

```
# Write, reading, reading methods.
```

```
f = open("abc.txt", "w")
f.write("My name is Sungi" + "\n")
f.close()
f = open("abc.txt", "a")
f.write("studying a professional course")
f.close()
print(f.read(), "\n") / f = open("abc.txt", "r")
print(f.read(), "\n")
f = open("abc.txt", "r")
print(f.readlines())
f = open("abc.txt", "r")
print(f.readlines())
f.close()
```

```
# attributes of a file.
```

```
print(f.closed)
print(f.mode)
print(f.name)
print(f.softspace)
```

```
# input method
```

```
>>> a = rawinput("Enter string")
Enter string: Hello
>>> a
'Hello'
>>> b = input("Enter expression:")
Enter expression: 8 * 9
>>> c
```

72

O/P

```
My name is Sungi
studying a professional course
My name is Sungi
[My name is Sungi 'n', 'studying
  a professional course']
True
"
abc.txt
0
```

PRACTICAL No.1

Aim-

Programs to read and write files, attributes, read operations and accessing mode of a file.

Algo 1-

1. Open a file with write accessing mode and assign it to be variable (file object).
2. Write a content in opened file and close it afterwards
3. Open the same file with append accessing mode, assigning it to a variable (file object)
4. Write an append new content is opened file and close it
5. Open the file in reading accessing mode and read the content of the specified file.
6. Open the file in reading accessing mode and use read, readline, readlines method and print their o/p.
7. Use attributes of a file and print them.

Algo 2-

1. Use raw input with will accept input as a string
2. Use input method which will take expression and will return value of it.

Algo 3-

1. Create a file object and open the file from algo1 in `w` accessing mode, an `a` accessing mode.

- 2 Use read method to read the 5 character from the file and display the content.
- 3 Now use the seek method with two arguments as zero and then certain number of character and display the read characters similarly use of seek method with zero and one value indicating the reference position of a cursor as current location, while zero and two indicates the referenced position of the cursor from the end of the file.
- * Program to put '*' after every letter -

Algo -

- 1 Create a file object and open a file in read mode.
- 2 Read one character from file and store it in a variable
- 3 Use while conditional statement till the length of variable in step 2 is zero.
- 4 Print the variable from step 2 and use 'end' method to print 'R' and increase the counter by reading the content of file once again.

Program to find the length of each line of the content

Algo -

- 1 Create a file object and open a file in rt accessing mode.
- 2 Write a content in a file along the file object.
- 3 Use readline method and print it, also assign it to variable.
- 4 Print the variable from step 2 and use 'end' subsequently

```
# seek and tell
f = open("abc.txt", "w")
f.write("Hello world")
f.close()
f = open("abcd.txt", "r")
print(f.read(5))
print("Current position:", f.tell())
print("Position (referenced) changed to 'beginning'; f.seek(0,0)")
print(f.read(7))
print("reference position", f.seek(0,1), f.read())
print("position changed to end:", f.seek(0,2))

o/p
Hello
Current position :5
Current position changed to beginning :0
Hello w
Reference position 7
and
position changed to end :11
```

program for adding '*' after every letter.

```
g=open ("abcd.txt", "r")
c=g.read(1)
while len(c)>0:
    print (c, end ='*')
    c=g.read(1)
```

O/P

h*e*i*l*o*w*n*i*t*d*

program for counting length of a line in a file.

```
f=open ('abc.txt', "r")
text=f.readlines()
i=0
for l in text:
    print (len(l))
    i+=1
```

O/P

My name is Furaj 17

Studying in professional course 30

print length of line

Practical No. 2

Aim-

Demonstrate the use of iterations and iterables

Program:-

Algo

- Define a variable of list data type containing names of fruit.
- Now iterate over that variable and print each element using next method or some conditional loops statements

Program 2--

To display odd no. till 6.

Algorithm-

- Create a class within that define a iter method with one argument and initialize the value and return value.
- Define the next method with one argument.
- Use if conditional statement to check whether the variable in 1st step is smaller than 16.
- Increase the variable's value by 2 and return it.
- Use else to stop iteration.
- Create an object of given class and pass the object in iter method.
- Use while conditional loop to print the next method.

1] code
`i = ["Apple", "Banana"]
 myiter = iter(i)
 for k in i:
 print(k)`

O/P
 Apple
 Banana.

