



Laxmi Charitable Trust's
**Sheth L.U.J. & Sir M.V. College of
Arts, Science & Commerce**

Dr. S Radhakrishnan Marg, Andheri (E), Mumbai - 400 069.

Certificate

*This is to certify that, Mr./Ms. Neeraj Appasi
Seat No. F129 studying in F.Y.B.Sc. SEM-II Computer
Science has satisfactorily completed the Practicals in the
Subject of Python as prescribed by University of
Mumbai, during academic year 2019-2020.*

Signature
Subject in charge
Date: -

Signature
Co-ordinator B.Sc. C.S
Date: -

Signature
External Examiner
Date: -



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

PLNR AD

Date	Aim	Sign
6/01/19	Python file methods and file mode	J 1/1/19
6/01/19	Tkinter widgets GUI programming	J 9/1/19
16/01/19	Connect SQLite to python	J 16/1/19
29/01/19	Re-Regular Expression operators	J 31/1/20
18/01/19	Network Handling	
27/02/19	Exception Handling	



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ACNRA0

Practical No 1

Aim: Python file methods and file modes

Sr No	Method	Description
1	close()	Closes the file
2	detach()	Returns the separated raw stream from buffer
3	fileno()	Returns a number that represents the stream from the operating system
4	flush()	Flushes the internal buffer
5	isatty()	Returns whether the file stream is interactive or not
6	read()	Returns the file content
7	readable()	Returns whether file stream can be read or not
8	readline()	Returns one line from file
9	readlines()	Returns a list of lines from file
10	seek()	Change the file position
11	seekable()	Returns whether file allows us to change the file position
12	tell()	Returns the current file position
13	truncate()	Resizes the file to specified size
14	writable()	Returns whether file can be written or not
15	write(s)	Write the specified string to file
16	writelines()	Write a list of strings to file



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

Algo. Dr.

file modes:

No	Mode	Description
1	'w'	Opens a file for writing. Truncating the file first.
2	'r'	Opens a file for reading.
3	'x'	Creates a new file. If the file already exists, the operation fails.
4	'a'	Opens file in append mode. If file doesn't exist, creates a new file.
5	't'	Opens file in text mode.
6	'b'	Opens file in binary mode.
7	'+'	Opens file for reading and writing.



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

FLNFAU

```
import os as c  
def PI():
```

Write or Create a file

```
Name = input ("Enter your name:")
```

```
No = input ("Enter your no:")
```

```
File1= Name+No+".txt"
```

```
with open (File1,'w') as k:
```

```
k.write (Name)
```

Read file

~~File2= open (File1,'r')~~~~print (File2.read())~~

append file

With open (abFile1,'a') as k2:

k2.writelines ("Password")

print ("file appended").

delete file

```
file3= input ("Enter file name:")
```

```
file3+ ".txt"
```

```
if c.path.exists (file3):
```

```
c.remove (file3)
```

```
print ("file name deleted")
```

Output

Enter your Name : NU

Enter your NO : 123

NU123

file appended

Enter file Name : NU123

NU123 deleted



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

DLMR file

Practical No 02.

Aim- GUI Programming

Sr No	Widget	Description	Syntax
1	Window	All the widgets are placed in window	win = window(name = Tk())
2	Label	A label is a text used to display some message or information about widgets	win = Label(master, options)
3	Entry	The entry widget is used to display the single line text field to the user. It is commonly used to accept user values.	win = entry(parent, options)
4	Button	The button is used to add various kind of buttons to applications	win = Button(parent, options)
5	Scale	It is used to provide the slider to the user	win = Scale(top, options)
6	Checkbutton	The checkbutton is used to display the checkbutton on the window	win = checkbutton(master, options)



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

A(HRPO)

- 1) Radiobutton The radiobutton provides user various options and the user can select only one option among them $w = \text{Radiobutton}(n, \text{options})$
- 2) Spinbox It is an entry widget used to select from option on value $w = \text{Spinbox}(n, \text{options})$
- 3) Messagebox This module is used to display message box function. name the message-box in the desktop (title, message [options]) based applications
- 4) Scrollbar It provides the scrollbar to the user so that the user can scroll the window up and down $w = \text{Scrollbar}(n, \text{options})$
- 5) Menu It is used to add menu items to the user $w = \text{Menu}(n, \text{options})$
- 6) Menubutton The menubutton is used to display the menu items to the user $w = \text{Menubutton}(n, \text{options})$
- 7) Text It is used to write and edit text $w = \text{Text}(n, \text{options})$
- 8) List box It is used to display list or options to user $w = \text{listbox}(n, \text{options})$
- 9) Message The message widget is used to display messagebox to user $w = \text{message}(\text{parent}, \text{options})$



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

PLNRT

Practical - 2

window
label widget

from tkinter import *

m=Tk()

n=Tk()

m.title ("Window Loop")

label widget

k=Label (n,text = "Label design")

k.pack

check button

u=Checkbutton (n, text = "Checkbutton")

u.pack()

entry

from t=Entry (m)

from l grid (row=0, column=5)



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

Button widget

from tkinter import *

k = Tk()

k.title ("Window")

k.geometry ("200 x100")

b = button (k, text = "Button")

b.pack()



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

PLTW

Entry widget.

