

150,

Practical - 1

Objective :- Demonstrate the use of different file accessing modes, different attributes 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 and then use read(), readline() and readlines(); and store the output in variable and finally display the contents of variable

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

```
file obj = open ("abc.txt", "w") # write mode
file obj.write ("computer science subjects" + "022n")
file obj.write ("DBMS\n PYTHON\n DS\n")
file obj.close () # file close

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

>>> ('The output of read method:', 'Computer science
    Subjects\n DBMS\n PYTHON\n DS\n')

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

>>> ('The output of readline method:', 'computer science
    subjects\n')

# readlines()
file obj = open ("abc.txt", "r")
str3 = file obj.readlines()
print ("The output of readlines method:", str3)
file obj.close()

>>> ('The output of readlines method:', ['computer science
    subjects\n', 'DBMS\n', 'PYTHON\n', 'DS\n'])

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

```

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

# write mode  

c=fileobj.mode  

print("file mode",c)  

>>>('file mode','r')  

d=fileobj.read()  

print("software",d)  

>>>('software',0)  

# w+ mode  

fileobj=open("abc.txt","wt")  

fileobj.write("loukik sir")  

fileobj.close()  

# r+ mode  

fileobj=open("abc.txt","rt")  

s1=fileobj.read()  

print("output of rt",s1)  

fileobj.close()  

>>>('output of rt','loukik')  

# append mode  

fileobj=open("abc.txt","a")  

fileobj.write("data structure")  

fileobj.close()  

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

s3=fileobj.read()  

print("output of append mode:",s3)  

fileobj.close()  

>>>('Output of append mode:','loukik sir data
structure')

```

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

Step 5:- Open file obj in read mode display the update written contents and close open again in 'rt' mode with parameter passed and display the output subsequently.

Step 6:- Now open file obj in append mode open write method write contents close the file obj again open the file obj in read mode and display the appending output

BSO:

Step 7: Open the file obj in read mode, declare a variable and perform file obj dot tell method and store the output consequently in variable.

Step 8: Use the seek method with the arguments with opening the file obj in read mode and closing subsequently.

Step 9: Open file obj with read mode also use the readlines method and store the output consequently in and print the same for.

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

#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")
st1 = fileobj.seek(0,1)
print ("seek (0,1) is : ", st1)
fileobj.close()
>>>('seek (0,1) is : ', None)

fileobj = open("abc.txt", "r")
st2 = fileobj.seek(0,2)
print ("seek (0,2) is : ", st2)
fileobj.close()
>>>('seek (0,2) is : ', None)

#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))
fileobj.close()
>>>('output : ', ['loukik sir data structure'])
```

lot
over

PSQ.

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

y=iter(odd())

while True:
 print(next(y))

Output :-

1
3
5
7
9
11
13
15
17
19

[Errno 2] [Errno 2]

Practical 2

025

Objective :- To demonstrate the use of iterators & iterable.

odd numbers.

Algorithm:-

Step 1 :- Define a iter method with an argument and initialize 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 Increment the value by 2.

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

750

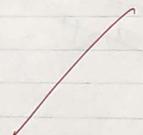
Power

Algorithm

Step 1:- Define iter method with 3 Arguments.
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 in the iter method & use the while conditional statement to print.



026

```
class myiter:  
    def __iter__(self):  
        POW.n=1  
        POW.n1=int(input("Enter the number"))  
        POW.n2=int(input("Enter the Maximum  
        Limit of power"))  
  
    def __next__(self):  
        if POW.n>=POW.n2:  
            raise StopIteration  
        num=POW.n1**POW.n  
        POW.n+=1  
        return num
```

y = iter(myiter())
while True:
 print(next(4))

Output :-

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

2
4
8
16

as0

```

class myrange:
    def __iter__(self):
        self.a = 1
        return self

    def __next__(self):
        if self.a <= 30:
            x = self.a
            self.a += 1
            return x
        else:
            raise StopIteration

myclass = myrange()
myclass.__iter__()

```

```
myclass = myrange()  
mydata = iter(myclass)  
for x in mydata  
    print(x)
```

Output,

1	16
2	17
3	18
4	19
5	20
6	21
7	22
8	23
9	24
10	25
11	26
12	27
13	28
14	29
15	30

030
0829

Range

~~Range~~ Algorithm

Step 1:- Define a iter method with an argument & initialize the value & return that value.

Step 2 - Define the next method within an argument & compare, the upper limit by using a conditional statement.

Step 3:- Now, * create an object of the given class & pass this object in the iter method & use the for statement to print.

Practical No: 3

Aim: Program to demonstrate exception handling.

- 1] 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.

- 2] Write a program for accepting the file in a given mode and use the environment error as an exception for the given input.

Step 1:- Within the try block open the file using the write mode and write mode and write some content on the file.

Step 2:- Use the except block with IO error and 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.

Program :-

```
while True
    try:
        x = int(input("Enter class"))
        break
    except ValueError:
        print("Enter Numeric value")
```

Output :-

```
enter class: 467
```

Program :-

```
try:
    f0 = open("abc.txt", "w")
    f0.write("Tushar Dhule")
except IOError:
    print("Error writing on the file")
else:
    print("Operation carried out successfully")
f0.close()
```

Output:-

```
Operation carried out successfully.
```

3]. Write a program using the assert() to check if list elements are empty.