2] code
 class odd:
`def __init__(m):
 m.value = 1
 return m
def __next__(m):
 if m.value < 15:
 m.value += 2
 return m.value
 else:
 raise StopIteration
y = iter(odd())
while True:
 print(next(y))`

O/P
 1
 3
 5
 7
 9
 11
 13
 15

3] code -
 class fact:
`def __iter__(m):
 m.value = 1
 return m
def __next__(m):
 m.value *= 1
 return m
def __next__(m):
 m.value += 1
 if (m.value == 7):
 raise StopIteration`

facto=1
 for i in range(1, m.value + 1):
 facto = facto * (i)
 print("The factorial of",
 ", is", facto)
 m.value += 1
 if (m.value == 7):
 raise StopIteration

```
y=iter(fact())
while True:
    next(y)
```

O/P

The factorial of 1 is 1
The factorial of 2 is 2
The factorial of 3 is 6
The factorial of 4 is 24
The factorial of 5 is 100.
The factorial of 6 is 720.

de.

class power:

```
def __init__(pow):
    pow.num = 1
    pow.num = int(input("Enter number"))
    return pow
```

```
def __next__(pow):
    if pow.num <= 6:
        nu = pow.num ** pow.num
        pow.num += 1
        return nu
```

```
else:
    raise StopIteration
    print(power())
True:
```

```
print(next(y))
```

5] code

```
class range1:
    def __init__(o):
        o.num = 0
    return o
    def __next__(o):
        if o.num <= 10:
            nu = o.num
            o.num += 1
            return nu
        else:
            raise StopIteration
y=iter(range1())
while True:
    print(next(y))
```

O/P

Enter number 2	O/P
2	2
4	3
8	4
16	5
32	6
64	7
	8
	9
	10

Program 5-

To display the numbers in range 10.

Algorithm

1. Create a class and define iter method with an arguments, initialize a value and return it.
2. Now define next method check whether the initialized value is smaller than 10, print the value.

Program 3-

Factorial of number till 10.

Algo-

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

Program 4-

Power of number taken from user.

Algorithm -

1. Create a class and define a iter method with arguments, initialize.

SS

a value . and return it also take a input from user.

2. Define a next method with an argument
3. If the initialized value is bigger than 5 that make stop iteration or else return the multiplication of arguments by itself.
4. Create a object of given class and pass it to iter method and use while . loop to we .next method

1) Code

```
# arithmetic error
while True:
    try:
        x = int(input("Enter class"))
        break
    except ValueError:
        print("Enter Numeric Value")
```

>>> Enter class:fgs

```
Enter Numeric value
Enter class:15
```

2) code

```
try:
    fo=open('abc.txt','w')
    fo.write("CS")
except IOError:
    print("Error writing onto file")
else:
    print("Operation successful")
fo.close.
```

>>> Operation successful.

PRACTICAL No. 3

Topic - Exception Handling
 1. WAP using exception method of nature arithmetic error.

Algo-

1. Use try block and accept the input using input method and convert it into integer datatype and subsequently terminate the block.
2. Use the except block with the reception name as value errors and display the appropriate message if code is part of try block.
3. WAP of accepting file in a given mode and use environment error as reception for given input.

Algo-

1. Within the try block open the file using the write mode and write some content onto the file.
2. Use the except block with IO error and display the message regarding missing of file or incompatibility of mode.
3. Use the else block to display the message that operation is carried out successfully.

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

Age -

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

Use the condition to check whether the input age in given age and if return the age use the user exception.

Define while loop to check whether the expression holds the try block to accept age of students and terminate the looping condition.

Use the except with value error and print the message not a valid age.

Write program using assert method to check if the list elements are empty.

1. Define a function which accepts an argument and the check using list in an empty list and accordingly return the message.

2. Close the function and body of program define certain elements in list and take some action

code

```
def acceptage():
    age = int(input('Enter age'))
    if age > 30 or age < 10:
        raise ValueError
    return age
valid = False
while not valid:
    try:
        age = acceptage()
        valid = True
    except ValueError:
        print('Not a valid age')
```

```
>>>
Enter age : 10
Not a valid age
Enter age : 20
```

code

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

```
>>> list is empty
None
```

*Last
Date 21/07*

```

1] code
# repace.
import re
string = 'abc def ghi'
pattern = r'\ls+'
replace = ''
v1 = re.sub(pattern, replace, string)
print(v1)

>> abcdefghi

2] code
# group()
import re
sequence = 'python is interpreted language'
v = re.search(r'Apthon', sequence)
print(v)
v1 = v.group()      >>>
                    <_sre.SRE_Match object; span=(0, 6),
                    match='python'>

3] code
# list
import re
list = ['8257210522', '1253457890', '7852100589']
for value in list:
    if re.match(r'[8-9]\d{9}', value):
        print('Match for cellNumbers')
    else:
        print('Match failed')
        >>> Match failed:
                    Match failed:
                    Match failed:

```

PRACTICAL No. 4.