from Tkinter import *

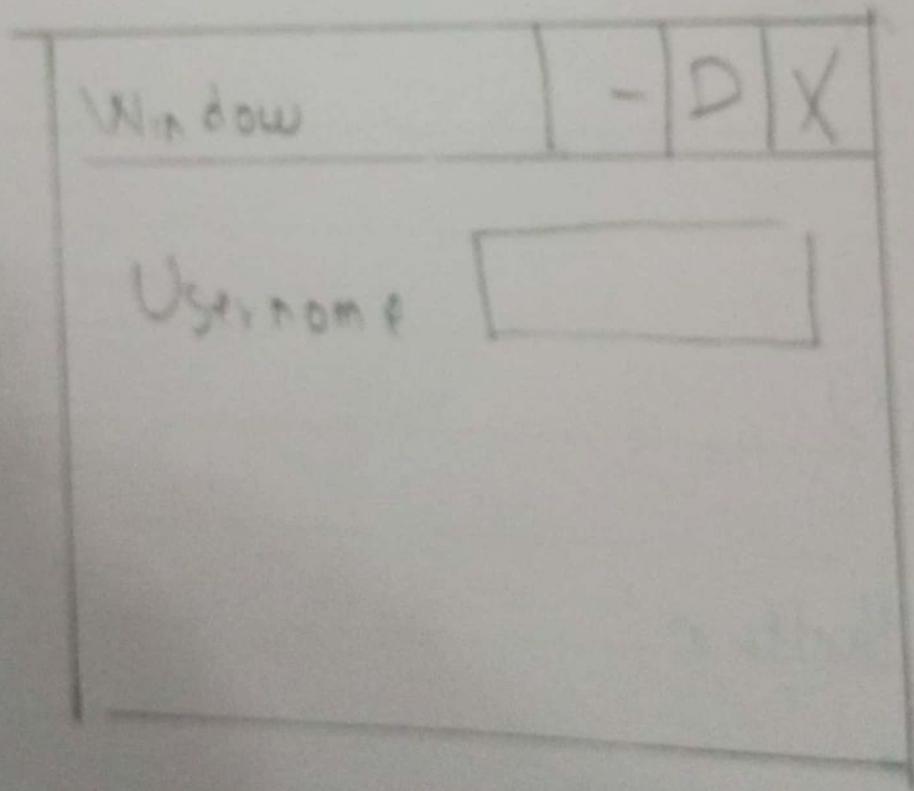
p = Tk()

p.geometry ("500x300")

p.title ("Window")

en = Label (p, text = "username").place (x = 35, y = 70)

ent = Entry (p).place (x = 35, y = 70)





**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

P2L1108Ae

Scale widget

From tkinter import *

l2 = Tk()

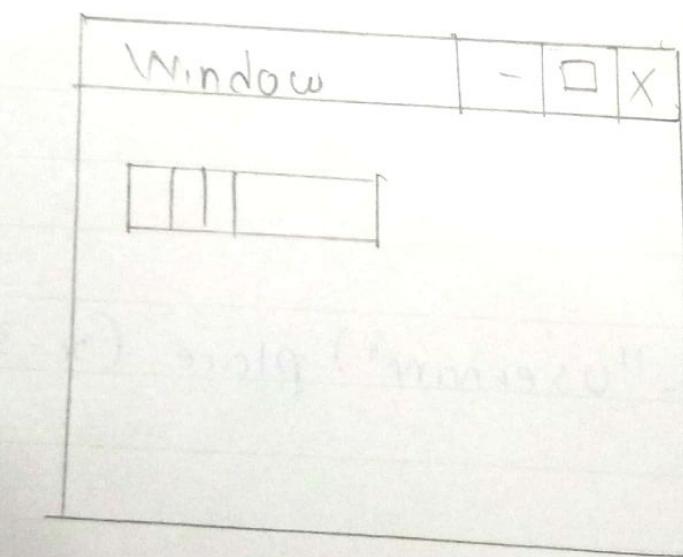
k = geometry ("200x100")

k.title ("Window")

v = DoubleVar()

sc1 = Scale (k, variable=v, from=1 to 50, orient)

sc1.pack (conchan=CENTER)





**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

PLMS

Scale widget

From tkinter import *

l = Tk()

l. geometry ("200x100")

l. title ("Window")

v = DoubleVar()

sc1 = Scale (l, variable=v, from=1 to 50, orient)

sc1. pack (conchar.CENTER)



