

8.

9.

10.

11.

12.

Step 7: Open the fileobj is made
declare a variable and perform file object
set tell method and store the output
consequently it is available.

Step 8: Use the seek method with the argument
with opening the file obj is made
read and closing subsequently.

Step 9: open file obj with read mode also use
the readlines method and store the
output consequently in and print the
same file, counting the length use
the len, conditions statement and
display the length.

~~for i in range(0, len)~~

OUTPUT:-

SOURCE CODE:-

class odd

```
def __iter__(self):
    self.num = 1
    return self
def __next__(self):
    if self.num <= 0:
        raise StopIteration
    num = self.num
    self.num += 2
    return num
else:
```

OUTPUT:-

1
3
5
7
9

11
13
15
17
19

def __iter__(self):

self.num = 1

return self

def __next__(self):

if self.num <= 0:

raise StopIteration

num = self.num

self.num += 2

return num

else:

raise StopIteration

y = iter(odd(5))

y = next()

2) Odd Numbers :-

25

Step 1 :- Define a iter method with an argument and initializes the value and return the value.

Step 2 :- Define the next method with an argument and compare the upper limit by using a conditional statement to increment the value by 2.

Step 3 :- Now create an object of the given class and pass this object in the iter method.

Power

Step 1 :- Define iter method with 3 Argument
1. Initialize the "first argument" as 1.
Initialize the other two arguments
as "Enter the number" and
"Maximum limit of power"
respectively.

Step 2 :- Define the next method with an argument and compare it by using a conditional statement
Increment the value by 1.

Step 3:- Now create an object of the given class and pass this object as the item needed to use the entire condition inside Statement no. period.

SOURCE CODE :-

class registrars

Step 1: Define the method with 3 arguments
Initialize the first argument as 1
Inside the other two arguments
as "Enter the number" and
"Maximum limit of power"
respectively

Step 2 :- Define the next method with an argument and compare it by using a conditional statement to increment the value by 1.

Step 3:- Now create an object of the given class and pass this object as the item received & use the entire conditional statement to print

EIGHTY-FIFTH

less replete :
def-ite - (prou) :

```
paw.n1 = int(input("Enter the number:"))
paw.n2 = int(input("Masukkan bilangan ke-2"))
jumlah_paw
```

```

-- next_ - (paw) :
if paw . n < paw . n2 :
    num = paw . n1 + paw . n
    paw . n + = 1
    return num
else :
    print Step Iteration
    tree[Confiter(1)]
    print (next(1))

```

INPUT

→ Enter the number = 2
→ Maximum limit of power = 4

24816

#PROGRAM :-

SOURCE CODE :-

While True:

 try:

 a = int(input("Enter a number : "))

 print("Valid number")

 break

except ValueError:

 print("Not a valid number! Try again")

10. OUTPUT :-

>> Enter a number :

Not a valid number!

Enter a number ! Try again

Valid number.

PRACTICAL No. 3

27

Aim:- To demonstrate exception handling

Ques:- Write a program using the exception method of the nature arithmetic 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 terminate 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.

PROGRAM

SOURCE CODE:-

28

```
try:
    fo = fileobject = open("xyz.txt", "w")
    fileobject.write("Hello world")
except IOError:
    print("Error writing on the file")
else:
    print("operations carried out successfully")
    fileobject.close()

Step 1:- Within the try block open, write into the file using the mode "w" and meanwhile read and write content on the file.

Step 2:- Use the except block with IOError and displays the message regarding missing of the file. In case of the mode "w" it overwrites the file otherwise it displays a message that the operation is carried out successfully.
```

No

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

PROGRAM

SOURCE CODE:-

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

OUTPUT :-

list is empty.

2.9

- 3) Write a program using the assert() do check if the list element are empty.
- Step 1 :- Define a function which accepts an argument and check using the assert statement whether the given list is empty list and accordingly return the message.

Step 2 :- Use the function and in the body of program and define certain elements in list and take some appropriate action.

