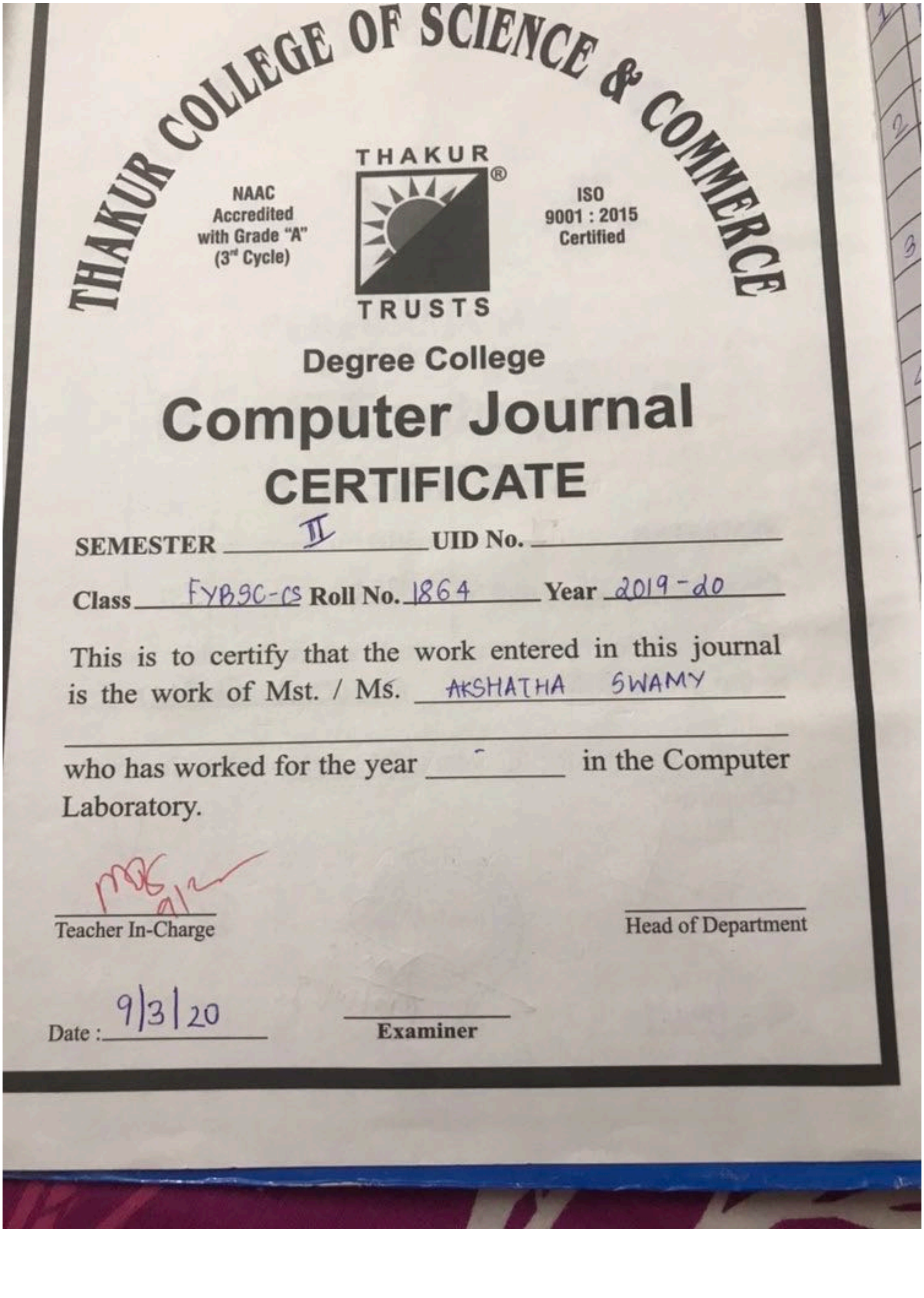


THAKUR COLLEGE OF SCIENCE & COMMERCE

THAKUR TRUSTS

NAAC Accredited with Grade "A" (3rd Cycle)

ISO 9001 : 2015 Certified

Degree College

Computer Journal

CERTIFICATE

SEMESTER II UID No. ____________

Class FYBSC-CS Roll No. 1864 Year 2019-20

This is to certify that the work entered in this journal is the work of Mst. / Ms. AKSHATHA SWAMY

who has worked for the year ________ in the Computer Laboratory.

[illegible]

Teacher In-Charge

Head of Department

Date : 9/3/20

Examiner

Practical - 01

Objective:- Demonstrate the use of different file accessing mode, different attributes and Record method

Step1 - Create a file, Object using open method and use the write accessing mode followed up by writing some contents onto the file and than closing the file.

Step2 - Now open the file in read mode and than use read(), readline () and readlines () and store the output in variable and finally display the contents of variable.