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

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

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

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

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

Program :-

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

Output :-

list is empty.

Program :-

```
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: 4
Not a valid age
Enter age: 18
```

850

age = int(input("Enter age: "))
if age < 18:
 print("You are not old enough to vote")
else:
 print("You are old enough to vote")
print("Have a nice day!")

032

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

Step 4:- Use except with ValueError and print the message not a valid range.

JMR

Practical No. 4

Aim: Demonstrate the use of regular expression.

Theory: Regular expressions represents the sequence of characters which is mainly used for finding & replacing the given pattern in a string and for this we import re module and common usage of regular expression involves following functionalities.

- * searching a given string
- * finding a given string
- * breaking a string into smaller substring
- * Replacing part of string

Q.1. Write a regular expression segregating numeric and alphabetic values from a given string.

Algorithm:

Step 1: Now apply string & pattern in.findall() and display the output.

Step 2: \d is used for matching all decimal digits whereas . is used to match non-decimal digits.

Code1:-

```
import re
string = "hello 1234 abc4567"
result = re.findall("\d", string)
result = re.findall(".", string)
print (set(result))
print (result+1)
```

Output:-

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

✓

Code 2

```
import re  
string = "Python is an important language"  
result = re.search("\A Python", string)  
print(result)  
if result:  
    print("match found")  
else:  
    print("Match not found")
```

Output

```
>> rematch object span=(0,6)  
      match="Python"  
>> match found
```

033

Q2] Write a regular expression for finding the match string at the begining of given sequence.

Algorithm:

Step 1: Import re module and apply a string.

Step 2: Use search() with "\A Python" and string as two parameters.

Step 3: Now display the output.

Step 4: Now use if conditional statement for user to know whether the match is found or not.

880

Q3) Write a regular expression to check whether the given mobile number starts with 8 or 9 & the total length 10.

Algorithm:

Step 1: Import re module and apply a string of module no. 8.

Step 2: Now use for conditional statement to find if the number starts with 8 or 9 and the total number should length of 10. use match() inside for statement to find the match in given string.

Step 3: Use if conditional statement to know whether we have a match or not if we have use group() to display the output and if we don't display incorrect mobile no.

Code 3:-

```
import re  
li = ["9876543210", "8765432109", "67654321098",  
      "65432109867"]  
for element in li:  
    result = re.match("[8-9]{1}[0-9]{9}", element)  
    if result:  
        print("correct mobile no.")  
        print(result.group(0))  
    else:  
        print("Incorrect mobile no.")
```

Output

```
>>> correct mobile no.  
9876543210  
Correct mobile no.  
8765432109  
Incorrect mobile no.
```

Q80.

code 4

```
import re
string = "Python is Important"
result1=re.findall("\w+",string)
result2=re.findall("\w+\t",string)
print(result1)
print(result2)
```

Output:-

```
>>>["Python", "is", "important"]
['Python', 'is', 'important']
```

035

Q4] Write a regular expression for extracting a word from given string along with space characters in between the word and subsequently extract the word without space character.

Algorithm:-

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

Step 2:- Use findall() re extract a word from given string.

Step 3:- Use "\w+*" to extract word along with space & use "\w+" to extract word without space.

Step 4:- Now display the output.

280

Q5] Write a regular expression for extracting first and last word from a string.

Algorithm:

Step 1: Import re module and apply a string.

Step 2: Use.findall() in which use '^\\w+' as one parameter to find first word of string & then use '\\w+\$' as parameter to find last word of string.

Step 3: Display the output.

036

Code 5

```
import re
string="Python is important"
result=re.findall("^\\w+",string)
result1=re.findall("\\w+$",string)
print(result)
print(result1)
```

Output:-

```
>>>['Python']
>>>['Important']
```

Q80.

Code 6

```
import re  
string = "Amit 201.26-12-2019"  
result = re.findall("\d{2}-\d{2}-\d{4}", string)  
print(result)
```

Output

```
>>> ["26-12-2019"]
```

037

Q6) Write a regular expression for extracting the data in format dd-mm-yyyy by using the findall() where the string has following format AMIT 201 24-12-2019.

Algorithm :-

Step 1: Import re module and apply string.

Step 2: Use findall method and we "\d{2}-\d{2}-\d{4}" as an parameter.

Step 3: Now display the output.

Amit 201 24-12-2019

Method is ok now

Applauded

Method is ok now

- Q7]: Write a re for extracting the following from email-id.
- ① Username from email-id.
 - ② Host Hostname from email-id.
 - ③ Both Username & hostname from email-id.

Algorithm:-

Step 1:- Import re module and apply a string

Step 2:- Use.findall() to find username, hostname
& both as email-id.