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

Step 1

P.T.O

PROGRAM

SOURCE CODE:-

```
30  
def acceptAge():  
    age = int(input("Enter age : "))  
    if (age > 30 or age < 16):  
        raise ValueError  
    return age  
  
valid = False  
while not valid:  
    try:  
        age = acceptAge()  
        valid = True  
    except ValueError:  
        print("Not a valid age")
```

OUTPUT :-
Enter age : 6
Not a valid age

Enter age : 28

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

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

Step 3:- Define the while loop to check whether the below expression holds true use this try block to accept the age of student and terminate the looping condition.

Step 4:- Use except with value Error and print the message not or valid range.

Code

import re

```
string = "hello 123 4x24 456"
result = re.findall("1\d", string)
print(result) # ['123']
result = re.findall("11", string)
print(result) # ['11']
```

OUTPUT

```
>>> [ '1234', '4567' ]
>>> [ 'hello', 'xyz' ]
```

PRACTICAL - 04

31

Aim: Demonstrate the use of regular expression

THEORY: Regular expression represents the sequence of characters which is mainly used for finding & replacing the given pattern in a string and for this we import module and various usage of regular expression provided following functions.

- Searching a given string
- Finding a string
- Finding a string into smaller substring
- Replacing part of string.

1) Write a regular expression resequating numeric & alphabetic values from a given string Algo.

Step 1: Now apply & string & patterns in find all & and display the output.

Step 2: \d is used for matching all decimal digit where as \w is used to match non decimal digits.

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

Algorithm:

Step 1:- Import module to apply a string

Step 2:- We wants cuts "A Python" and string as two parameters.

Step 3:- Now display the output

Step 4:- Now we if condition statement
for were we know whether match is found or not:

code:

import re

string = "Python is an intended language"

result = re.match ("A Python", string)

print (result)

if result

print ("Match Found")

else:

print ("Match not found")

16

17

18

19

20

21

22

23

24

25

match = "Python"

>>> result

code:

Programme

```
l1 = ["8828207485", "9892009000", "9892040208"]
```

For element in l1:

```
result = re.match("[8-9]-[0-9]{3}-[0-9]{4}")
```

if result:

```
    print("correct mobile no")
```

```
    print(result.group(1))
```

else:

```
    print("In correct mobile no.")
```

OUTPUT:

```
>>> correct mobile no.
```

```
8828207485
```

```
>>> correct mobile no.
```

```
9892009000
```

#

```
>>> incorrect mobile no.
```

3) Write a regular expression to check whether the given mobile numbers start with 8 or 9 of the total length 10.

Algorithm:

Step 1: Import we module & apply a leting of mobile no. &

Step 2: Now we have conditional statement to find wether the number start with 8 or 9 & the total number should be of length 10. We matches 15 prede you statement to find the matches in given leting.

Step 3: We if conditional statement to know whether we have to match on root if we have we group(1) too. display the output & if we didn't display in correct mobile no.

- 4) 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 & apply a string
- Step 2: we findall() to extract a word from given string.

Step 3: We "\w*" to extract word along with space & we "\w+" i.e extract word without space.

Step 4: Now display the output

```
# code
import re
string = "Python is an intended language"
result1 = re.findall("\w+", string)
result2 = re.findall("\w+", string)
print(result1)
print(result2)
```

```

# code:
Impact all
string = "Python is a programming
language"
result = re.findall(r"\w+", string)
result2 = re.findall(r"\w+\$+", string)
print(result)
print(result2)

# OUTPUT
>>> Python
>>> ['Python']

>>> ['language']

```

- 5) Write a regular expression for extracting
first & last word from a string
- Algorithm:-

Step 1:- Import re module & apply a string.