Topic - Regular Expression.

Step 1 - Import module declare pattern and declare sequence use match method with declare arguments if arguments matched than print the same otherwise print pattern NOT found!

Step 2 - Import re module declare pattern with literal and meta character. Declare string value use the.findall() with arguments and print the same.

Step 3: Import re module declare pattern with meta data character use the split() and print the O/P.

Step 4- Import re module declare string and accordingly declare pattern blank space with no-space use .sub() with 3 arguments and print the string without spaces.

Step 5- Import re module declare a sequence use search method for finding subsequently use group() with dot operator or search() gives memory corruption using group() it will show up matched string.

Step 6- Import .re module declare list with numbers use the conditional statements here we have used if for condition statement use if condition for checking first number is either 8 or 9 and next number are in range of 0 to 9 and check whether the entered number are equal to 10.

If criteria matches print one number matches otherwise print failed.

Step 7 -

Import re module declare a string use the module with findall() for finding the vowels in string and declare the same.

Step 8 -

Import re module declare the host and domain name declare pattern for separating the host and domain name use the findall() and print the output respectively.

Step 9 -

Import re module enter a string use pattern to display only two elements of partition string use findall() declare variables with initial value as zero use for condition and subsequently use if condition check whether condition satisfy odd up to or else increment value and display the values subsequently.

counting of first 2 Letter

```
import re
```

```
s='mr.a,ms.b,mc.c,mc.t'
```

```
p=re.findall(r'[ms]/[mc]t')
```

```
o=re.findall(p,s)
```

```
print(o)
```

```
m=0[zero]#
```

```
f=0[zero]#
```

```
for v in o:
```

```
    if(v=='ms'):
```

```
        f=f+1
```

```
else:
```

```
    m=m+1
```

```
print("No.of Males is :"m)
```

```
print("No.of Females is :"f)
```

```
>>> [ms',ms','ms','ms']
```

```
No.of Males : 1
```

```
No.of Females : 3
```

#vowels

```
import re
```

```
str='Save Earth'
```

```
output=re.findall(r'\b[aeiouAEIOU]\w+',str)
```

```
print(output)
```

```
>>> ['Earth']
```

#host and domain

```
import re  
seq = 'abc.tsc@edu.com , xyz@gmail.com'  
pattern = r'[w]+.-]+[wl.-]'  
output = re.findall(pattern, seq)  
print(output)  
>>> ['abc:tsc', 'edu.com', 'xyz', 'gmail.com']
```

Yash
5/12 -

PRACTICAL NO.5

Aim- GUI components.

Step 1- Use the primary key tkinter library for importing the feature of text widgets.

Step 2- Create a variable from the Text method and position it on parent window

Step 3- Use the pack method along with the object created from Text method.

Step 4- Use the mainloop method for triggering of corresponding events.

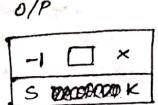
Step 5- Use the tkinter library for imposing the feature of text widget

Step 6- Create a variable from Text method and position in onto parent window

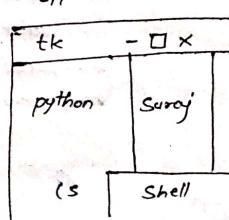
Step 7- Use the pack method along with the object (created from Text method) and use the parameter