Step 3:- Use "\wt+" for username use "\w+\.\w+"
for hostname and use "[\w\.-]+\." for both
as parameter in.findall()

Step 4:- Display the output.

code]

```
import re
string = "abc @ tcs.edu"
result 1 = re.findall ["^ \wt+", string]
result 2 = re.findall ["^ \wt . \w+ $", string]
result 3 = re.findall ["[\w\.-]+\.", string]
print [result 1]
print [result 2]
print [result 3]
```

038

Output:-

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

Jm M

```

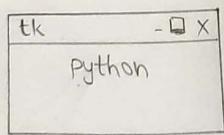
# 1: 880
# Radio Button
from tkinter import *
root = Tk()
root.geometry("500x500")
def select():
    selection = "You just selected " + str(var.get())
    l1.config(text=selection, bg="white",
              fg="green")
l1.pack(side=TOP)
var = StringVar()
l1 = Listbox()
l1.insert(1, "List 1")
l1.insert(2, "List 2")
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)
root.mainloop()

```

```

# creation of parent window
from tkinter import *
root = Tk()
l = Label(root, text="python")
l.pack()
root.mainloop()

```



Practical No. 5

039

Topic :- GUI Components

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

Step 2:- Create an object using the TK().

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

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

2 :-

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

Step 2:- 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.

1> side = LEFT, padx = 20

2> side = LEFT, pady = 30

3> side = TOP, padx = 40

4> side = TOP, pady = 60

E&O.

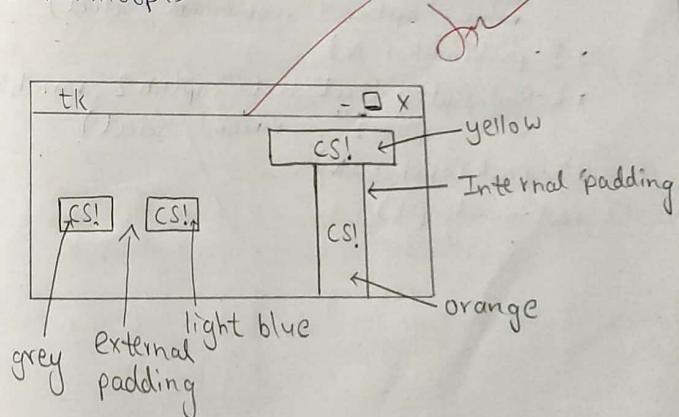
Step 4:- Use the mainloop() for the triggering of the corresponding events.

Step 5:- Now repeat above steps with the label() 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 background fg and then use the pack() with a relevant padding attributes.

2
from tkinter import *
root = Tk()
l = Label(root, text="python")
l.pack()
l1 = Label(root, text="CS!", bg="grey",
fg="black", font="10")
l1.pack(side=LEFT, padx=20)
l2 = Label(root, text="CS!", bg="light blue",
fg="black", font="20")
l2.pack(side=LEFT, pady=30)
l3 = Label(root, text="CS!", bg="yellow",
fg="black", font="10")
l3.pack(side=TOP, ipadx=40)
l4 = Label(root, text="CS!", bg="orange",
fg="black", font="10")
l4.pack(side=TOP, ipady=50)
root.mainloop()

040



```

# 1:
from tkinter import *
root = Tk()
root.geometry("500x500")
def select():
    selection="You just selected "+str(var.get())
    L1=Label(text=selection, bg="white",
              fg="green")
    L1.pack(side=TOP)
var = StringVar()
L1=Listbox()
L1.insert(1,"List 1")
L1.insert(2,"List 2")
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)
root.mainloop()

```

Practical No: 5 (B)

041

Aim: GUI components.

1 :-

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()` declaring specific pixel size of the parent window.

Step 3: Now define a function which tells the user about the given selection made from multiple option available.

Step 4: Now define the parent window and define the option with control variable.

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 radio button which will take following arguments parent window object text variable which will take the values option no 1, 2, 3 ... variable argument corresponding value & trigger the function declared.

140

Step 7: Now call the pack() for radio object so created and specify the argument using anchor attribute.

Step 8: Finally make use of the mainloop() along with parent object.

#2:-
Step 1: Import relevant methods from the tkinter library:

Step 2: Create a parent object corresponding to the parent window.

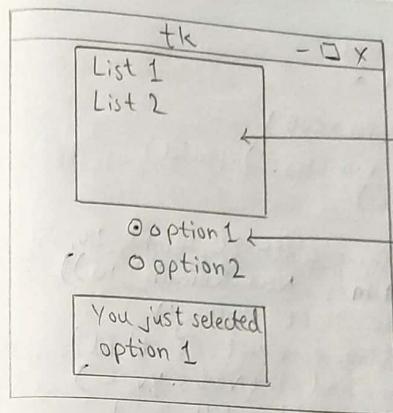
Step 3: Use the geometry() for laying of the window.