Step 2:- Use.findall() in which we "\w+" as
one parameter i.e find first word
of string there are "\w+\\$+" all
parameters to find last word
of string

Step 3:- Display the output

- Q) Write a regular expression for extracting the date in format `dd-mm-yyyy` by using the `findall` where the string has fallowing format.
- `AUTHOSH 12-12-2019`
- Algorithm:
Step 1:
- Step 1:- Import all module & apply a string

No.	1.
2.	
3.	
4.	
5.	
6.	
7.	
8.	
9.	
10.	
11.	
12.	

```
# Code:
import re
string = "AUTHOSH 12-12-2019"
result = re.findall(r'\d{2}\-\d{2}\-\d{4}", "string'
print(result)

# OUTPUT
>>> ['12-12-2019']
```

Step 2:- We findall method & use `\d{2}-\d{2}-\d{4}` as a parameter.

Step 3:- Now display the output

code

import re

string = "abc. @tesc. edu"

result1 = re.findall ("[^@]+", string)

result2 = re.findall ("[^.]+@[^.]+", string)

print(result1)

print(result2)

print(result3)

```
# output
>>> ['abc']
>>> ['abc' '@tesc. edu']
>>> ['abc' '@tesc. edu']
```

- Want a user from extracting the
 ① Username from email.
 ② Hostname from email.
 ③ Both username & hostname from email id.
- Algorithm:
- Step1: Import the module & apply string
- Step2: Use findall [] to find username, host
 name both from email id.
- Step3: Use "\not" for username use "+\w" + "\w"
 w]. co + \$ " for host name & use "[\w]"
 + " for both as parameter for findall
- Step4: Display the output.

Practical No - 5

Ans:- GUI components

Step 1:- Use tkinter library for implementing the features of the text widget.

Step 2:- Create an object using the Tk()

Step 3:- Create a variable using the widget label : and via the text method.

Step 4:- Use the mainloop() for triggering of the corresponding about mention words.

Step 2:- Use the tkinter library for import the feature of the text widget parent window.

Step 3:- Create a variable from the text method and position it on the parent window.

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

- 1) side = LEFT , padx = 20
- 2) side = LEFT , padx = 20
- 3) side = TOP , padx = 30
- 4) side = TOP , padx = 40
- 5) side = TOP , ipadx = 50

creation of parent window

from tkinter import *

```
root = Tk()
l = Label (root, text = "Python")
l . pack()
root . mainloop()
```

OUTPUT:-



#2

from tkinter import *

root = Tk()

```
l = Label (root, text = "Python")
l . pack ()
```

l1 . pack ()

```
l2 = Label (root, text = "CS", bg = "yellow",
fg = "black", font = "10")
```

l1 . pack (side = LEFT, padx = 20)

```
l2 = Label (root, text = "CS", bg = "Blue",
fg = "black", font = "20")
```

l2 . pack (side = LEFT, padx = "30")

```
l3 = Label (root, text = "CS", bg = "grey",
fg = "black", font = "10")
```

l3 . pack (side = TOP, ipadx = 20)

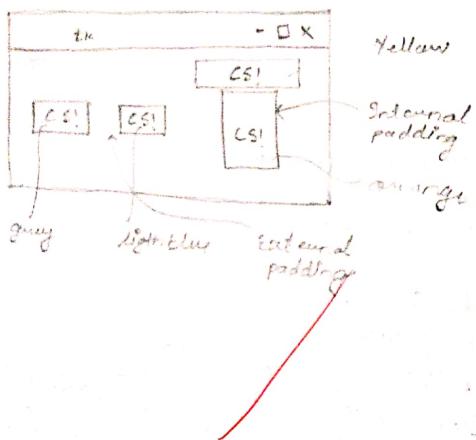
88

```

l4 = Label(window, text = "CS!", bg = "orange",
           fg = "black", font = "10")
l4.pack(side = TOP, ipady = 50)
window.mainloop()

```

ii) OUTPUT:-



39

Step 4:- Use the mainloop() function for triggering of the corresponding events.

Step 5:- Now repeat above steps with the label 'l3' which takes the following arguments

- 1) Name of the parent window
- 2) Text attribute which defines the string
- 3) The background color (bg)
- 4) The foreground fg and there use the pack() with a relevant padding attribute.