Step 8- side LEFT
side LEFT
side TOP
 , padx = 20
 , pady = 30
 , padx = 40

```
#  
from tkinter import *  
root = Tk()  
l1 = Label(root, text = "Sun")  
l1.pack(padx = 10, pady = 50, side = TOP)  
l2 = Label(root, text = "Sun")  
l2.pack(ipadx = 50, ipady = 100, side = LEFT)  
root.mainloop()
```



```
#  
from tkinter import *  
root = Tk()  
l1 = Label(root, text = "Sun")  
l1.pack(padx = 10, pady = 50, side = TOP)  
l2 = Label(root, text = "Sun")  
l2.pack(ipadx = 50, ipady = 100, side = LEFT)  
t1 = Text(root)  
quote = "Platinum is the New Gold"  
t1.insert(END, quote)  
t1.pack(padx = 50, ipady = 10, side = TOP)  
root.mainloop()
```



side = TOP ; ipady = 50

#C Radio button

```

from tkinter import *
root = Tk()
def set():
    selection = 'you selected the option ' + str(v.get())
    l = Label(root, text=selection, justify=LEFT)
    l.pack(anchor=S)
    v = IntVar()
    r1 = Radiobutton(text="option 1", variable=v,
                      value=1, command=set)
    r1.pack()
    r2 = Radiobutton(text="option 2", variable=v,
                      value=2, command=set)
    r2.pack()
    r3 = Radiobutton(text="option 3", variable=v,
                      value=3, command=set)
    r3.pack()
    root.mainloop()

```

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

step 10- Now repeat the above steps with the label method which takes the following arguments:

- 1) Text .attribute which defines string.
- 2) Bg .(background) colour.
- 3) Name .of the window
- 4) None of the pack method with sele .value .padding attribute.

Radio button

WAP making use of control variable and button widget for selection of given option

~~step 1- Use the tkinter to import relevant method~~

~~step 2- Define the configuration method along with the label object and call the variable as an argument within the method~~

~~step 4- Now define the parent window and define the option using control variable~~

~~step 5- Now create an object from the radio button method which will take the argument~~

- 1) Positioning on parent window
- 2) Defining the .text variable [1,2,3,4]

- 3) Define the variable argument based on user choice
 3) Corresponding value and trigger the given function

step 6- Pick method for corresponding studio object to created and specify the attribute as an anchor attribute.

step 7- Now define the label object from the corresponding method and place it on parent window subsequently we pack method for window and make use of main loop method.

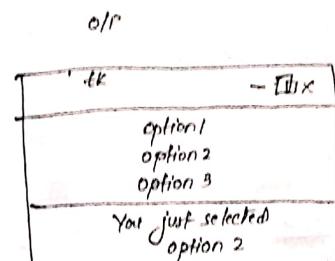
step 8- Import the relevant method from tkinter library.

Step 9- Define the object corresponding on above window ; and define parent window in terms of no. of pixels.

step 10- Define the frame object from the method and place it onto parent window.

step 11- Create another frame object the left frame and put it on parent window on itself side.

step 12- Similarly define the right frame and subsequently define button placed onto given frame with attribute as text active bg and fg.



frame object

from tkinter import *

top = Tk()

top.geometry("100x200")

frame = Frame (top)

left = Frame (top)

left.pack (side = LEFT)

b1 = Button (frame, text = "select", active bg = "red", fg = "black")

b1.pack()

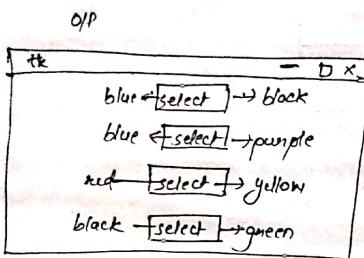
b2 = button (frame, text = "modify", active background = "blue", fg = purple)

b2.pack()

b3 = button (frame, text = "Add", active background = "yellow", fg = red)

b3.pack()

top.mainloop()



37

Step 13 -

Now use the pack method along with the side attribute

Step 14 - similarly create the button object corresponding to modify option and put in frame object with side equal to right attribute set.

Step 15 - Add another button and put it on right frame object and turn it as EXIT.

Step 16 - Use the pack method for all object and finally use the main loop method.

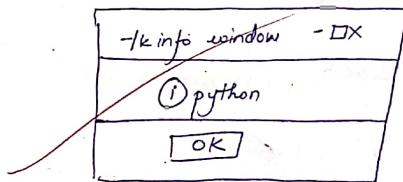
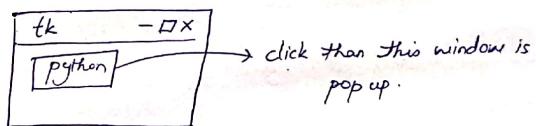
8/2/19

messagebox
Aim - GUI components.