SIR M.V. COLLEGE OF SCIENCE & COMMERCE

Department of Computer Science

DLNRAC

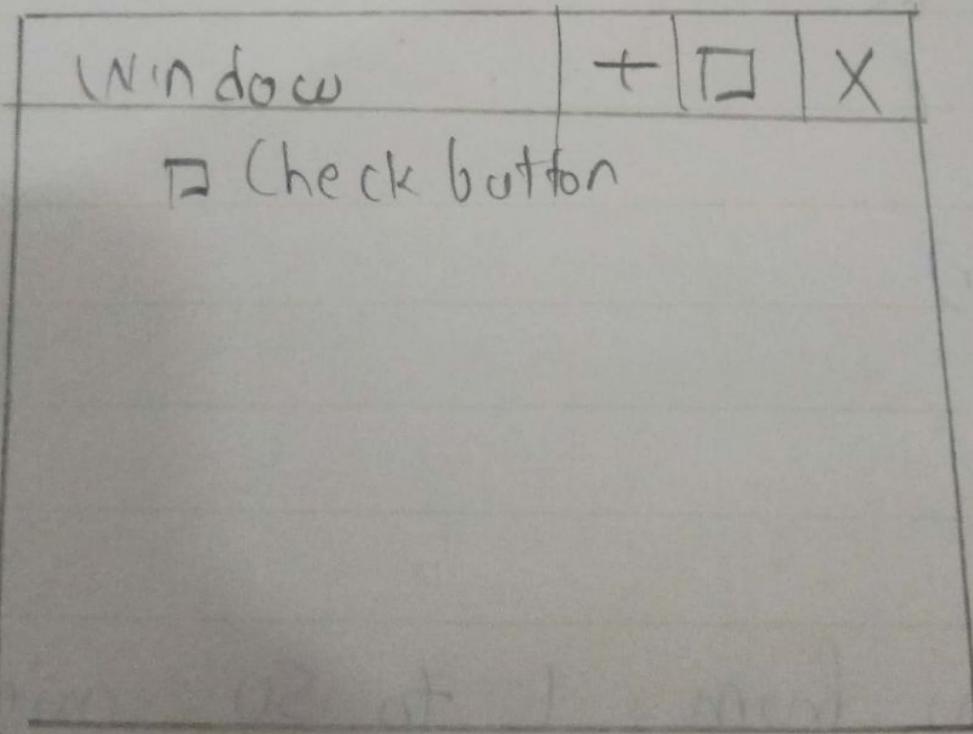
Checkbutton widget

from tkinter import *

L = Tk()

L.title("Window")

u = Checkbutton(L, text="Checkbutton")
u.pack()





**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

RadioButton Widget

from tkinter import *

n = Tk()

n.title('Window')

n.geometry("200x200")

b = RadioButton(n, text = "RadioButton")

b.pack()

Window

I - X

① Radio button



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science PLPRAC

Spinbox Widget

from tkinter import *

k=Tk()

k.title ("Window")

k.geometry ("200x200")

spin= Spinbox(k from = 0 to 2)

spin.pack()

Window	-	D	X
20		D	



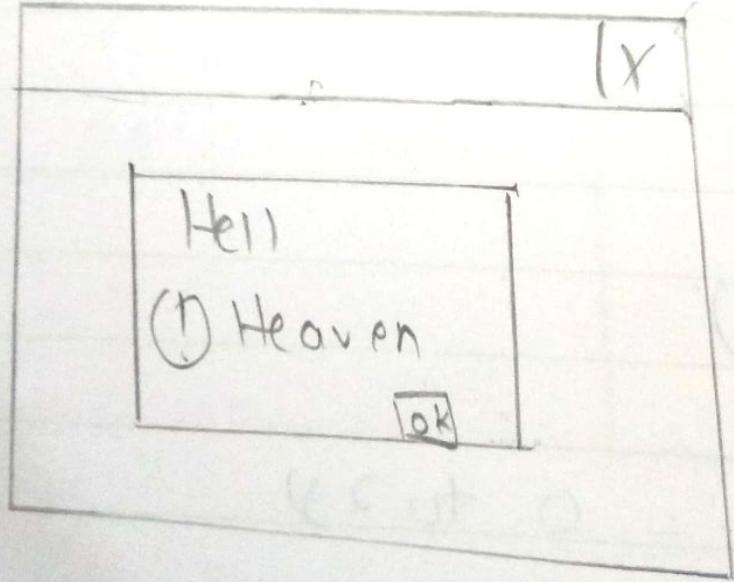
**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

MessageBox widget

From tkinter import *

```
n = Tk()  
n.title("Window")  
n.geometry("100x100")
```

```
messagebox.showinfo("hell", "Heaven")  
n.pack()  
n.mainloop()
```





**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

ALNFAO

Scrollbar widget

from tkinter import *

n=7k()

sb=Scrollbar(h)

sb.pack(side=Right, fill=Y)

h.title("Window")

list=listbox(top, yscrollcommand=)

for line in range 30(30)

list.insert(END, 'Number' str(line))

list.pack(side=left)

sb.config(command=mylist.yview)

Windows

Number 11

Number 12

Number 13





**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

ANIRUPA

Menu widget

```
from Tkinter import *
```

```
n = Tk()  
n.title("Window")
```

```
def ex1():  
    print("Ex1")
```

```
menubar = menu(n)
```

```
menubar.add_command(label="Ex", command=hello)
```

```
menubar.add_command(label="Ex^2", command=topn)
```

```
top.config(menu=menubar)  
n.mainloop()
```

\Nlindow	\ D\ A
Hello\n\t	



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

PLAIRAO

Text widget

from Tkinter import *

k = Tk()

text = Text(k)

text.insert(INSERT, 'NAME')

text.insert(END, 'Phone no:')

text.title("Window")

text.pack()

window

- D X

Name... phone no. ...



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

Algo

Message widget

from tkinter import *

k=Tk()

k.title("Window")

k.geometry("100 x 100")

var=StringVar()

msg=Message(k, text="Hi")

msg.pack()

k.mainloop()

Window

I - X

H₁

Canada

Commodity

(Window) -

(Window) -



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ACNIP

Radiobutton

```
from tkinter import *
n=Tk()
```

```
n.title('Window')
```

```
n.geometry("200x200")
```

```
o= Radiobutton(n, text="Radiobutton")
```

```
o.pack()
```

```
.
```

MessageBox

```
from tkinter import *
```

```
n=Tk()
```

```
n.title('Window')
```