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 mainloop() with the parent object

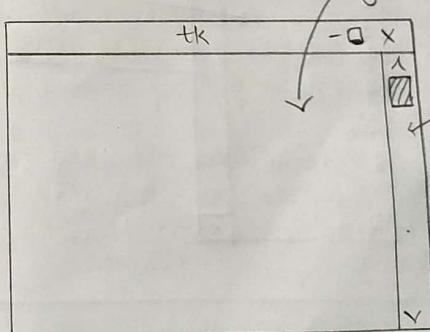
042



Listbox()
Radio Button

#2:-

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



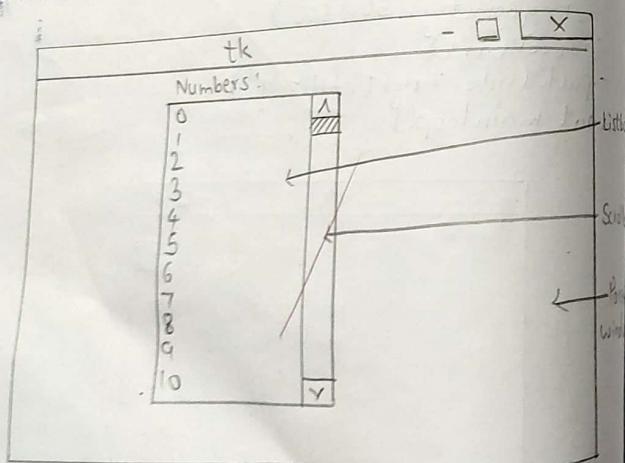
geometry()

scrollbar

```

## 3:
# Using frame widget
from tkinter import *
window = Tk()
window.geometry("680x500")
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.yview)
scrollbar.pack(side="right", fill="y")
for x in range(100):
    listNodes.insert(END, str(x))
window.mainloop()

```



043

3:

Step 1: Import the relevant libraries from the tkinter method.

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

Step 3: Use the geometry manager with pixel size [680x500] or any other suitable pixel value.

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

Step 5: Use the frame widget along with the parent object created and use the pack method.

Step 6: Use the listbox method along with the attributes like width, height, font. Do create a listbox methods object. use pack() for the same.

Step 7: Use the scrollbar() with an object use attribute of vertical. Then configure the same with object created from the scrollbar() and use pack().

Step 8: Trigger, the events using mainloop.

#4:-
Step 1: Import relevant method 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 window.

Step 4:- Create another frame object method termed as the left frame and put it on the parent window on its LEFT side.

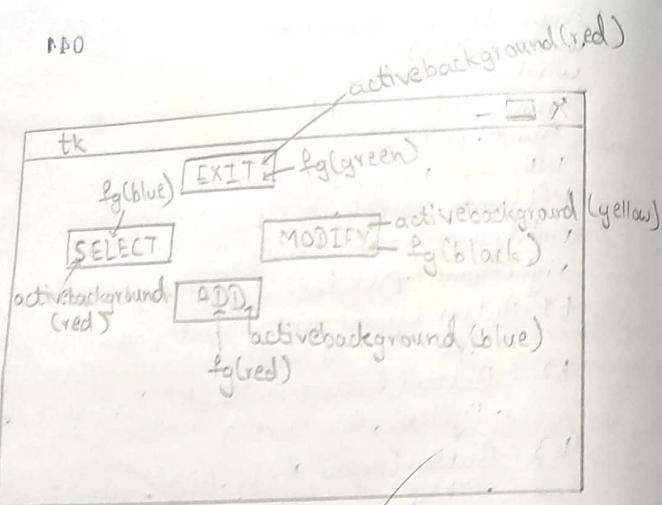
Step 5:- Similarly define the RIGHT frame and subsequently define the button object placed onto the given frame with the attribute as text, active background and foreground.

Step 6:- Now use the pack() along with the side attribute.

Step 7:- Similarly create the button object corresponding to the MODIFY operation. Put it into frame object on side="right".

```
from tkinter import *
window = Tk()
window.geometry("680x500")
frame = Frame(window)
frame.pack()
leftframe = Frame(window)
leftframe.pack(side="left")
rightframe = Frame(window)
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")
b4 = Button(frame, text="EXIT", activebackground
            ="red", fg="green")
b1.pack(side="LEFT", padx=20)
b2.pack(side="right", padx=20)
b3.pack(side="bottom", pady=20)
b4.pack(side="top")
```

040



045

Step 8:- Create another button object & place it onto the RIGHT frame & label the button as ADD.

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

Step 10:- Use the pack() simultaneously for all the objects & finally use the mainloop().

740

Write the program implementing the 5g's style of relief attributes.

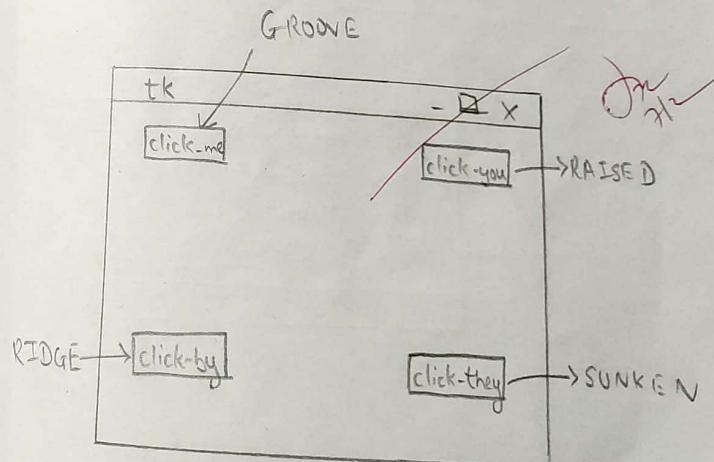
Step 1:- Defined a button object & place it onto the object corresponding the parent window.