- Step 1 - Import the relevant method from Tkinter library
- Step 2 - Import messagebox
- Step 3 - Define a parent window object along with parent window.
- Step 4 - Define a function with will use messagebox with showinfo method along with method along with info window attribute.
- Step 5 - Declare a button with parent window object along with command attribute.
- Step 6 - Place the button widget onto the parent window and finally call mainloop() for triggering of the events called above.

```
#messagebox
from Tkinter import *
import tkMessageBox
root = Tk()
def function():
    tkMessageBox.showinfo("info window", "python")
    b1 = Button(root, text = "python", command = function)
    b1.pack()
root.mainloop()
```

O/P



```
# Multiple Window
# Different button (Relief, etc())
from Tkinter import *
root = Tk()
root.minsize(300, 300)
def main():
    top = Tk()
    top.config(bg = "black")
    top.title("HOME")
```

```
L = Label (top, text = "SAN FRANCISCO")
In place of interest. In Golden gate
bridge. In Lombard street. In chinatown.
In coit Tower.)
L.pack()

b1 = button (top, text = "next"; command = second)
b1 = button (top, text = "exit", command = terminate)
b2.pack (side = LEFT)
top.mainloop()
```

39

step 1-

Import the relevant methods 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 second and use the attribute accordingly and use config(), title(), minsize(), label() has we use button () and use pack() & mainloop() simultaneously.

step 4-

Declae another function button along with parent object and declare button with attributes like FLAT, RIDGE, GROOVE, RAISED, SUNKEN, along with relief widget

step 5- Similarly define the function second and use attribute accordingly.

step 6- Finally called the mainloop() for event driven programming

Dm

#multiple windows.

40

```
def second():
    top2 = Tk()
    top2.config(bg = "orange")
    top2.title("About US!")
    top2.minsize(300, 300)
    L = Label(top2, text = "Created by: Tanvi
Name in for details contact to our official
account")
    L.pack()
    b3 = button(top2, text = "prev", command = main)
    b3.pack(side = LEFT)
    b2 = .button.(top 2, text = "exit", command
    b2.pack() → top2.mainloop()

def button():
    top3 = Tk()
    top3.geometry("300x300")
    b3 = Button(top3, text = "raised button", relief
    -RAISED")
    b3.pack()
    b1 = Button(top3,
    text = "flat button", relief = FLAT)
    b1.pack()
    b2 = Button(top3, text = "sunken button", relief
    = SUNKEN)
    b2.pack()
    b4 = button.(top 3, text = "ridge button")
    b4.pack()
    relief = RIDGE
    top3.mainloop()
    def terminate():
        quit()
    b5 = Button(root, text = "TOUR DETAILS")
    command = main
    b5.pack()
    root.mainloop()
```

#transcuring.

```

from tkinter import *
root = Tk()
def main():
    root = Tk()
    root.config(bg="Green")
    root.title("Main")
    root.minsize(200, 200)
    l = Label(root, text="ENTER!")
    l.pack()
    l1 = Label(root, text="Next", command=se)
    B1 = Button(root, text="Next", command=terminate)
    B1.pack(side=RIGHT)
    B2 = Button(root, text="Terminate", command=ter)
    B2.pack(side=BOTTOM)
    root.mainloop()

def se():
    master = Tk()
    master.config(bg='RED')
    master.title("Pg. 2")
    master.minsize(400, 200)
    l2 = Label(master, text="Workout")
    l2.pack()
    l3 = Label(master, text="Workout Health")
    l3.pack()
    B3 = Button(master, text="Prev", command=main)
    B3.pack(side=LEFT)
    B4 = pack(side=BOTTOM)
    master.mainloop()

```

Aim- Displaying the image spinbox

Algo-

1. Create an object corresponding in parent window and use the following 3 methods - Title, config, maxsize.
2. Create a leftzone frame object from frame method and place it onto the parent window with the height, width and bg specified.
3. Now create rightframe object frame method with width, height- specified the now column value.
4. Create a label object from the label method onto the leftframe with text attribute denoting original image with attribute denoting original image with relief attribute used as RAISED subsequently use of grid method with now, column value specified as (0,0) with some , external padding view
5. Now use photo image method with file attribute specified
6. Use the subsample method with object at image X given by x,y co-ordinate value.
7. Use the label method, position onto the leftframe and placing the image after the sampling use the grid method for positioning front row.
8. Create label object positioning it into the leftframe & specifying background attribute with now and column .attribute specifying as (0,0)
9. Now define various function for different toolbar options provided in leftframe.
10. Define the various object from frame method position it into the leftframe which height width provided with leftframe.