Step3 - Now use the fileobject for finding the name of the file, the file mode in which it is opened whether the file is still open or close and finally the output of the softspace attribute

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

fileobj = open ("abc.txt", "r") # read mode
#read ()
str1 = file obj.read()
print ("The output of read method :", str1)
fileobj . close ()
>>>("The output of read method : ", "computer science Subjects
      in DBMS in python in DS\n")

 #readline ()
 file obj= open ("abc.txt", "r")
 Str2 = fileobj .readline ()
 print ("The output of readline method:", str2)
 fileobj . close ()
>>>('The output of readline method :', ' Computer Science
        Subject \n')
 #readlines ()
 fileobj = open ("abc.txt" "r")
  str 3 = fileobj. readlines ( )
  Print ("The output of readlines method:", str 3)
  file obj.close ()
>>> ('The output of readlines method :', ['Computer
      Science subjects \n', 'DBMS \n', 'python\n', 'DS\n'])
```

```
#~~SD~~ file attributes
 a = file obj. name
 print ("Name of the file (name attribute): ", a)
>>> (Name of file (name attribute , abc txt)

 b = fileobj.closed
 print ("(close) attribute : ", b)
>>> (close) attribute , True

 c = file obj. mode
 print ("file mode ", c)
>>> ('file mode', 'r')

 d = fileobj. softspace
 print ("softspace, " d)
>>> ('softspace :', d)

# wt mode
fileobj = open ("abc.txt", "wt")
fileobj . write ("akshatha")
fileobject. close ()

# rt mode
fileobj = ~~open~~("abc.txt", "rt")
str1 = ~~fileobj. read~~(r)
print ("output of rt", str1)
fileobject . ~~dose~~ close ()
>>> ('output of rt", akshatha)
```

```
# write mode
 fileobj = open ("abc.t
 fileobj. write ("DBM
 fileobj . close ()

# readmode
 file obj = open ("abc.t
                    "r")
 str2= file obj. read()
 print ("output of read
        mode !", str2)
>>> ("output of read
       mode", 'akshatha")
```

Step 4 - Now open the file object in write mode write some anothere content close subsequently - Then again open the fileobject in 'wt' mode that is the update mode and write content.

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

Step 6 - Now open fileobject in append mode open ~~with~~ write method, write content close the file object in read mode and display the append output.

Step7 - Open the fileobject in readmode. Declare a variable and perform fileobject dot tell method & store the output consequently in variable.

Step8 - Close the seek method either the the arguments with opening the fileobject in read mode & closing subsequently

Step9 - Open fileobject with read mode also use the readlines method & store the output consequently in & print. the same for counting the length use the for conditional statement & display the length.

```
# file attributes
~~a = fileob~~
# append mode
fileobj. open ("abc.txt", "a")
fileobj. write ("Data structure")
file. close ()
file object = open ("abc.txt", "r")
str 3 = file obj. read()
print (" output of append mode :", "Akshatha",
        "Data structures").

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

# seek ()
 file obj = open("abc txt", "r")
 str 4 = fileobj. seek (0,0)
 str 8 = fileobj. read (10)
 print ("The begining of the time is =", str8)
```

```
class odd:
  def---iter---(self):
        self.num = 1
        return self
  def---next---(self):
        if self.num<=10:
            num=self.num
            self.num+=2
            return num
        else:
            raise stop iteration

>>>y=count()
>>>z=iter(y)
>>>z.next()
1
>>>z.next()
3
>>>z.next()
5
```

Practical No-2

Aim - Demonstrate the use of iterable & iterators.

Theory - In python, iterator is an object which implements iterator class which has 2 methods namely ---iter()--- and ---next()---
list, tuple, dictionary & the set all represent a iterable object.

Q1] Write a program using using iterable objects for displaying the odd numbers in range 1 to 10

Algorithm -
Step1 - Define a iter() with argument and initialize the value and return that value.

Step2 - Define the next() with an argument & compare the upper limit by using a conditional statement.

Step3 - Now create an object of the given class & pass the object in the iter method.

2] Write a program using an iterator for calculating the power of a given no. For instance no entered is 2 then value calculated should be 1, 2^1, 2^2, 2^3.

Algorithm
Step 1 - Define iter() with argument & initialize value & return the value.

~~Step 2~~ -

Step2 - Now define next() with an argument and compare the upper limit by using conditional statement.

Step3 - Now create an object of the given class & pass the object in the iter method.

```
class power:
    def __iter__(self):
            self.num = 0
            return self
    def next (self):
        if self.p <= 10:
            num = self p
            self p+ = 1
            po = 2 ** num
            print ("2 **", self p-1, "=", po)
            return po
        else:
            raise stop Iteration.
>>> p=power()
    x = iter (p)
>>> x.next ()
    a = 21
>>> x.next ()
    2 ** 1 = 2
>> >x.next ()
    2 **2 = 4
```

```
#code:
class fact:
    def ---iter---(self):
        self.f = 1
        return self
    def next(self):
        if self.f <= 10:
            num = self.f
            self.f + = 1
            fac = 1
            for i in range(1, num+
                fac = fac * f
            print(self.f - 1, "!=", fac)
        else:
            raise stop iteration

>>> f = fact()
>>> x = iter(f)
>>> x ~~*~~ next()
1! = 1
>>> x.next()
2! = 2
>>> x.next()
3! = 6
```

029

3] Write a program using iterable concept to find factorial of number in range 1 to 10:

Algo:

Step 1 - Define a iter() with argument & initialize the value and ~~value~~ return it.

Step 2 - Define the next () with an argument & compare the upper limit by using a conditional statement.

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

Q4] Write a program using iterable concept to display multiple of 2 in range 1 to 10

Algo -

Step 1 - Define a iter ( ) with argument & initialize the value & return the value.

~~Step 2~~ -

Step2 - Defini the next () with an argument and compare the upper limit by using a conditional statement.

step3 - Now create an object of the given class & pass the object in the iter method.

code.