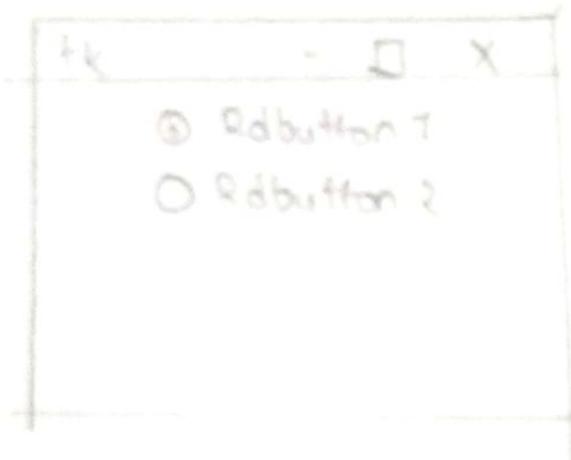
```
n.geometry("200x200")
```

```
messagebox.showinfo ("Good", "Bad")
```

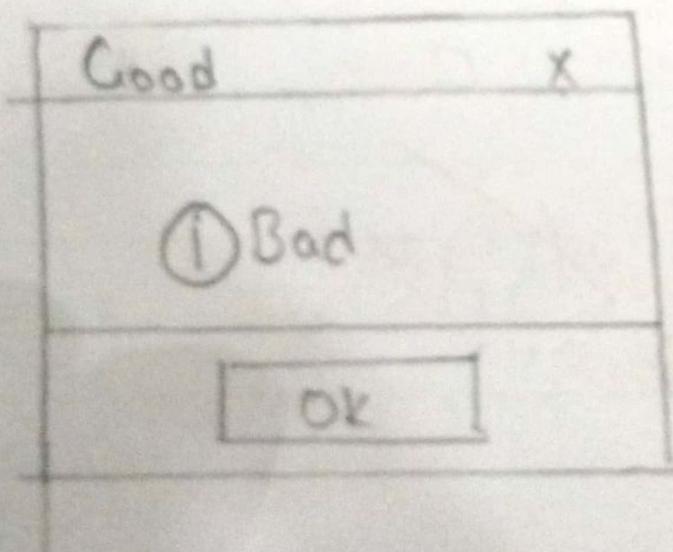
```
n.mainloop()
```

9/11/20

• Radio button



• Message box





**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

ALMRM

H button widget

```
from tkinter import *
```

```
k = Tk()
```

```
k.title("Roll")
```

```
k.geometry("200x100")
```

```
b = Button(k, text="Button")
```

```
b.pack()
```

```
k.mainloop()
```

Spinbox

H Scrollbar widget

```
from tkinter import *
```

```
k = Tk()
```

```
k.title("Roll")
```

```
k.geometry("200x200")
```

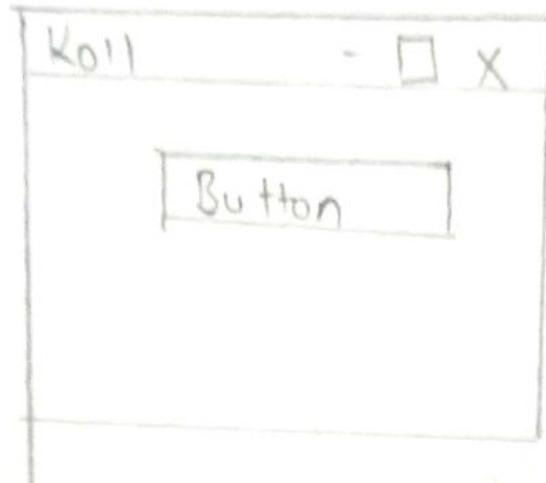
```
spin = Spinbox(k, from_=0, to=25)
```

```
spin.pack()
```

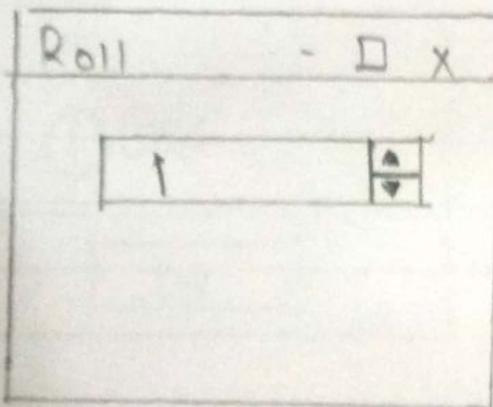
```
k.mainloop()
```

Roll to create a button
button = Button(roll, position=100, width=100,
height=50, text="Button")

H Button



Spinbox





Practical - 3

Aim : Connect Sqlite3 to Python

S.No	Methods	Description
1	cursor	It will create a cursor which accepts a single optional parameter
2	execute	It creates an intermediate cursor object by calling the cursor methods then calls execute method without
3.	executemany	It executes on sqls command against all parameter sequence)
4.	executescript	It executes multiple Sql statements at once provided in the form
5.	commit	It is used to save the changes in the database
6	Totalchange	It returns the total number of database rows that have been inserted or modified
7	Roll back	It rolls back anychanges to the database since the last call to commit
8	close	It closes the database connection
9)	fetchone	It fetches the next row of a query set, returning a single sequence
10)	fetchall	It returns all the data in the database in the form of list.