- a. from label method position it on to toolbar with the next title
 personal information & position it same row next column.
 b. Now make use of mainloop() method

```

def test1()
    quit()
B5 = Button(root, text = "ENTER", command = main)
B5.pack()
root.mainloop()

# spinbox,
from tkinter import *
root = Tk()
root.minsize(1000,900)
root.config(bg = 'black')
leftframe = frame(root, bg = "pink", height = "200", width = "200")
leftframe.grid(row = 0, column = 0)
rightframe = frame(root, bg = "green", height = "300", width = "250")
rightframe.grid(row = 0, column = 0)
label(leftframe, text = "photo", height = 2, width = 20)
grid()
image1 = photoimage(file = 'dance.gif')
image1.subsample(1,2)
image2 = photoimage(file = "diggif")
image2.subsample(3,2)
label(leftframe, image = image1).grid(row = 0, column = 0)
label(rightframe, image = image2).grid(row = 0, column = 0)

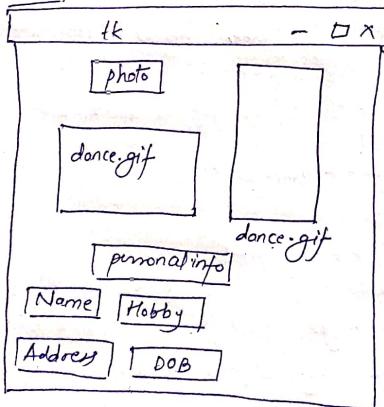
```

```

toolbar = frame (leftframe, width=200,
                 height=400, bg="white")
grid (row=2, column=0)
label (toolbar, text="personal info")
height=2, width=20, relief=RAISED)
grid (row=0, column=0)
def name():
    print ("Name:")
def add():
    print ("Address: Number")
def dob():
    print ('dob : 07-07-2001')
button (toolbar, text='Name', height=1, width=16,
        command=name).grid (row=1, col=0)
Button (toolbar, text="Address", command=add)
grid (row=0, column=0)
root.mainloop()

```

O/P



Aim- GUI component (paned window, canvas)

paned window

Algo

1. Create an object from paned window () and use the pack() with attribute fill and expand.
2. Create an object from label method & put it into the panel window with the .label.(text) attribute to add method to add new object
3. Similarly, create a second paned window object, add into the first paned window with orientation specified.
4. Now create another level object place in onto the second paned window & add .in onto the second trigger the mainloop.

Canvas Widget

Algo.

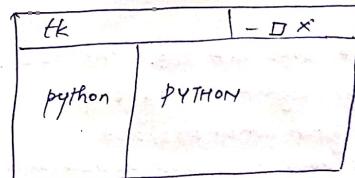
1. Create an object from the canvas method and use the attribute height, width, by paned window object.
2. Use the method create_line, create oval & create arc along the canvas to created & use the co-ordinate values

panedwindow

Code -

```
from tkinter import *
root = Tk()
p = PanedWindow(bg="pink")
p.pack(fill=BOTH, expand=1)
l = Label(p, text="python")
p.add(l)
p1 = PanedWindow(p, orient=VERTICAL, bg="white")
p.add(p1)
l1 = Label(p1, text='python')
p1.add(l1)
root.mainloop()
```

O/P



canvas

Code -

```
from tkinter import *
root = Tk()
c = canvas(root, height=200, width=250, bg="orange")
arc = c.create_arc(10, 20, 30, 40, start=0, extent=100,
                   fill="red")
line = c.create_line(50, 60, 70, 80)
oval = c.create_oval(90, 100, 110, 120)
c.pack()
root.mainloop()
```

```

import sqlite3
conn = sqlite3.connect('student.db')
cur = conn.cursor()
cur.execute("create table sinfo (RNO int, name text, DOB date)")
sqlite3 cursor object at 0x2f62f20
cur.execute('insert into sinfo values(01, "Anuj", "02-02-2001")')
sqlite3 cursor object at 0x02f6f20
cur.execute('select DOB from info')
sqlite3 cursor object at 0x02f6f20
cur.fetchall()
cur.execute("update sinfo set DOB = '03-05-2001' where RNO=3")
sqlite3 cursor object at 0x02f6f20
sqlite3.cursor object at 0xf62f205
cur.execute('alter table sinfo add address text')
sqlite3.cursor object at 0x2f525205
cur.close()

```

```

[('02-09-2001'), ('09-02-2001')]
cur.execute('select DOB from info')
cur.fetchall()

```

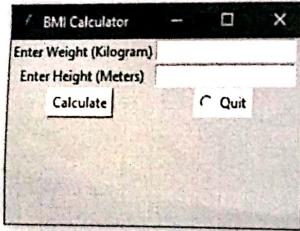
database connectivity Database Connectivity.