```
class mult:
    def __iter__(self):
        self.m = 1
        return self.
    def next(self):
        if self.m <= 10:
            num = self.m
            self.m += 1
            table = 2 * num
            print ("2 *", num, "=", table)
        else:
            raise stop iteration

>>> m = mult ()
>>> x = iter (m)
>>> x . next ()
2* 1 = 2
>>> x.next()
 2* 2 = 4
>>> x.next()
2* 3 = 6
>>> x.next()
2*4 = 8
```

```
#code
def accept_age ():
    age=int (input("enter your age:"))
    if age >30 or age < 16 :
        raise value error
    else:
        print ("your age is ", age)
valid= False
while not valid:
  try:
    age =accept_age()
        valid = True
  except value error .
        print ("your age is not in
               range"))

>>> Enter your age: 15
  your age is not in range
>>> enter your age: 32
  your age is not is range.
>>> enter your age : 18
  your age is in ~~age~~ range.
```

Practical No-3

Aim - Demonstrate the use of exceptions on handling

Theory - An exception is an event which occurs during execution of program which disrupt the normal flow of program. Thus a exception represent object which represents an error. This object is derived ~~for~~ from given class & when python script raises an exception it must be handled immediately otherwise it will terminate & close the program.

Q1] Write a program to check the range of the age of the students in given class & if age does not fail in given range use value error exception otherwise return the valid no.

Algorithm.

Step 1 - Define a function which will accept the age of the student from standard i/p.

Step 2 : Use if conditional to check whether the input age value in range & so return the age else use Value error exception.

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

Step 4 - Use except with value error & print the message not a valid range.

Q 2] Write a program to check whether the print no in given class & if the number is a floating point use value error as exception for the given input.

Algorithm.

Step1 - Use try block & accept the input using input () & convert it into integer data-type and subsequently terminate the block.

```
# code.
while true :
    try :
        a = int (input ("Enter a number"))
        print (valid number)
        break
    except (value error:
        print ("Not a valid number!. try again")
```

```
>>> enter number : 1.6 2
Not a valid number.
>>> Enter a number : 16
  Valid Number
```

code

```
def divide ~~dive~~(a,b):
    ans = a/b
    return ans
while true :
    try:
      a= int (input("Enter first no"))
      b= int (input("Enter second no"))
      ans = divide (a,b)
      print ("division of "a", and "b"
              ans")
      break
    except  Zero division error.
      print ("error")
```

```
>>> enter first no: 1
>>> enter second no : 0
~~Division of 1 and 0~~
~~is Error~~
```

m

Step 2 - Use the except block with exception as value error & display appropriate message is suspicious code is part of try block.

Q3] Write a program to demonstrate use of zero division error.

Algorithm

Step 1 - Use the try block & accept the input using input() & then convert it into integer datatype.

Step 2 - Define a function ~~with~~ with 2 programeter to divide the no's given by user.

Step 3 - Define while loop to check whether the boolean expression holds true.

Step 4 - Use except with zero division error & print the message.

km
19/12/17

24/2

Practical - 4

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 re module and common usage of regular expression involves functionalities :-

- Searching a given string
- finding a string
- Breaking a string into smaller substring
- Replacing part of string

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

Algorithm

Step 1 - Now display string & pattern in findall and display the output.

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

```
① import re
string = "hello 1234 abc 567"
result = re.findall("\d+", string)
result2 = re.findall("\D+", string)
print(result)
print (result 2)

# output
>> ['1234', '567']
>> ['hello', 'abc']
```

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

#output
>>> <rematch object : span (0,6).
         match = "python">
>>> match found.
```

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

Algorithm -

Step1 - Import re module and apply a string

Step2 - Use search() with "\A python" and string as two parameters

Step3: Now display the output

Step4: Now use the conditional statement for user to know whether the match is found or not.

Q3] Waite a program on regulator to check whether the given mobile no starts with 8 or 9 the total length of digit should be atmost 10.

Algo

Step1 - Import re module and apply a string of mobile no.8.

Step2 - Now use for conditional statement to find in the number starts with 8 or 9 and the total number should length of 10 else match () inside for statement to find the match is given string.

Step3 - Use if conditional statement to know whether we have a math or not the o/p we have use of group() to display incorrect mobile no.

```
③ Import re
li=["99202 34212", "8412322334",
    "72913 19872", "925312 10"]
for element in li:
  result = re.match ('[8-9] {1} [0-9] {9} ", element)
if result :
   print ("correct mobile no")
   print ( result . group (1))
else :
   print ("incorrect mobileno")

>>> correct mobileno
99202 34212
```

```
(4) import re
    string-"Python is Important"
result~~String~~1 = re.findall ("\w*", string)
result~~String~~2 = re.findall("\w*", string)
    print (result1)
    print (result2)

#output
>>> ['Python', '', 'is', 'important']
    ['Python', 'is', 'important']
```

4) Q) Write a expression for extracting a word from given string along with space character in b/w the word & subsequently extract the word without space character.

Algo

Step1 - Import re module and apply a string

Step2 - Use findall() to extract a word from given string

Step3 - Use "/W*" to extract word along with space & "W~~*~~+" to extract word without space.

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

Algo
Step1 - Import re module & apply a string
Step2 - Use findall () in which use "^\w" as one parameter to find first word of string then use "\w*" as parameter find last word.