Jain

#3 Simple Tkinter Example

40

root = Tk()

root.geometry ("500x500")

~~Step 1:- Import the relevant methods from the tkinter library create an object with the parent window.~~

~~Step 2:- Use the parent window object along with the geometry () defining specific pixel size of the parent window.~~

~~Step 3:- Now define a function, which tell the user about the given selection made from multiple option avail-~~

~~Step 4:- Now define the parentwindow and define the option with control var.~~

~~Step 5:- Use the listbox() and insert options on the parent window along with the pack() with specifying anchor attribute.~~

~~Step 6:- Create an object from macDoos button which will take following arguments & parent window object, it set variable which will take the values option one, two, three - variable argument, values & trigger the function defined.~~

~~var = StringVar()
l1 = Listbox()
l2 . present (l1, "List 1")
l1 . pack (anchor = N)
r1 = Radiobutton (root, text = "option 1", variable
= var, value = "option 1", command = select)
r1 . pack (anchor = N)
r2 = Radiobutton (root, text = "option 2", variable
= var, value = "option 2", command = select)
r2 . pack (anchor = N)~~

Dm

OUTPUT:

tk

List 1
List 2

41

Step 3:- Now call the pack() function from radius object using anchor & side.

#Listbox

Step 8:- Finally make use of the mainloop()

○ option 1
○ option 2

You just selected
option 1

#Rackig
Buttons

Step 1:- Import relevant methods from the tkinter library

Step 2:- Create a parent object corresponding to the parentwindow

Step 3:- Use the geometry() function of the window

#2
scrollbar() =
scroll + button import *
root = Tk()

root.geometry("500x500")
s = scrollbar()
s.pack(side = "right", fill = "y")
root.mainloop()

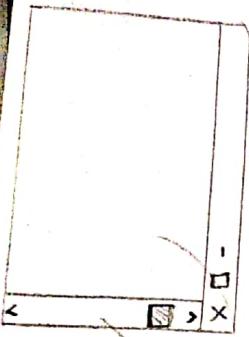
Step 4:- Create an object and use the scrollbar

Step 5:- Use the pack() along with the scrollbar object with side and fill attributes

Step 6:- Use the root.mainloop with the parent object

geometry()

scrollbar()



#5

from Tkinter import *

window = Tk()

window.geometry ("650x500")

label (window, text = "numbers :").pack()

frame = Frame (window)

frame.pack()

listNodes = Listbox (frame, width = 20, height = 20,

font = ("Times New Roman", 10)),

listNodes.pack (side = "left", fill = "y")

scrollbar = Scrollbar (frame, orient = "vertical")

scrollbar.config (command = listNodes .view)

scrollbar.pack (side = "right", fill = "y")

for i in range (100):

listNodes.insert (END, str (*))

window. mainloop ()

output:



numbers :

list

-

x

listbox

Step 1:- Import the relevant libraries.
the tkinter module.

Step 2:- Create an corresponding object of the parent window.

Step 3:- Use the geometry manager with pin size (650x500) or any other suitable pixel value.

Step 4:- Use the frame widget along with the parent object created and use the pack method.

Step 5:- Use the listbox method along with the attrbute, then use pack, for the same.

Step 6:- Use the scrollbail attrbute an object use the attrbute of vertical, then configue the same.

Step 7:- Trigger the event using mainloop

42

6 frame window import

windows = Tk()

windows = geometry ("680 x 500")

frame = frame(windows)

frame.pack()

leftframe = frame(windows)

leftframe.pack(side: "LEFT")

rightframe = frame(windows)

rightframe.pack(side: "RIGHT")

b1 = Button(frame, text: "Select", activebackground

= "red", fg = "blue")

b2 = Button(frame, text: "Modify", activebackground

= "yellow", fg = "black")

b3 = Button(frame, text: "Add", activebackground

= "Blue", fg = "red")

b1.pack(side: "LEFT", padx: 20)

b2.pack(side: "RIGHT", padx: 30)

b3.pack(side: "bottom", pady: 20)

#6 step1:- Import relevant methods from tkinter library.