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

ALNRAO

P-3

a) Creating a table

from sqlite import *

```
c = sqlite3.connect('employee1.db')
print("employee db database created.")
```

C → S

```
cur = c.cursor()
```

```
cur.execute("Create Table emp (NAME TEXT, AGE INT);")
```

```
print("Table created successfully").
```

ALNRAO

Output

Employee database created

Table created successfully



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

ALNFAO

b) Insert record.

c.execute

import sqlite3 as k

p = k.connect('stud.db')

c = p.cursor()

c.execute("create table stud (subject, marks)")

c.execute("insert into stud values ('Science', 80)")

c.execute("insert into stud values ('Maths', 90)")

c.execute("insert into stud values ('History', 100)")

print ("record inserted")

c.commit

c.close()

89

Record inserted by

giant a

* today will

(db trapping) (mico-
tac) (radiotag) (spine)



**SHEETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

H) Select from table.

import sqlite3 as k

p=k.connect('stud.db')
c=p.cursor()

~~c.execute~~
x=c.execute('select * from stud')
for x in c:
print(x)

Subject	Marks
Science	80
Maths	90
History	100



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

ALNIRAO

⑧ Update table

import sqlite3

c = sqlite3.connect('emp.db')

cuv = c.execute

cuv . execute ("UPDATE emp set age=30, where NAME = No")

c . commit

print("update successful")

c.close()

Update Successfull



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

ALNRAO

c3) Append record

```
import sqlite3  
c = sqlite3.connect('emp.db')
```

```
cur1 = c.cursor()  
cur1.execute("Update emp set NAME = concat(NAME, 'Rahul')")  
print("Appended record")  
cur1.commit()  
c.close()
```

DEPARTMENT OF COMPUTER SCIENCE & COMMERCIAL COMPUTER APPLICATIONS

DEM RTO

Notable

Est. 1982

(DB.QMS) Branch Est. 1982

Scanned by CamScanner



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

ALNRAO

deDelete record

import SQLite3 as k

p=k.connect('stud.db')

p c=k.cursor()

c.execute("Delete DELETE from stud where marks=80")

pt

c.commit

c.close()



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

g) Drop table.

import sqlite3 as k

p = k.connect("stud.db")

c = p.cursor()

c.execute(c'drop table stud')

c.close()



Practical - 4

Aim: Regular expression operations

Special characters

Characters	Meaning
.	In default mode, matches any character except newline if dotall flag is specified it matches even new line
^	Matches the start of the string, and in MULTILINE mode also matches immediately after newline
\$	Matches the end of the string or just before the newline at the end of the string and in MULTILINE mode also, before newline
*	Causes the resulting RE to match 0 or more repetitions preceding RE
?	Causes the resulting RE to match 0 or one repetitions preceding RE
?	Causes the resulting RE to match 0 or 1 repetitions preceding RE
?, +?, ??	The '', '+', '?' qualifiers are greedy, the match as much text as possible Adding '?' makes it perform the match in non greedy or minimal fashion
{m}	Specifies that exactly m copies of the previous RE should be matched
{m,n}	Causes the resulting RE to match from m to n repetitions of the preceding RE

**SHEETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science **B.Tech**

(m.n)

Causes the resulting RE to match on the repetitions of the preceding RE, attempting to match as many repetitions as possible.
either: Braces special characters or signals
a special sequence

(.)

Used to indicate a set of characters to a set
• Characters can be listed individually
• Ranges of characters can be indicated by
grouping two characters and separating them
• Special characters like \$, ., ^, \$, \n, \t, etc
• Character classes such as [a-z] or [A-Z]
also accepted if set
- To P1B, where A and B can be arbitrary REs
creates a regular expression that will match
either A or B

()

Matches the regular expression is inside the parentheses
and indicates the start and end of the group
The first character after the ? determines the
meaning of further syntax.

? (Locus)

re A(?B) - only matching, re I (ignore case)
re I (case dependent) re m (multi-line), it's dot matches
and .x.v (univoc matching) are re.X(babev)

(?P-name)

A backreference to a named group if earlier
whatever text was matched by leftmost
group named corr.

? b)

? (comment): the content of parenthesis ignored
Matches if ... matches next but doesn't continue it

?)

Matches if doesn't match next

? ()

Matches if the current position in string is preceded
by match for



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALNRAO

(?cl...)	Matches if the current position in the string is not preceded by a match for ...
(?Cid/nore)	Will try to match with yes-pattern if the group with given id or nore, pos exists, and with no pattern if it doesn't
\number	Matches the contents of group of some number
\A	Matches only at the start of the string
\b	Matches the empty string but only at the beginning or end of the word
\B	Matches the empty string, but only when it is not at the beginning or end of a word
\d	For unicode patterns Matches any unicode decimal digit
\D	for 8-bit patterns Matches any decimal digit.
\s	Matches any character which is not a decimal digit
\S	For unicode patterns Matches unicode whitespace characters and also many other characters
\w	for 8-bit patterns Matches characters (considered whitespace) in ASCII characters
	Matches any characters which is not a whitespace characters
	For unicode patterns Matches unicode word characters
	for 8-bit patterns Matches characters considered Alphanumeric in ASCII character set



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

Affan N

\n Matches any character which is not a word
2 Matches only at the end of the string



Module	Module Contents Description
1 re.compile (pattern, flags=0)	Compile a regular expression pattern into regular expression object.
2 re.P	
3 re.ASCII	Make \w, \W, \b, \B, \d, \D, \s and \S perform ASCII-only matching instead of full Unicode matching
4 re.DEBUG	Display debug information about compiled expression
5 re.T	
6 re.IGNORECASE	Perform case-insensitive matching; expression like [A-Z] will also match lowercase letters
7 re.L	
8 re.LOCAL	Make \w, \W, \b, \B and case-insensitive matching dependant on the current locale
9 re.M MULTILINE	When specified, the pattern character '^' matches at the beginning of the string and at the beginning of each line
10 re.S	
11 re.DOTALL	Makes the '.' special character match any character at all, including a newline;
12 re.X	
13 re.VERBOSE	This flag allows to write regular expressions look nicer and more readable by allowing it to visually separate logical sections of patterns and their components.