Q6] Write a expression for extracting the date in format dd-mm-yyyy by using the findall() where string has following format
Amit 201 24-12-2019

Algo
Step1 - Import re module and apply string
Step2 - Use findall method & use '\d{2}-\d{2}-\d{4}' as an parameter
Step3 - Display output

Code 5
```
import re
string= "python is important"
result= re.findall ("^\w+", string)
result 1 = re.findall ("\w$", string)
print (result)
print (result 1)

>>>['python']
>>>['Important']
```

Code 6
```
import re
string= "Amit 201 24-12-2019"
string= re.findall ('\D{2}-\d{2}-{D{4}',string)
print(result)

>>>['24-12-2019']
```

```
import re
~~resu~~
string = "abc@tcsc.edu"
result = re.findall("^|w+", string)
result1 = re.findall("t|w+.|w+$", string)
result2 = re.findall("t w|.- ", string)
print(result)
print(result1)
print(result2

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

Q7] Write a re for extracting the
① Username from email id
② hostname from email id
③ Both Username & hostname from email id.

Algo
Step1 - Import 're module' & apply a string

Step2 - Use findall() to find username, hostname & both of email id.

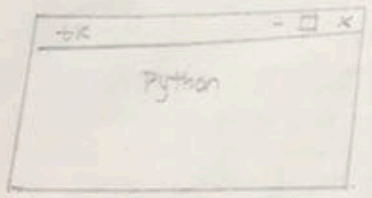
Step3 - Use "^|w+" for username.
Use "t|w+.|w+$" for hostname
Use "[|w|.-]+" for both as parameter in findall()

nm
07/01/22

#Creation of parent window

```
from Tkinter import*
root = TK()
l = Label (root, text = "python")
l. pack()
root mainloop()
```

#Output



#2 label, attributes

```
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", ~~font~~ padx = "20")
l2 = Label (root, text = "CS", bg = "blue", fg= "black" font=
l2 pack (side = LEFT, pady = "30")
l3= Label (root, text = "CS", bg= "yellow", fg = "black",
font = "10")
 l3. pack (side=TOP, ipadx = 40)
```

Practical-5

TOPIC - GUI COMPONENTS

Step1 - Use the tkinter library for importing the features of the text widget.

Step2 - Create an object using the TK()

Step3 - Create a variable using the widget Label and use the text method

Step4 - Use the mainloop() for triggering of the corresponding allow mention events

~~Step5~~ - #2 :- Labels , Attributes

Step1 - Use the tkinter library for importing the features of the text widget

Step2 - Create a variable from the text method and position it on the parent window

Step3 - Use the pack() along with the object created from the text () and use the parameter.

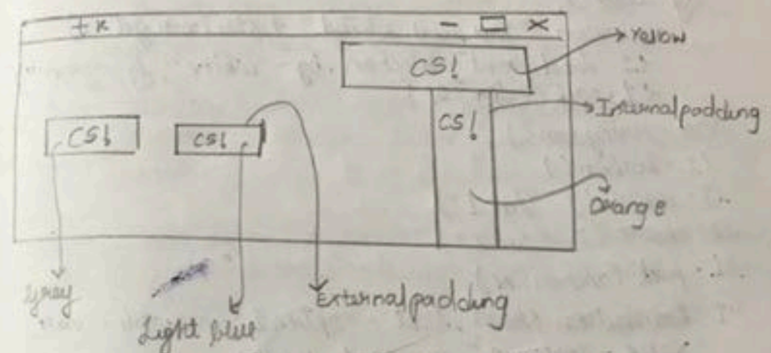
1) side = LEFT, padx = 20
2) side = LEFT, pady = 30
3) side = TOP, ipadx = 40
4) side= TOP, ipady = 50

Step 4 - Use the mainloop () for 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 colour (bg)
4) The foreground colour (fg) & than use the pack () with a relevant padding attributes.

```
l4 = Label (root, text = "CS!", bg = "Orange", fg="black", font="10")
l4 .pack (side-TOP, pady =50)
root mainloop ()
```

output



#Radiobutton