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

Step 3:- Now define the frame object from the method and place it on the parent windows.

Step 4:- Similarly define the RIGHT frame and subsequently define the bottom object placed onto the given frame with the attr but as text, active background and framename.

Step 5:- Now use the pack along with the side attribute.

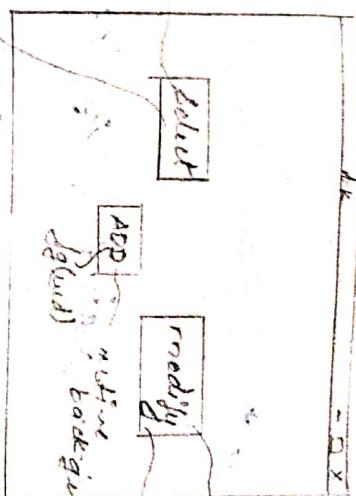
Step 6:- Similarly create the bottom object corresponding to MODIFY operation Put it into frame object on side = "right".

8.

Step 8:- Create another button object & place it on the RIGHT.

Step 9:- Add another button & puts it on the top of square and label it as EXIT.

Step 10:- Use this pack() function all the objects & finally use the mainloop()



No.

8.

Source Code

from chapter important *

~~text = Tk()~~

b1 = Button(text="RAISED", relief="RAISED")

b1.pack()

b2 = Button(text="GROOVE", relief="GROOVE")

b2.pack()

b3 = Button(text="RIDGE", relief="RIDGE")

b3.pack()

b4 = Button(text="FLAT", relief="FLAT")

b4.pack()

b5 = Button(text="SUNKEN", relief="SUNKEN")

b5.pack()

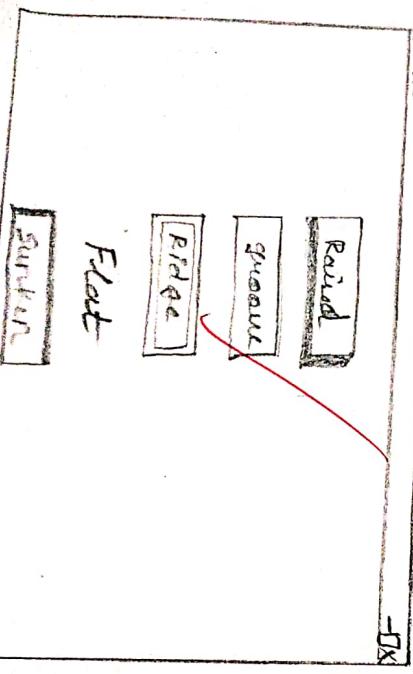
Output:-

Step 1:-

Step 2:- Define a button object and place it on to the object corresponding to the parent window.

Step 3:- Use the text attribute for specifying the title to the button object.

Step 4:- Use the positioning the widget object on to the parent window and trigger the corresponding event by calling the mainloop().



Writ a program on various attributes which a button widgets may assume related to the relief attribute.

Step 1:- Define a button object and place it on to the object corresponding to the parent window.

Step 2:- Use the text attribute for specifying the title to the button object.

Step 3:- Use the relief attribute with one style at a time with different button object.

Step 4:- Use positioning the widget object on to the parent window and trigger the corresponding event by calling the mainloop().