Step 2:- Use the text attributes for specifying the title of the bottom object.

Step 3:- Use the relief attributes with one style at time involved event trigger.

Step 4:- On either the packed & grid method for positioning widgets object a onto the parent window & trigger the corresponding event by calling the mainloop method.

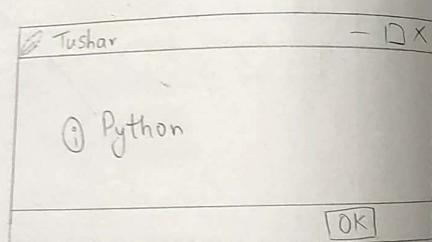
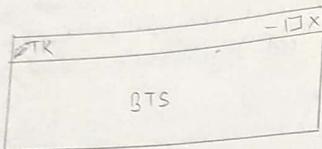
Using Relief Method onto the window & using button
from tkinter import*
root = Tk()
046
b1 = Button(root, text="click-me", fg="black", activebackground="red", relief=GROOVE)
b2 = Button(root, text="click-you", relief=RAISED, fg="black", activebackground="red")
b3 = Button(root, text="click-by", relief=RIDGE, fg="black", activebackground="red")
b4 = Button(root, text="click-they", relief=SUNKEN, fg="black", activebackground="red")
root.mainloop()



```

240
from tkinter import *
top = Tk()
messagebox
def msgb():
    messagebox.showinfo("BTS", "Python")
b1 = Button(top, text="BTS", command=msgb)
b1.pack()
top.mainloop()

```



Practical No: 5 [C]

047

Write a program to implement the message box widget and the different method which these particular widget may assume.

Showinfo()

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

Step 2: The attributes which has a given method takes will specify the 2 strings on related to the title correspond to the message displayed.

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

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

540

Showwarning()

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

Step 2:- The attributes which a given method take will specify the strings in related to the message message displayed to corresponding to the message.

Step 3:- Now create an object from the method and place it on the parent window with the title after button object specified and finally use the command attribute to execute the relevant function.

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

048

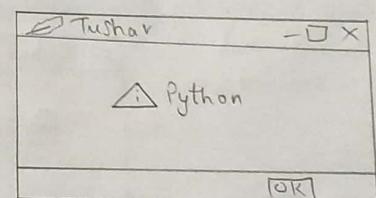
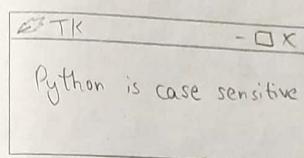
```
from tkinter import *
top = Tk()
messagebox
```

```
def msgb():
```

```
    messagebox.showwarning("Tushar", "Python")
b1 = Button(top, text="Python is case sensitive",
            command=msgb)
```

```
b1.pack()
```

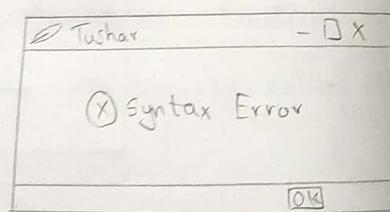
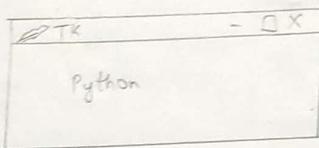
```
top.mainloop()
```



```

840 from tkinter import*
top=Tk()
messagebox
def msgb():
    messagebox.showerror("Tushar","Syntax Error")
bl=Button (top,text="Python",command=msgb)
bl.pack
top.mainloop()

```



049

Show Error()

Step 1:- Define a function `msgb` which will use the `showerror()` derived from the `messagebox` library.

Step 2:- The attribute which a given method `msgb` takes
 will specify 2 string.
 1) related to the title
 2) corresponding to the message

Step 3:- Now create a obj from the button
 method and place it on the parent window
 with the title of the `msgb` obj.

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

Q10

askyesno()

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

Step 2:- The attributes which a given method takes of the window(1) related to the title of the window (1) corresponding to the message displayed.

Step 3:- Now create an obj from the button method and place to the parent window with title of the button obj specified and finally use the command attribute to execute function

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

Q10

```
from tkinter import*
top = Tk()
messagebox
```

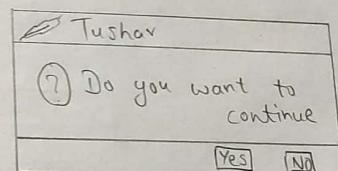
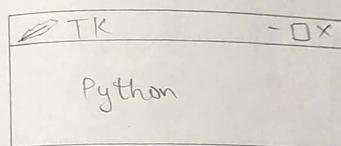
```
def msgb():
```

```
messagebox.askyesno("Tushar", "Do you want to
```

```
continue")
```

```
b1.pack()
```

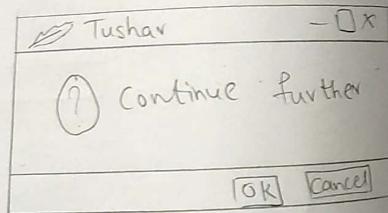
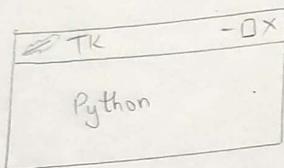
```
top.mainloop()
```



```

050
from tkinter import*
top = Tk()
messagebox.askokcancel("Tushar", "Continue further")
def msgb():
    messagebox.askokcancel("Tushar", "Continue further")
    bb1 = Button(top, text="Python", command=msgb)
    bb1.pack()
top.mainloop()

```



051

askokcancel

Step 1:- Define a function which uses the askokcancel() method from the messagebox library.