Algo-

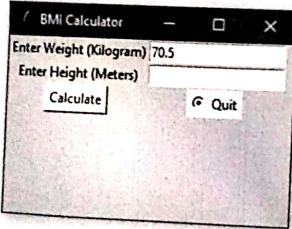
1. Import the relevant libraries for database and the operating system functionality
2. Now create an object for working the connection to given database.
3. Further create object corresponding the cursor area for execution of different query statement.
4. Use of cursor object so created for implementing the structure of database and values with the DS.
5. Use the execute() for implementation of select for filtering the information.
6. Now use the fetchall() along with cursor object for displaying the value onto the screen.

Project 1
GUI Components.

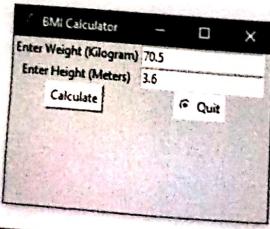
-- RESTART: D:\Projects\Bsc\SKBMI.py ---



-- RESTART: D:\Projects\Bsc\SKBMI.py ---



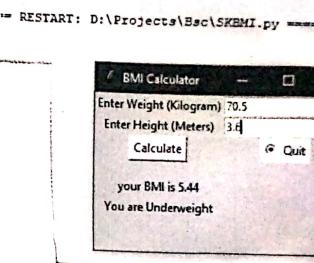
-- RESTART: D:\Projects\Bsc\SKBMI.py ---



```

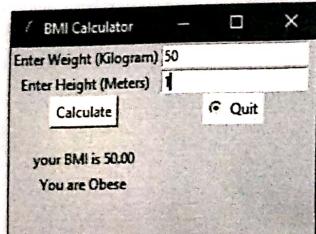
from tkinter import *
root=Tk()
root.configure(background="light green")
root.title("BMI Calculator")
root.geometry("265x160")
def calculate():
    weight=float(kgs.get())
    height=float(ms.get())
    bmi=float((weight)/(height**2))
    bmino.set("\nyour BMI is %0.2f"%bmi)
    if bmi<18.5:
        status.set("You are underweight")
    if 18.5<=bmi<25:
        status.set("You are normal")
    if 25<=bmi<30:
        status.set("You are overweight")
    if 30<=bmi>30:
        status.set("You are obese")
Label(root,text="Enter weight in kgs",bg="light green").grid(row=0)
kgs=StringVar()
e1=Entry(root,textvariable=kgs).grid(row=0,column=1)
Label(root,text="Enter height in meter",bg="light green").grid(row=1)
ms=StringVar()
e2=Entry(root,textvariable=ms).grid(row=1,column=1)
b=Button(root,text="Calculate",command=calculate).grid()
Radiobutton(root,text="Quit",value=0,command=root.destroy).grid(row=2,column=1)
bmino=StringVar()
Label(root,textvariable=bmino,bg="light green").grid()
status=StringVar()
Label(root,textvariable=status,bg="light green").grid()
root.mainloop()