```
from tkinter import *
root = TK()
root.geometry ("500x500")
def select():
    selection = "You just selected" + str(var.get())
    t1 = Label (text = selection, bg = "white", fg = "green")
    t1.pack (side = TOP)
var = string.var ()
l1 = Listbox()
l1.insert (1, "List 1")
l2.insert (2, "List 2")
l1.pack (anchor = N)
r1 = Radiobutton (root, text = "option1", variable = var,
     value = "option 1", command = select)
r1.pack (anchor = N)
r2 = Radiobutton (root, text = "option2", variable = var,
     value = "option 2", command = select)
r2.pack (anchor = N)
root.mainloop ()
```

Step1 - Import the relevant methods from the tkinter library. Create an object with parent window.

Step2 - Use the parent window object along with geometry () declaring specific fixed size of the parent window

Step3 - Now define a function which tells the user allow the given selection ~~are~~ made from multiple option available

Step4 - Now define the parent window and define the option with control variable.

Step5 - Use the listbox() and insert options on the parent window along with the pack () with specifying anchor attribute.

Step6 - create an object from radiobutton which will take following arguments { parent window object, text variable which will take values option no 1, 2, 3 ... variable argument, corresponding value & trigger the function declared.

Step7 - Now call the pack() for radio object so created & specify the argument using anchor attribute.

Step8 - Finally make use of the mainloop() along with parent object.

#2
Step1 - Import relevant method from the tkinter library.

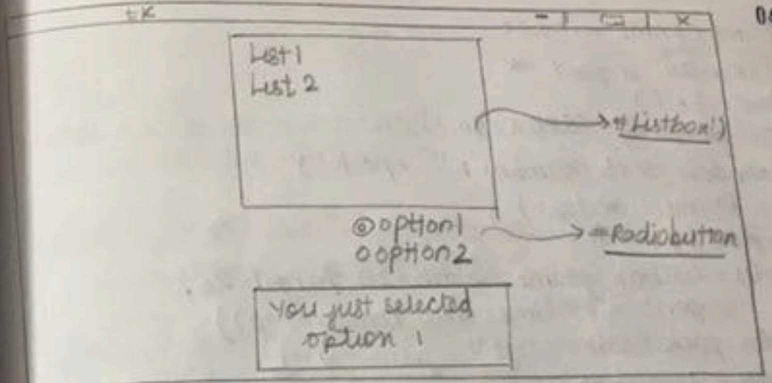
Step2 - Create a parent object corresponding to the parentwindow.

Step3 - Use the geometry() for laying out of the window

Step4 - Create object & use scrollbar()

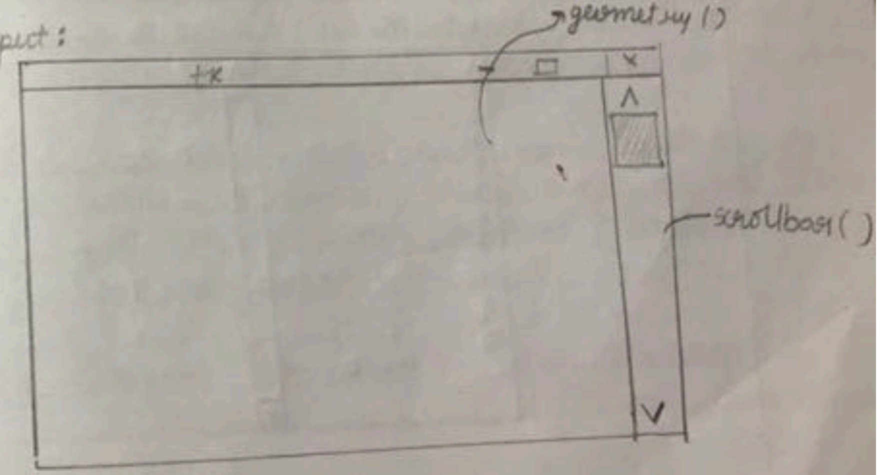
Step5 - Use pack() and along with scrollbar object with side and fill attributes.

Step6 - Use the mainloop with parent object



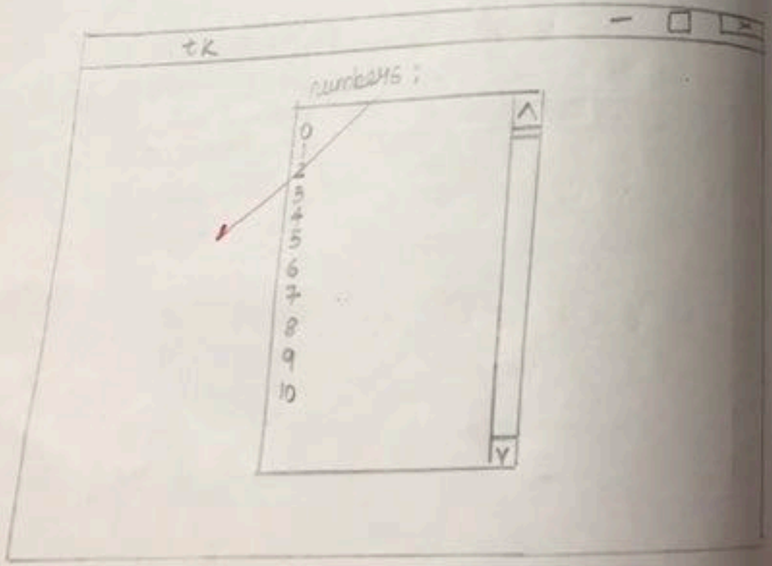
#2
scrollbar()
```
from tkinter import *
root = TK()
root.geometry("500x500")
s=Scrollbar()
s.pack(side="right", fill="y")
root.mainloop()
```

output:



#3 - USING FRAME WIDGET

```
from tkinter import *
window = Tk()
window.geometry("680x500")
Label(window, text="number : ").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)
for x in range(100):
    listnodes.insert(END, str(x))
window.mainloop()
```



#3 :

Step1 - Import the relevant libraries from the tkinter method.

Step2 - Create an corresponding object of the parent window.

Step3 - Use the geometry manager with pixel size (680 X 500) or any other suitable pixel size.

Step4 - Use the label widget along with parent object created & subsequently use the pack method.

Step5 - Use the frame widget along with the parent object created & use pack method.

Step6 - Use the listbox attribute along with the functions like width, height, font to create a listbox method's object. Use pack() for the same.

Step7 - Use the scrollbar() with an object use the attribute of vertical, then configure the same with object created from the scrollbar() and use pack()

Step8 - trigger the events using mainloop.

#4 :

Step1 : Import relevant method from tkinter library

Step2 - Define the object corresponding to the pa window & define it's size of parent windo terms of no of pixels

Step3 - Now define the frame object from the method & place it on the parent window

Step4 - Create another frame object termed as le and put it on the parent window on the left side.

Step5 - Similarily define the RIGHT frame & sub ently define the button object placed onto th given frame with the attribute as text active background & foreground.

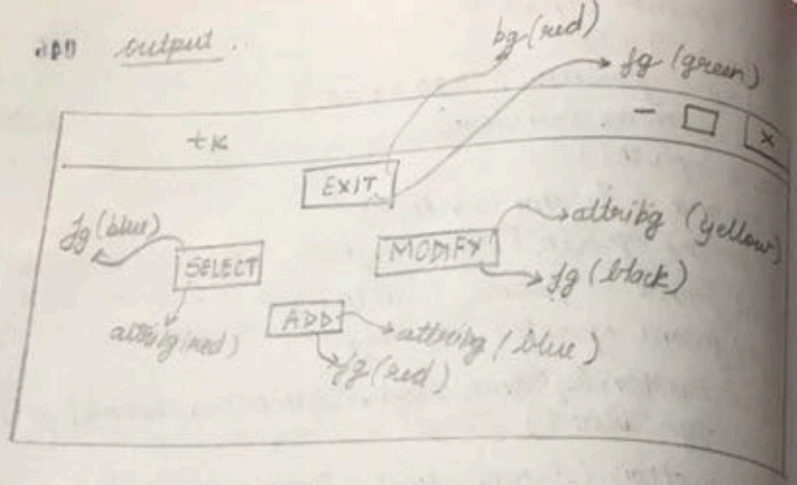
Step6 - Now use the pack() along with the side attribute.

Step7 - Similarily create the button object correspo to the MODIFY operation put it into frame object on side = RIGHT.

t :