#8

7B

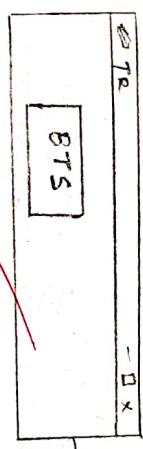
SOURCE CODE:-

shawinfo()
 step 1:- Define a function which will use the
 shawinfo() derived from the
 message box library.

step 2:- The attribute which is given method
 takes with specify the 2 strings or
 values so the letter coming from
 function may displayed

step 3:- Now create an object from the
 button method and place it on to
 the parent window with the title
 of the button object and finally
 use the command attribute

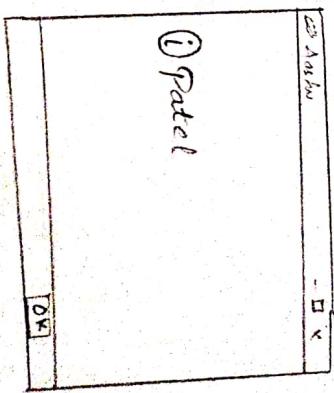
OUTPUT:-



```

b1 = Button(top, text="BTs", command=msg)
b1.pack()
top.mainloop()
  
```

step 4:- Turn into the programming by calling
 the mainloop().



46

from tkinter import *

top = Tk()

messagebox

def msgbox():

messagebox. showwarning ("shujin", "Python")

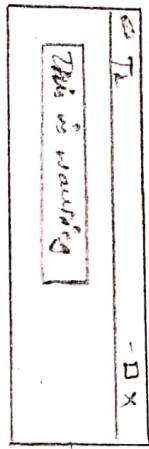
b1 = Button (top, text = "This is a warning")

command = "msgbox")

b1.pack()

top.mainloop()

OUTPUT:-



Show warning()

Algorithm:-

Step 1:- Define a function which will use the

showwarning() method from the
messagebox library.

Step 2:- The attributes which a given method

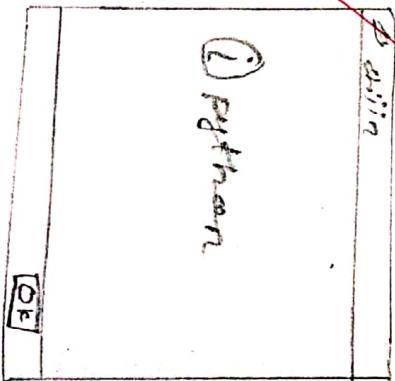
take will specify the 2 steps on
related to the message display &
corresponding to the message.

Step 3:- Now create an object from the method
and place it on the parent window

with the title of the button object
specified and finally use the command
attribute to control the relevant part.

Steps:- To illustrate the program by calling the
mainloop()

① Python



Show Examples

Algorithm :-

Step 1: Define a function which will use the `showmessage()` function from the message box library.

Step 2: In the object which a given as takes in specify the 2 strings

- related to little
- corresponding to the message.

Step 3: Now create an object from the button method and place it on the steps parent window.

Step 4: Terminate the program by using `mainloop()`.

function `tkinter`, import
`keep = Tk()`

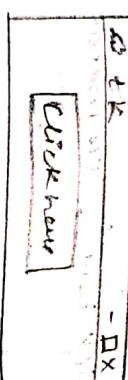
`msg = messagebox
 def msgbox:`

messagebox.

`button (text = "click here", command = msg)`

`def mainloop ()`

`output`



CODE :-
 from tkinter import*
 top = Tk()

No.
 8.
 messagebox
 msgbox

def msg_box

messagebox.askyesno("Python", "Do you want
 to exit")

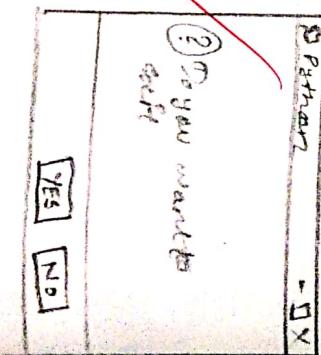
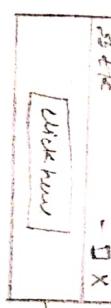
b) = Button (top, text="click here", command=

msg_box)

b). pack()

top.mainloop()

OUTPUT:-



Algorithm:-

Step 1:- Define a function which will use the askyesno() defined from the messagebox library.