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALNAPAO

15 re.search
(pattern, strings,
flags=0)

16 re.match
(pattern, strings,
flags=0)

17 re.fullmatch
(pattern, strings,
flags=0)

18 re.split
(pattern, string,
maxsplit=0, flags=0)

19 re.findall(pattern,
strings, flags=0)

20 re.finditer
(pattern, strings,
flags=0)

21 re.sub(pattern,
repl, string, count=0
flags=0)

22 re.subn
(pattern, flags=0)

23 re.escape
(pattern)

Scan through string looking for the first location where regular expression pattern produce a match, and return a corresponding match object.

If zero or more characters at the beginning of string match the regular expression pattern return a corresponding match object.
If the whole string matches a regular expression pattern, return a corresponding match object.

Split string by the occurrence of pattern.
If capturing parenthesis are used in pattern, then the text of all groups in the pattern are also returned as part of the resulting list.

Return all non-overlapping matches of pattern in string as list of string.

Return an iterator yielding match objects over all non-overlapping matches for the RE pattern in string.

Return the string obtained by replacing the leftmost non-overlapping occurrences of pattern in string by the replacement repl.

Perform the same operation as sub(), but return a tuple.

Escape special characters in pattern. This is useful if you want to match an arbitrary literal string that may have regular expression metacharacters in it. Use the regular expression cache.



Regular expression objects

pattern-search (string[, pos[, endpos]])

Objects	Description
1 pattern-search (string[, pos[, endpos]])	Scan through string looking for the first location where regular expression produces a match
2 pattern-match (string[, pos[, endpos]])	If zero or more characters at the beginning of string match this regular expression return a corresponding match object
3 pattern-fullmatch (string[, pos[, endpos]])	If the whole string matches this regular expression, return a corresponding match object.
4 pattern-split (string[, maxsplit=0])	Identical to split() function, using the compiled pattern. accepts pos and endpos
5 pattern-findall (string[, pos[, endpos]])	Similar to findall() function, using the compiled pattern, accepts pos and endpos
6 pattern-finditer (string[, pos[, endpos]])	Similar to the finditer() function, using the compiled pattern accepts pos and endpos
7 pattern-sub (rept, count)	Identical to sub(), using the compiled pattern
8 pattern-subn(repl, count=0)	Identical to subn(), using the compiled pattern
9 pattern-flags	The regex matching flags. This is a combination of flags given to compile
10 pattern-groups	The number of capturing groups in pattern
11 pattern-group(index)	A dictionary mapping any symbolic group names defined by (?P<group>) to group number
12 Pattern-pattern	The pattern string from which



Match objects

No	Objects	Descriptions
1	Match.expand(template)	Return the string obtained by doing backslash substitution in template as done by sub() method
2	Match.group(group)	Returns one or more subgroups of the match
3	Match.__getitem__(g)	This is identical to m.group(g) - This allows easier access to an individual group from a match
4	Match.groups(default=None)	Return a tuple containing all the subgroups of the match from 1 to however many groups are in the pattern
5	Match.groupdict(default=None)	Return a dictionary containing all the named subgroups of the match, keyed by subgroup name
6	Match.start([group])	
7	Match.end([group])	Return the indices of the start and end of substring matched by group group defaults to zero
8	Match.span([group])	For a match m, return the 2-tuple (m.start(group), m.end(group))
9	Match.pos	The value of pos which was passed to search() or match() method of regeon object
10	Match.endpos	The value of endpos was passed to search() or match() method of regeon object
11	Match.lastindex	The integer index of the last matched capturing group, or None if no group was matched at all
12	Match.lastgroup	The name of the last matched capturing group, None if the group didn't have a name.
13	match.re	The regular expression object whose match or search()
14	match.string	The string passed to match() or search()



```
import re
```

```
#
```

```
string = "Big 26 black 56 bug bit a big black bear"
```

```
print(string, "\n")
```

```
#findall
```

```
var = re.findall("black", string)
```

```
print(var)
```

```
#split
```

```
var = re.split("\s", string)
```

```
print(var)
```

```
#sub
```

```
var = re.sub("black", "brown", string)
```

```
print(var)
```

```
# Search
```

```
var = re.search('black', string)
```

```
print(var)
```

ALOIDAO

Output:

Big 26 black 56 bug bit ' a big black bee

{'black', 'black'}

L'Big', '26', 'black', '56', 'bug', 'bit'; a, 'big', 'black')

Big 26 brown 56 bug bit a big brown
block



**SHEETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science

PINPAC

meta characters

import re

txt = The 34 mouse is in the house"

x = re.findall ("(a-m)", txt)
print (x)

x = re.findall ("l d", txt)
print (x)

x = re.findall ("mo. - e", txt)
print (x)

x = re.findall ("nT he", txt)
print (x)

PLN RAO

Output :

('t', 'e', 'm', 'i', 'l', 'h', 'e', 'h', 'e']

[3, w])

(mouse))

('The')



**SHETH L.U.J. COLLEGE OF ARTS &
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**
Department of Computer Science ALNIRAO

special Sequences

import re

txt = "The 3H mouse is in the house"

```
x=re.findall(r"\AThe",txt)  
print(x)
```

```
x=re.findall(r"\bmo\b",txt)  
print(x)
```

```
x=re.findall(r"\selb",txt)  
print(x)
```

```
x=re.findall(r"\Bse",txt)  
print(x)
```