```
from tkinter import *
window = TK()
window = geometry ("680 x 500")
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", padn = 20)
b2 . pack (side = "right", padn = 30)
b3 . pack ( side = "bottom", pady = 20)
b4 . pack (side = "TOP")
window. mainloop()
```

output.



Step8 - Create anothere button object & place it on the RIGHT frame & label the button as ADD.

Step9 - Add anothere button & put it on the top of the frame & label it as EXIT

Step10 - Use the back () simultaneously for all the objects & finally use mainloop ()

Step1 - Import the relevant method for tkinter library.

Step2 - Import tkMessageBox.

Step3 - Define a parent window Object along with parent window

Step 4 - Define a function which will use tkMessageBox with show info method along with info window attribute

Step5 - Declare a button with parent's window object along with command attribute

Step6 - Place the button widget onto the parent window & finally call the mainloop () for triggering of events called above.

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

```
#Multiple window
# Different button (Relief ())
from Tkinter import *
root = Tk()
root. minisize (300, 300)
def main():
     top = Tk ()
     top . config (bg ="black")
     top.title ("HOME")
     top.minisize(300, 300)
     L = Label( top, text = "SAN FRANCISO \n Plac
       Interest : \n Golden Gate Bridge \n Lomb
       Street \n chinatown \n coit Tower")
   L.pack ()
  b1 = Button (top, text = "next", command = sec
  b1.pack (side=RIGHT)
 b2 = Button (top, text ="exit," } command = te
 b2.pack (side = Left)
 top.mainloop()
```

Step1 - Import the relevant method from the tkinter library along with parent window object declared.

Step2 - Use parentwindow-object along with minisize function for window size

Step3 - Define a function main, declare parent window object & use config (), title (), minisize(), Label ( ) as use as button () and use pack() & mainloop () simultaneously.

Step4 - similarily define the function second & use the attribute accordingly.

Step5 - Declare another function button along with parent object and declare button with attribute like FLAL, RIDGE, GROOVE, RAISED, SUNKEN along with the relief widget.

Step6 - Finally called the mainloop () for event driven programming

Practical-6

Aim - Demonstrate the use of GUI by creating a human face and converting celsius into fahrenhiet

Q1] Write a program to draw human face using GUI.

Algo

Step1 - Import relevant methods from tkinter library

Step2 - Create an object corresponding to the parent window from TK()

Step3 - Create an object from canvas() & place it onto parent window along with height & width.

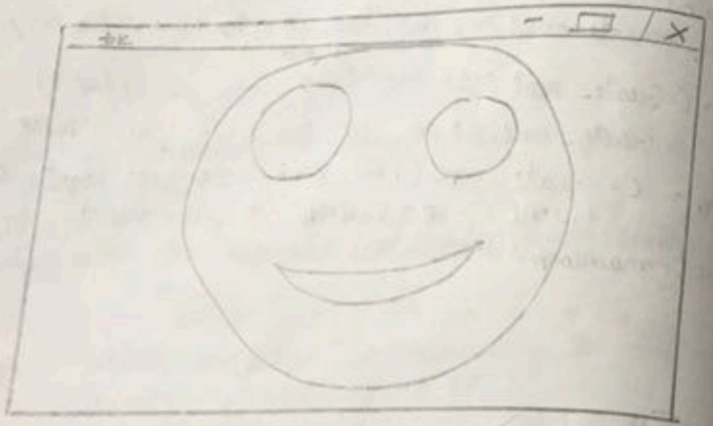
Step4 - Now use pack() for positioning of widget onto the parent window

Step5 - Now create an object face & use object. Create_oval() with co-ordinates 50,50,350 & outline='black', fill="yellow"

code