```

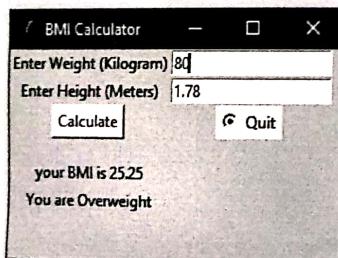


84

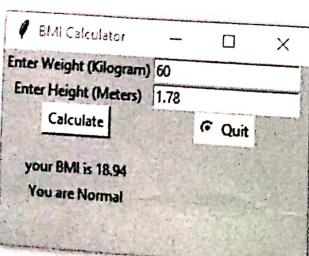
== RESTART: D:\Projects\Bsc\SKBMI.py ==



== RESTART: D:\Projects\Bsc\SKBMI.py ==



== RESTART: D:\Projects\Bsc\SKBMI.py ==



```
from tkinter import *
import sqlite3,sys

def connection():
    try:
        conn=sqlite3.connect("student.db")
    except:
        print("cannot connect to the database")
    return conn

def verifier():
    a=b=c=d=e=f=0
    if not student_name.get():
        t1.insert(END,"<>Student name is required<>\n")
        a=1
    if not roll_no.get():
        t1.insert(END,"<>Roll no is required<>\n")
        b=1
    if not branch.get():
        t1.insert(END,"<>Branch is required<>\n")
        c=1
    if not phone.get():
        t1.insert(END,"<>Phone number is required<>\n")
        d=1
    if not father.get():
        t1.insert(END,"<>Father name is required<>\n")
        e=1
    if not address.get():
        t1.insert(END,"<>Address is Required<>\n")
        f=1
    if a==1 or b==1 or c==1 or d==1 or e==1 or f==1:
        return 1
    else:
        return 0

def add_student():
    ret=verifier()
    if ret==0:
        conn=connection()
        cur=conn.cursor()
        cur.execute("CREATE TABLE IF NOT EXISTS STUDENTS(NAME TEXT,ROLL_NO INTEGER,BRANCH TEXT,PHONE_NO INTEGER,FATHER TEXT,ADDRESS TEXT)")
        cur.execute("insert into STUDENTS values(?,?,?,?,?,?)",
        (student_name.get(),int(roll_no.get()),branch.get(),int(phone.get()),father.get(),address.get()))
        conn.commit()
        conn.close()
        t1.insert(END,"ADDED SUCCESSFULLY\n")

def view_student():
    conn=connection()
    cur=conn.cursor()
    cur.execute("select * from STUDENTS")
    data=cur.fetchall()
    conn.close()
    for i in data:
        t1.insert(END,str(i)+"\n")

def delete_student():
    ret=verifier()
    if ret==0:
```

```
conn=connection()
cur=conn.cursor()
cur.execute("DELETE FROM STUDENTS WHERE ROLL_NO=?", (int(roll_no.get()),))
conn.commit()
conn.close()
t1.insert(END,"SUCCESSFULLY DELETED THE USER\n")

def update_student():
    ret=verifier()
    if ret==0:
        conn=connection()
        cur=conn.cursor()
        cur.execute("UPDATE STUDENTS SET NAME=?,ROLL_NO=?,BRANCH=?,PHONE_NO=?,FATHER=?,ADDRESS=?  
where ROLL_NO=?", (student_name.get(),int(roll_no.get()),branch.get(),int(phone.get()),father.get(),address.get(),int(roll_no.get())))
        conn.commit()
        conn.close()
    t1.insert(END,"UPDATED SUCCESSFULLY\n")

def close():
    sys.exit()

if __name__ == "__main__":
    root=Tk()
    root.title("Student Management System")

    student_name=StringVar()
    roll_no=StringVar()
    branch=StringVar()
    phone=StringVar()
    father=StringVar()
    address=StringVar()

    label1=Label(root,text="Student name:")
    label1.place(x=0,y=0)

    label2=Label(root,text="Roll no:")
    label2.place(x=0,y=30)

    label3=Label(root,text="Branch:")
    label3.place(x=0,y=60)

    label4=Label(root,text="Phone Number:")
    label4.place(x=0,y=90)

    label5=Label(root,text="Father Name:")
    label5.place(x=0,y=120)

    label6=Label(root,text="Address:")
    label6.place(x=0,y=150)

    e1=Entry(root,textvariable=student_name)
    e1.place(x=100,y=0)

    e2=Entry(root,textvariable=roll_no)
    e2.place(x=100,y=30)

    e3=Entry(root,textvariable=branch)
    e3.place(x=100,y=60)
```

```

e4=Entry(root,textvariable=phone)
e4.place(x=100,y=90)

e5=Entry(root,textvariable=father)
e5.place(x=100,y=120)

e6=Entry(root,textvariable=address)
e6.place(x=100,y=150)

t1=Text(root,width=80,height=20)
t1.grid(row=10,column=1)

b1=Button(root,text="ADD STUDENT",command=add_student,width=40)
b1.grid(row=11,column=0)

b2=Button(root,text="VIEW ALL STUDENTS",command=view_student,width=40)
b2.grid(row=12,column=0)

b3=Button(root,text="DELETE STUDENT",command=delete_student,width=40)
b3.grid(row=13,column=0)

b5=Button(root,text="CLOSE",command=close,width=40)
b5.grid(row=15,column=0)

root.mainloop()

```