Step 2:- The attribute which a given method takes will specify the 2 strings related to the title & corresponding to the message displayed

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

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

120

Practical No: 5[D]

Step 1: Import the relevant method from the tkinter library along with parent window object declared.

Step 2: Use parent window object along with minsize function for window size.

Step 3: Define a function main, declare parent window object and use config(), title(), minsize(), label() as well as no J button and use pack() & mainloop() simultaneously.

Step 4: Similarly define the function along with parent object and declare button with attribute like FLAT, RIDGE, GROOVE, RAISED, SUNKEN, along with the relief widget.

Step 5: Finally call the mainloop() for

052

```

## Multiple window
## Different button (relief())
from tkinter import*
root = Tk()
root.minsize(200,200)

def main():
    top = Tk()
    top.config(bg="black")
    top.title("HOME")
    top.minsize(200,200)
    L = Label(top, text="SAN FRANCISCO")
    L.pack()
    b1 = Button(top, text="next", command=second)
    b1.pack(side=RIGHT)
    b2 = Button(top, text="exit", command=terminate)
    b2.pack(side=LEFT)
    top.mainloop()

def second():
    top2 = Tk()
    top2.config(bg="orange")
    top2.title("")
    top2.minsize(300,300)
    L = Label(top2, text="created by: Tushar Dhule")
    L.pack()
    b1 = Button(top2, text="")
    b1.pack(side=LEFT)
    b2 = Button(top2, text="exit", command=terminate)
    b2.pack(side=RIGHT)
    top2.mainloop()

```

In Golden Gate Bridge In Lombard Street In Chinatown
In Coit Tower)

```

def button():
    top3 = Tk()
    top3.ge
    b1 = Button(top3, text="flat button", relief=FLAT)
    b1.pack()
    b2 = Button(top3, text="groove button", relief=GROOVE)
    b2.pack()
    b3 = Button(top3, text="Raised button", relief=Raised)
    b3.pack()
    b4 = Button(top3, text="Sunken button", relief=SUNKEN)
    b4.pack()
    b5 = Button(top3, text="Ridge button", relief=RIDGE)
    b5.pack()

    top3.mainloop()

def terminate():
    quit()

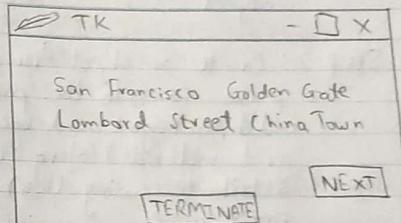
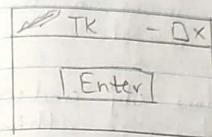
    AI
    b5 = Button(root, text="TOUR DETAILS",
                command=main)
    b5.pack()

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

    root.mainloop()

```

053



Write a program to insert an image in the frame widget using the other widgets.

Step 1:- Create the parent window obj and use the method title, config and minsize within this object.

Step 2:- Create an object from the frame method and place it onto the parent window object with width, height and bgcolour and use the grid method() along with row and column attribute using (0,0) with some extreme positioning.

Step 3:- Similarly create the right frame method with row and column attribute making the values (0,1)

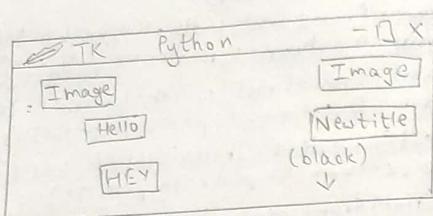
Step 4:- Use the label & the parent window object corresponds to left frame with text and relief attribute and use the grid method with row and column value (0,0).

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

```

from tkinter import *
root = Tk()
root.title ("Python")
root.config(bg = "black")
root.maxsize (200,200)
leftframe = Frame (root, width = 100, height = 200, bg = "white",
grid (row = 0, column = 0, padx = 10, pady = 20)
rightframe = Frame (root, width = 100, height = 200, bg = "black",
grid (row = 0, column = 1, padx = 10, pady = 20)
Label (leftframe, text = "hello", relief = RAISED).grid
(crow = 0, column = 0, padx = 10, pady = 20)
Label (rightframe, text = "hey", relief = SUNKEN)
image = PhotoImage (file = "earth.gif")
on-image = image.subsample (3,3)
Label (rightframe, img = on-image).grid (row = 0,
column = 1, padx = 10, pady = 20)
Label (toolbox, text = "title", relief = SUNKEN).grid
(crow = 0, column = 1)
def abl():
    print ("new title")
toolbox = Frame (leftframe, width = 100, height = 100)
grid (row = 2, column = 0, padx = 10, pady = 10)

```



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 attributes and the row, column specified value specified in grid().