ALNRAo

Output

['The']

['mo']

['se', 'se']

{'se', 'Se'}



Sets

import re

txt = "The 34 mouse is in the house"

```
x = re.findall("[a-n]", txt)  
print(x)
```

```
x = re.findall('[a-n]', txt)  
print(x)
```

```
x = re.findall('^\^a[n]', txt)  
print(x)
```

```
x = re.findall("0123", txt)  
print(x)
```

~~```
x = re.findall("fog", txt)
print(x)
```~~

ALP1RPO

Output

[l' u']

[l'b' e l'm' e l'n' n' e' z h k' e'] HE sdT " d

{l', b', e', m', n', e', z, h, k', s'} (1) fad

( )

(rst, 'ond/\*x) uobnd

{b' e'}

(rst, 'd / e' x) uobnd

(d)



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**  
Department of Computer Science

A.C.N.RAO

$x = \text{re.findall} ("(0-9)", \text{txt})$   
 $\text{print}(x)$

$x = \text{re.findall} ("[0-5][0-9]", \text{txt})$   
 $\text{print}(x)$

$x = \text{re.findall} ("a-2, bA-2", \text{txt})$   
 $\text{print}(x)$

$x = \text{re.findall} ("(*3", \text{txt})$   
 $\text{print}(x)$

ALNRAO

Output

[ '3', '4' ]

{ }

[]

let ' [val] ' () => val

(let, [a of] ) fibon

(let " [val] " ) fibon



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**  
Department of Computer Science

ALNIRAO

txt  
string = {"Big 24 Black 13 Bug bit a big black 46 bear"}.

x = re.findall ("Block \$", txt)  
print (x)

x = re.findall ("CR", txt)  
print (x)

x = re.findall ("Bug Right", txt)  
print (x)

x = re.findall ("ig {23", txt)  
print (x)

Output

{'Block'}

{'Ch', 'Cr'}

'Bug'

{'ig', 'ig'}

(net, "Σ x")

net, "S - A\\$ S - D"

(net, "P - O")

(net, "P - O")

(net, "P - O")

(net, "S - A\\$ S - D")



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

AL11040

$x = re.findall ("Block/2", +x1)$   
 $\text{print}(x)$

$x = re.findall ("\backslash D", +x1)$   
 $\text{print}(x)$

ALNRA

Output

('Block')

'B', 'l', 'o', 'k', ' ', 't', ' ', 'B', 'l', 'o', 'k', ' ', 't',  
'b', 'l', 'o', 'k', ' ', 'b', 'l', 'o', 'k', ' ', 'b', 'l', 'o', 'k',  
'b', 'l', 'o', 'k'

(text, "x4") { Hobnail }

text, "x4" } { Hobnail }

ALVARAO

Output

[1, 1, 1, 1, 1, 1, 1, 1]

('B', '1', 'g', '2', 'B', '1', 'o', 'c', 'k', '1', '3', 'B', '1', 'u', 'g',  
'y', '1', 't', 'o', 'b', '1', 'g', 'b', '1', 't', 'd', '1', 'c', 'k', 'b', '1', 'f', 'k',  
'b', 'e', 'o', 'v')

{1, 1, 1, 1, 1, 1}

('B', '1', 'g', '2', '1', 'B', '1', 'o', 'c', 'k', '1', '3', 'B', '1', 'g',  
'y', '1', 't', 'o', 'b', '1', 'g', 'b', '1', 't', 'd', '1', 'c', 'k', '1', '3',  
'b', 'e', 'o', 'v')



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALM RAD

`x=re.findall ("\\s",txt)  
print (x)`

`x=re.findall ("\\S",txt)  
print (x)`

`x=re.findall ("\\w",txt)  
print (x)`

`x=re.findall ("\\W",txt)  
print (x)`



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALNRAO

Practical - S

Aim: Sending Network Handling

import smtplib

```
gmailaddress = input("What is your account? \n")
gmailpassword = input("What is your password? \n")
mailto = input("What email address do you want to share? \n")
msg = input("Write message? \n")
mailserver = smtplib.SMTP('smtp.gmail.com', 587)
```

mailserver.starttls()

```
mailserver.login(gmailaddress, gmailpassword)
mailserver.sendmail(gmailaddress, mailto, msg)
print("Sent!")
mailserver.quit()
```

ALNRAO

Output

What is gmail account: apponietao@gmail.com  
password:

What email address yo want to send:opp

Sent

(Sat, 21<sup>st</sup>) 11:50 AM  
(pt) 100%

(Sat, 21<sup>st</sup>) 11:50 AM  
(pt) 100%

ALNRAO

Output : 2 - lost and

Email : Whatodo12345@gmail.com

Password: Wakondaforever

From: Whatodo12345@gmail.com

To: None

CC: None

Title: None

Body: Outputting



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**  
Department of Computer Science

ACNRA8

Step 1: (on command prompt) pip install easyimap

```
import easyimap
login = input("Enter your Email - ID:")
password = input("Enter your password:")
imapper = easyimap.connect('imap.gmail.com', login, password)
for mail_id in imapper.list_ids(limit=15):
 mail = imapper.mail(mail_id)
 print("From : ", mail.from_addr)
 print("To : ", mail.to)
 print("Cc : ", mail.cc)
 print("Title : ", mail.title)
 print("Body : ", mail.body)
 print("Attachments: ", mail.attachments)
 for attachment in mail.attachments:
 f = open("attachments/" + attachment[0], "wb")
 f.write(attachment[1])
 f.close.
```



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

PLNRAD