Step 2:- The attri bute which is given represents state of the window.
 1) related to the title of the window.
 2) corresponding to the message displayed

Step 3:- Now create an obj. of class the button method and place it on the parent window with title of the buttons as specified and finally use the command attri bute to specify functions.

Step 4:- Terminate the program by calling the mainloop.



all ok now()

Algorithm:-

Step 1:- Define a function which will take askcancel from the messagebox library.

Step 2:- The other part which will be asked takes only specify the string related to the title.
2) Corresponding to the message display by result.

Step 3:- Now create an object from the button method and place it on the parent window.

Step 4:- Turn into the program by calling mainloop().

CODE:-

step 1:

from tkinter import *

top = Tk()

messagebox

by result

messagebox.askcancel("Python", "Continue the game")

b1 = Button(top, text="asking", command=msgbox)

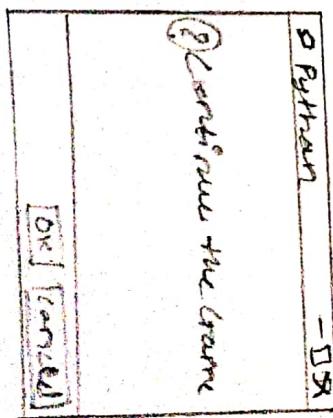
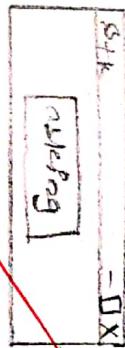
b1.pack()

step 2:

step 3:

step 4:

OUTPUT:-



Ques:-

Practical 5 →

ask question C

frame printer import

tcp = Tk()

messagebox

def msgbox:

messagebox . askquestion ("Python", "Codium")

b1 = Button (tcp, text = "click here", command=msgbox)

tcp . pack ()

tcp . mainloop ()

Output:-

<input type="checkbox"/>	-	<input checked="" type="checkbox"/>
click here		

<input type="checkbox"/>	-	<input checked="" type="checkbox"/>
click here		

(?) Continue

<input checked="" type="checkbox"/>	YES
<input type="checkbox"/>	NO

Step 3:- Now create our object from the button method and place it onto parent window.

(?) Continue

Step 4:- Terminate the program by calling mainloop()

→

CODE:-

from tkinter import *
root = Tk()

def main():

root = Tk()

root.config(bg="blue")

root.title("Main")

root.minsize(200, 200)

b = Label(root, text="Sounds")

b.pack()

l1 = Label(root, text="Soundwaves give relaxation")

l1.pack()

B1 = Button(root, text="Next", command=sec)

B1.pack(side=RIGHT)

B2 = Button(root, text="Temporary", command=rex)

B2.pack(side=BOTTOM)

root.mainloop()

def sec():

master = Tk()

master.config(bg="yellow")

master.title("Page 2.0")

master.minsize(400, 300)

l2 = Label(master, text="workout")

l2.pack()

l3 = Label(master, text="workout gives energy to life")

l3.pack()

B3 = Button(master, text="Previous", command=prev)

B3.pack(side=LEFT)