Step 8:- Similarly create the label() using right frame and the image attribute and the row, column value specified in grid().

Step 9:- Now define a function using part print statement which shall be called on clicking the window.

Step 10:- Create the label method position it onto the taskbar with some title.

Step 11:- Now use the mainloop method to terminate the given program.

260

Spin box

Step 1:- Use the tkinter library to import the relevant method.

Step 2:- Create the parent window object.

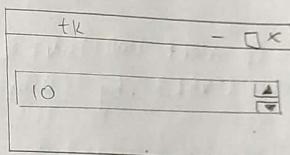
Step 3:- Create an object from a spinbox method and place it on the parent window with the option specified.

Step 4:- Now use the pack method to make the object visible onto the parent window and called the mainloop method.

056

Code :-

```
from tkinter import *
top=Tk()
s1=spinbox (top, from_=0, to=10)
s1.pack()
top.mainloop()
```



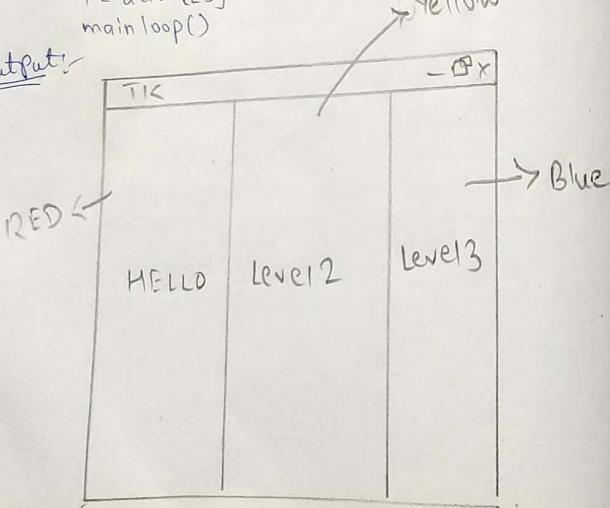
Code :-

```

class Tk:
    def __init__(self):
        self.root = Tk()
        p = PanedWindow(self.root, bg="pink")
        p.pack(fill=BOTH, expand=1)
        l1 = Label(p, text="HELLO", bg="red")
        p.add(l1)
        p1 = PanedWindow(p, orient=VERTICAL, bg="yellow")
        p.add(p1)
        l2 = Label(p1, text="LEVEL 2", bg="green")
        p1.add(l2)
        p2 = PanedWindow(p1, orient=HORIZONTAL, bg="blue")
        p.add(p2)
        l3 = Label(p2, text="LEVEL 3", bg="orange")
        p2.add(l3)
        self.root.mainloop()

```

Output:-



Practical No. 5 [CE]

057

* Paned Window :-

Step 1 : Create an object from the paned window method and use the pack method to make this obj visible.

Step 2 : Now create an obj from the entry widget and place it onto the paned window and use the add method similarly create an object of a paned window.

Step 3 : Create a button widget and place it onto the paned window define a functionality along with the button widget.

Step 4 : Use the pack method & mainloop method for the corresponding event to trigger.

520

Canvas Widget:-

Step 1:- Create an object from the canvas widget by using the attribute height width by colour color for parent window obj.

Step 2:- Use the corresponding method for drawing the simple geometrical shape like arc, oval, a line and specific the coordinate values.

Step 3:- Similarly use the create line & create oval method along with the coordinate values & the fill attribute for specifying the colour.

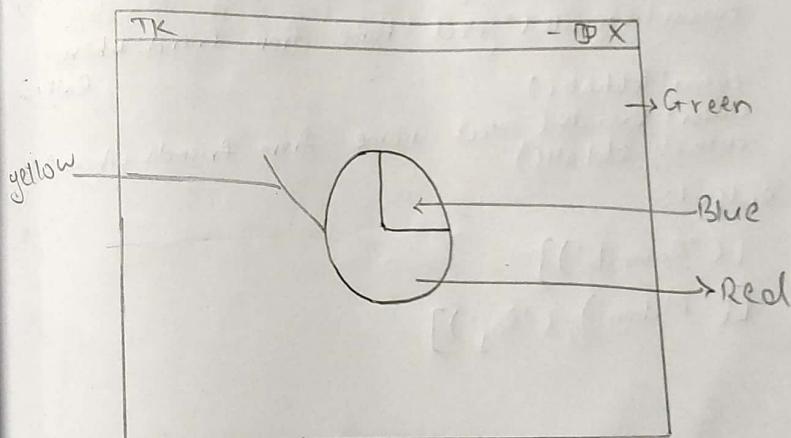
Step 4:- finally use the pack & mainloop method.