```
from tkinter import *
root = TK()
c=Canvas(root, width=500, height=500)
c.pack()
face = C.Create_oval(50,50,350,350, outline="black", fill=
                               "yellow")
eye1 = C.Create_oval(125,125,175,175, fill="black")
eye2 = C.Create_oval(225,125,275,175, fill="black")
mouth= C.create_arc(125,275,125,175, start=0,
               extent=-180, width=5, fill="red")
root.mainloop()
```

#output



Step6 - Now create eye1, object & again use object Create_oval () with appropriate co-ordinate ~~use~~ along with fill as attribute to create left eye.

Step7 - Now repeat the same step 6 to create right eye.

Step8 - Create an object mouth & use object Create_arc () with appropriate to co-ordinates start = 0, extent = -180 & fill - "red", width = 5 as attribute to create mouth.

Step9 - Finally use the main loop ().

Q2] Write a program to convert celsius use into farhenheit using GUI

Algorithm

Step1 - Import all the relevant methods in tkinter library.

Step2 - Create object corresponding to the parent window from Tk()

Step3 - Now initalize fahrehneit as Double Var() & it init to 320

Step4 - Now define a function convert with argu celsius, to convert celsius into fahrenheit using .set().

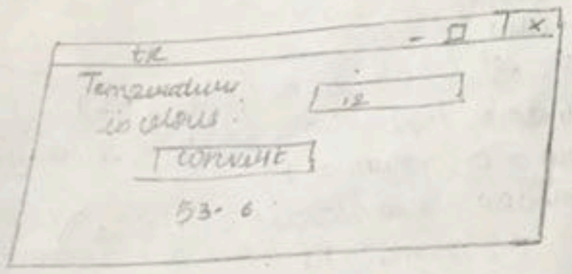
Step5 - Now create an object l2 using Label() & place it onto parent window & use text attribute as enter a no:

Step6 - Now use grid() for position the object onto the parent window

code

```
from tkinter import *
window = tk()
fahrenhiet = Double Var()
fahrenhiet set(320)
def convert (celsius):
        fahrenhiet set ((9.0/5.0)* celsius + 32)
 l1=Label (window, text = "Temperature in celsius:")
 l1. grid (row=0, column=1)
 e = entry (window, text variable = celsius)
 e. grid (row=0, column=1)
 celsius = Int Var()
 l2=Label (window, text variable = fahrenhiet)
 l2. grid (row=2, column=0, column span = 2)
 B = Button (window, text = "calculate", command = lambda:
                       convert (celsius. get()))
 B. grid (row=1, column = 0, columnspan = 2)
window.mainloop()
```

#output



Step6 - Initialize celsius as integers using IntVar()

Step7 - Create another object & use entry widget to enter the input and place it onto the parent window ()

Step8 - Now use grid() for positioning the object onto parent window with text variable attribute.

Step9 - Now again use label() along with textvariabl attribute to display output & use grid() for positioning

Step10 - Finally use mainloop()

Practical - 7

Aim - Write a program to find factorial of number & use arithmetic operations of two numbers using GUI

Q1] Write a program to find factorial of number using GUI

Algo

Step1 - Import relevant methods from tkinter library.

Step2 - Now define a function factorial to calculate factorial using recursive function.

Step3 - Define another function calculate to call factorial function.

Step4 - Now create an object with entry() and use pack() for positioning on parent window.

Step5 - Now create an object with button () along with command attribute to calculate factorial.

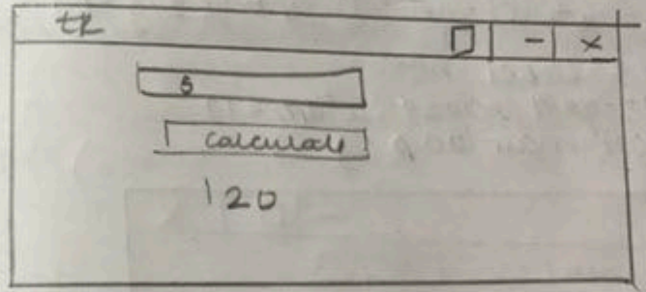
Step6 - Now again create an object with label() to show output.

Step7 - Finally use the mainloop()

code :

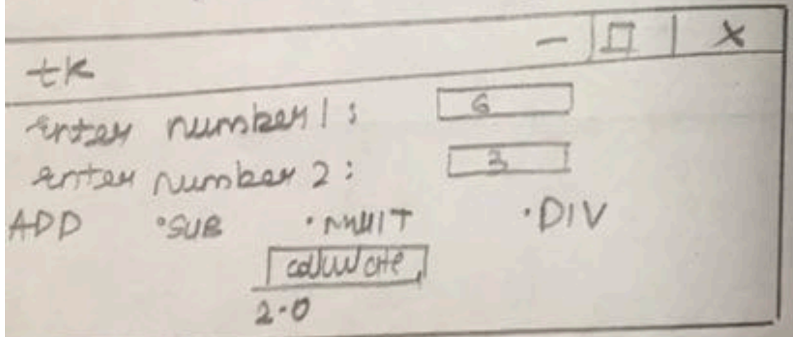
```
from tkinter import *
def factorial (n):
        if n==0 or n==1:
             return 1
        else :
             return n * factorial (n-1)
def calculate ():
     result = factorial(int (entrytext.get ()))
     info.config (text = result)
root = Tk()
entrytest = entry(root)
entrytest. pack()
bth = Button(root, text = "calulate", command=
                  calculate)
bth. pack ()
info = Label (root, text = "factorial")
info.pack ()
root. mainloop ()
```

output



~~#code~~ #code