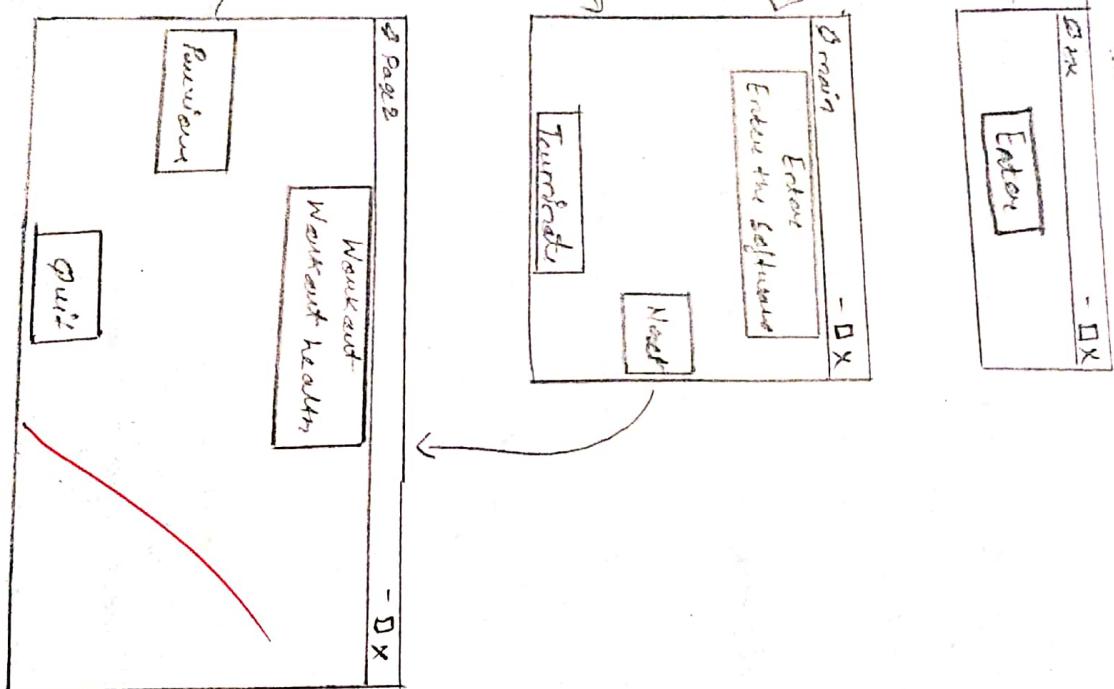
def prev():

root.destroy()

B4 = Button(root, text="End", command=root.destroy)

Instrument Image:-

Step 2:- Create the parent window object and use the method title , config and minsize with this object



Step 2:- Create an object from the frame method and place it into the parent window object with width , height and by column and use the grid method with row and column attributes as $(0,0)$ with same external padding.

Step 3:- Similarly create the rightframe object from the frame method with row and column attributes making the value $(0,1)$

Step 4:- Use the label() as the parent window object comes under the left frame with text and width attribute and use the grid method with row and column values as $(0,0)$:

CODE :-

54

54

Java Swing Impact

Step 5:- Similarly create the label for the right frame and use the title & menu, column value as (0,1) now,

now,

```
frame = Frame("Impact")  
frame.setLayout(new GridLayout(1,1))  
frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE)  
frame.setSize(300, 200)  
frame.setVisible(true)
```

Step 6:- Use the photo() with the file attribute specified and subsequently subsample() for specifying the image object.

Step 7:- Now use the label() using the left frame and the image attribute i.e.

Step 8:- Similarly create the label is using rightframe obj with the photo() attribute of now, column = (0,0)

```
label1 = new Label("Impact", "Impact", 100, 100)  
label1.setBounds(10, 10, 100, 100)  
label1.setLayout(new GridLayout(1,1))  
label1.add(photo1 = photo("Impact.gif"))  
label1.setVisible(true)  
  
label2 = new Label("Impact", "Impact", 100, 100)  
label2.setBounds(10, 10, 100, 100)  
label2.setLayout(new GridLayout(1,1))  
label2.add(photo2 = photo("Impact.gif"))  
label2.setVisible(true)
```

Step 9:- Now use the mainloop method to terminates the given program.

main().mainloop()

Output:-

18

Spinbox

Algorithm

- IX

Step 1:- Use the `steiner` library to import the `Steiner` method

`Image1`

`Image2`

Step 2:- Create the parent window object

Step 3:- Create an object from a spinbox method & place it on to the parent window with the options specified

Step 4:- Now, use the `pack` method to make the object visible onto the parentwindow.

Code:-

`from tkinter import*`

`top = Tk()`

~~`S = Spinbox (top, from_ = 0, to = 10)`~~

`S = pack()`

`top.mainloop()`

18	- IX
10	10