Code :-

```
from tkinter import *
```

058

```
root = Tk()  
c1 = canvas(root, height=500, width=500, bg="Green")  
oval = c1.create_oval(500, 12, 12, 400, fill="RED")  
line = c1.create_line(30, 20, 70, 60, fill="YELLOW")  
arc = c1.create_arc(300, 18, 18, 400, fill="Blue")  
c1.pack(side=TOP)  
root.mainloop()
```



Code :-
820.

```
Import sqlite3  
connection = sqlite3.connect("123.db")  
cursor1.execute("Create table friends(Name,char,Roll,int)")  
cursor1.execute("Create table")  
cursor1.execute("insert into friends values('Raj',1854)")  
cursor1.execute("insert into friends values('Ramesh',1925)")  
cursor1.execute("insert into friends values('Suresh',1721)")  
cursor1.execute("insert into friends values('Mohan',1823)")  
connection.commit()  
cursor1.execute("Select name from friends where Roll < 1860")  
cursor1.fetchall()
```

Output:-

[('Ramesh')]

[('Mohan'), ('Raj')]

Practical No.: 6

059

Aim :- To create database and tables and insert values.

Algorithm :-

- 1) Import the corresponding library taking of data base connection.
- 2) Now create connection objects using sqlite library and connecting method for creating the new database.
- 3) Now create the cursor objects using cursor method.
- 4) Now use the executing method for creating the table with the column name and respective data type.
- 5) Now with the cursor objects using cursor method from the connection objects create in steps.
- 6) Use commit method to complete the transition between connection objects.

820

060

- 7) Use the exec execute statement along with the cursor objects for specifying the value to the database using selecting from when clause.
- 8) Finally use the fetchall method for displaying the value for the table using cursor objects.
- 9) Use the execute method and the drop table syntax for terminating the database finally use the close method.

```
bmi.py - CAUsers\sankardhule\Desktop\bmi.py (3.4.4)
File Edit Format Run Options Window Help
from tkinter import *
from tkinter import messagebox

TOP = Tk()

TOP.geometry("400x400")
TOP.configure(background="black")
TOP.title("BMI Calculator")
TOP.resizable(width=False, height=False)

canvas=Canvas(TOP,width=300,height=300)
image=PhotoImage(file="bmi.gif")
canvas.create_image(0,0,anchor=NW,image=image)
canvas.place(x=50,y=250)

def get_height():
    height = float(ENTRY2.get())
    return height
def get_weight():
    weight = float(ENTRY1.get())
    return weight
def calculate_bmi():
    try:
        height = get_height()
        weight = get_weight()
        height = height / 100.0
        bmi = weight / (height**2)
    except ZeroDivisionError:
        messagebox.showinfo("Result", "Please enter positive height!")
    except ValueError:
        messagebox.showinfo("Result", "Please enter valid data!")
    else:
        if bmi <= 15.0:
            res = "Your BMI is " + str(bmi) + "\nRemarks: Very severely underweight"
            messagebox.showinfo("Result", res)
        elif bmi > 15.0 and bmi <= 16.0:
            res = "Your BMI is " + str(bmi) + "\nRemarks: Severely underweight"
            messagebox.showinfo("Result", res)
        elif bmi > 16.0 and bmi < 16.5:
            res = "Your BMI is " + str(bmi) + "\nRemarks: Underweight"
            messagebox.showinfo("Result", res)
        else:
            res = "Your BMI is " + str(bmi) + "\nRemarks: Normal"
            messagebox.showinfo("Result", res)
```

```
bmi.py - C:\Users\sankardhule\Desktop\bmi.py (3.4.4)
File Edit Format Run Options Window Help
messagebox.showinfo("Result", res)
else:
    if bmi <= 15.0:
        res = "Your BMI is " + str(bmi) + "\nRemarks: Very severely underweight!"
    elif bmi > 15.0 and bmi <= 16.0:
        res = "Your BMI is " + str(bmi) + "\nRemarks: Severely underweight!"
    elif bmi > 16.0 and bmi < 18.5:
        res = "Your BMI is " + str(bmi) + "\nRemarks: Underweight"
    elif bmi >= 18.5 and bmi <= 25.0:
        res = "Your BMI is " + str(bmi) + "\nRemarks: Normal."
    elif bmi > 25.0 and bmi <= 30:
        res = "Your BMI is " + str(bmi) + "\nRemarks: Overweight."
    elif bmi > 30.0 and bmi <= 35.0:
        res = "Your BMI is " + str(bmi) + "\nRemarks: Moderately obese!"
    elif bmi > 35.0 and bmi <= 40.0:
        res = "Your BMI is " + str(bmi) + "\nRemarks: Severely obese!"
    else:
        res = "Your BMI is " + str(bmi) + "\nRemarks: Super obese!"
    messagebox.showinfo("Result", res)
TABLE1 = Label(TOP, bg="light blue", text="Enter Weight (in kg):",
               font=("Helvetica", 10, "bold"))
TABLE1.place(x=80, y=30)
ENTRY1 = Entry(TOP, width=6)
ENTRY1.place(x=230, y=30)
TABLE2 = Label(TOP, bg="red", text="Enter Height (in cm):",
               font=("Helvetica", 10, "bold"))
TABLE2.place(x=80, y=80)
ENTRY2 = Entry(TOP, width=6)
ENTRY2.place(x=230, y=80)
BUTTON = Button(bg="yellow", bd=10, text="BMI", padx=33, pady=15, command=calculate_bmi,
                font=("Helvetica", 15, "bold"))
BUTTON.grid(row=3, column=0, sticky=W)
BUTTON.place(x=120, y=150)
TOP.mainloop()
```