```
from tkinter import *
def calculate():
    if int(v.get()) == 1:
        res = int(e1.get()) + int (e2.get())
        l3. config(text = res)
    elif int (v.get()) == 2 :
        res = int (e1.get()) - int (e2.get())
        l3. config (text = res)
    elif int (v.get()) == 3:
        res = int (e1.get()) * ~~te~~ int (e2.get()
        l3. config (text= res)
    else :
        res = int (e1.get()) / int (e2.get())
        l3. config (text = res)
        l3 = Label (root)
        l3. grid (row = 4, column = 1)
        root. mainloop ()
~~else:~~
    ~~res = int (e1. get()) / int (e2. get())~~
        l3 = Label (root)
        l3. grid (row = 4, column = 1)
        root mainloop ()
```



tk

enter number 1 : 6

enter number 2 : 3

ADD · SUB · MULT · DIV

calculate

2·0

Q2] Write a program to perform arithmetic operation on 2 numbers using GUI

Algo

Step1 - Import relevant method from tkinter library.

Step2 - Now create an object corresponding to parent window.

Step3 - Now define a function calculate to carry out arithmetic operations on 2 numbers.

Step4 - Now create object with label() as num1 & num 2 and use grid() to place it onto parent window.

Step5 - Create objects with entry() to take input from user()

Step6 - Now initialize V as integer using IntVar()

Step 7 - Now create 4 objects with RadioButton to choose any one of arithmetic operators and use grid() for positioning onto parent window.

Step 8 - Now create a object with button () along command attribute to carry out arithmetic operation of user's choice

Step 9 - Now create a object with label() to show output.

Step 10 - Finally use the mainloop().

```
root = TK()
L1 = Label (root, text = "Enter a no:")
L1 . grid (row= 0 Column = 0)
e1 = entry (root)
e1 . grid (row = 0, column = 1)
L2 = Label ( row = 0, column = 1)
L2 grid( row = 1 , column = 0)
e2 = entry ( root )
e2 . grid (row = 1, column = 1)
  r = IntVar ()
  r1 = RadioButton (root, text = "add", variable = v,
                    value = 1)
  r1 . grid (row=2 , column = 0)
  r2 = RadioButton ( root  text = "~~Multi~~ sub", Variable = v
                    value = 2)
  r2 . grid (row = 2 , column = 1)
  r3 = RadioButton (root, text = "Mult", Variable = V,
                    value = 3)
  ~~r3 = RadioButton (root, text =~~
       r3 . grid (row = 2, column = 3)
B = Button (root, text = "Calculate", Command =
                    calculate )
B . grid (row = 3, column = 1, column span = 2)
```

```
while true:
        data= conn. recv (1024) decode ()
        if not data:
            break
        print(" from connected user :" +str(data))
    ↳ data = input (" → ")
    ↳ conn. send (data.encode ())
    ↳ conn. close ()
```

(Now run the program and right ~~white~~ write client program)

#code

```
   import socket
  def client_program ():
       host =socket. gethostname ()
      port= 5000
 ← client socket = socket. socket ()
client_socket • connect ((host, port))
message=input (' → ")
while  message • lower () • strip ()!= 'bye'
                client- socket. send (message. encod
          data= client _socket . recv (1024) d
print ("Received from server:" + data)
```

Practical-8

Aim - To demonstrate the use of socket module and server client programs.

Write a program to demonstrate use of socket module & server client program

Algorithm:

Step1 - Import the socket module to import relevant methods.

Step2 - Define a function as server-program to get hostname

Step3 - Now get value for port variable to initialize portno above 1024.

Step4 - Use socket () to get instance.

Step5 - Now use bind () function to bind host address and port together to configure how many client the server can list simultaneously.

Step6 - Now use accept ( ) to accept new connecti

Step7 - Now print address.

Algo

Step1 - Import socket ~~meth~~ module to import methods that are relevant

Step2 - Define a function client-program get the hostname and give port a value Soc

Step3 - Now again initiate by using socket.socket().

Step4 - Use connect() to connect the serve

Step5 - Now take the input ("→")

Step6 - Use while conditional loop to send a message.

Step7 : Now use decode to receive response

step8 - Now show the data.

Step9 - Again take input

tep10 - Close the program by using ( )

```
message = input ("→")
client_socket close ( )
```

output for socket program

```
$ python3.4 socket server.py
connection from: ('127.0.0.1', 578822)
from connected user : Hi
↳ Hello
from connected user: Awesome!
↳ Ok then, bye!
```

output for client_program

```
$ python 3.6 socket_client.py
 ↳ Hi
Received from server: Hello
 ↳ How are you?
Received from server: Good
 ↳ Awesome!
Received from server: Ok then bye!
 ↳ Bye.
```

CODE IN SHELL ENVIRONMENT

```
>>> import sqlite 3
>>> conn = sqlite 3 ; connect ("Student 1 . db
>>> cur = conn . cursor ()
>>> cur. execute ('create table student
            (rollno int (5) primary key
            name varchar (50) not null, add
            varchar (50) not null, class varcha
            dob date) ')
<sqlite. cursor object at 0X0322EBED>
>>> cur.execute ("insert into student values
            ('101',"Akshatha", "MiraRoad", "FYCS",
            "15-10-2001")
<sqlite3. cursor object at 0X0322EB.ED>
>>> cur. execute ('insert into student values
            ('102', 'Alentina", "Borivali", "FYCS
            "20-05-2001")
<sqlite 3. cursor object at 0XB322EBED>
>>> cur. execute ("Select * from student")
<sqlite3. cursor object at 0X0322EBED>
>>> cur.fetchall
[('101','Akshatha', 'MiraRoad', 'FYCS' '15-01-20
('102','Alentina', 'Borivali', 'FYCS', 20-05-
>>> cur.close ()
```

Practical-9

Aim - Demonstrate the use of database connectivity

Algorithm

step1 - Import sqlite 3 module to import relevant methods.

step2 - Now initialise a variable conn to collect by using collect () ~~to~~ to new database using extension . db

step3 - Now initialize a variable to connect to cursor()

step4 - Now use cur. execute () to create a table, insert values into table & use DML, DDL statements to mainuplate data.

Step5 - Use fetchall () to show o/p.

Step6 - use commit () to save all changes

Step7 - use close to. ~~torimate~~ terminate the program.