```
import smtplib
import easyimap
```

```
gmailaddress = input("What is gmail account ? \n")
gmailpassword = input("What is your password ?")
```



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

PCNRAO

### Practical - S(C)

```
import time
```

```
import socket
```

```
import sys
```

```
print("Downloading chat room\n")
```

```
c = socket.socket()
```

```
host = socket.gethostbyname(host)
```

```
print(host, "(", 's, ()\n")
```

```
hostadd = input(str("Enter server address"))
```

```
n1 = input(str("Enter your name"))
```

```
port = 1234
```

```
print("\n To connect to ", hostadd, " (", port, ")")
```

```
time.sleep(1)
```

```
c.connect((hostname, port))
```

```
print("Connected...\n")
```

```
c.send(n1.encode())
```

```
c.n1 = c.recv(1024)
```

```
c.n1 = c.n1.decode()
```

"The man" Enter (e) for

ALNRAO

Downloading Chat room  
Connected by (192.168.190.3804)  
Enter your name

Neeraj have joined the room

Neeraj : Hi

Yash : Hi



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**  
Department of Computer Science ALN RAO

While True:

```
msg = c.recv(1024)
msg = msg.decode()
print(c.getname, ":", msg)
msg = input(str("me:"))
IP msg = "(e)"
msg = "left room"
c.send(msg.encode())
print("\n")
break
c.send(msg.encode())
```



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALNIRAM

SNMPLIB

Syntax

SNMP.setdebug level (level)

description

Set the debug level  
A value of 1 or 2  
for level results in debug  
message for connection  
and for all message  
sent to and received  
from the server.

SNMP.docmd (cmd, query s= "")

Send a command to cmd  
to the server  
connect to a host on  
a given port  
Identify yourself to an  
SNMP server using  
ECHO

SNMP.connect (host=local, host, port=0)

SNMP.echo (name = "")

Identify yourself to the  
SNMP server using  
ECHO

SNMP.hello (name=')

Return true if name is  
part of snmp server  
extension server  
Check the validity of  
and address on the  
server using SNMP URL  
log in on SNMP server  
that requires authentication

SNMP.has - extn (name)

SNMP.verify (addr)

SNMP.logon()



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

PLNR A

SMTP. sendMail

Sends the mail

SNMP.send-message

This is a convenient method for calling sendMail() with the message represented by an email message

SMTP.quit

Close the connection


**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**  
 Department of Computer Science

Page No.

10P

|    | Syntax                                             | Description                            |
|----|----------------------------------------------------|----------------------------------------|
| 1  | listids(limit=10, criterion=None)                  | Returns list of available<br>mail id's |
| 2  | listup(limit=0, criterion=None, include_row=False) | Returns list of mail<br>object         |
| 3  | unspen (limit=10)                                  | Returns list of mail object            |
| 4  | mail(unid, include_row=False)                      | Returns mail object                    |
| 5  | change_mail_box(mailbox)                           | Change mail box                        |
| 6  | quit()                                             | close and logout                       |
| 7  | mail object                                        | mail object                            |
| 8  | uid                                                | Returns uid(type=1)                    |
| 9  | title                                              | Returns string of search<br>header     |
| 10 | from_a_dor                                         | Returns string of From<br>header       |
| 11 | to                                                 | Returns string of To<br>header         |
|    |                                                    | ... string of 'date'                   |



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALMRAO

|    |              |                                       |
|----|--------------|---------------------------------------|
| 13 | body         | Returns string of 'Body'              |
| 14 | content-type | Returns string of content type        |
| 15 | references   | Returns string of 'Referend trade'    |
| 16 | replay-to    | Returns string of replay-to           |
| 17 | return-path  | Returns string of return-path         |
| 18 | mime-version | Returns string of MIME-version header |
| 19 | Message-ID   | Returns string of 'Message'           |
| 20 | attachments  | Returns list of tuples                |



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**  
Department of Computer Science

ALNRAO

| Base Classes                 | Description                                                                                | Pre |
|------------------------------|--------------------------------------------------------------------------------------------|-----|
| exception BaseException      | The base class for all built-in exceptions                                                 |     |
| exception Exception          | All built-in and all user-defined exceptions are derived from this class                   |     |
| exception ArithmeticError    | The base class for arithmetic errors: OverflowError, ZeroDivisionError, FloatingPointError |     |
| exception BufferError        | Raised when a buffer related operation cannot be performed                                 |     |
| exception LookupError        | The base class for the exceptions IndexError, KeyError                                     |     |
| Concrete exceptions          |                                                                                            |     |
| exception AssertionError     | Raised when an assert statement fails                                                      |     |
| exception AttributeError     | Raised when a attribute reference fails                                                    |     |
| exception EOFError           | Raised when input() hits EOF without reading any data                                      |     |
| exception FloatingPointError | Not currently used                                                                         |     |
| exception GeneratorExit      | Raised when generator or coroutine is closed                                               |     |



# SHETH L.U.J. COLLEGE OF ARTS & SIR M.V. COLLEGE OF SCIENCE & COMMERCE

Department of Computer Science

ALNIRAO

exception ImportError

Raised when import has trouble loading module

exception ModuleNotFoundError A subclass of ImportError when import cannot be located

exception KeyError

Raised when a mapping(dictionary) key is not found

exception KeyboardInterrupt Raised when the user hits interrupt

exception MemoryError

Raised when operation runs out of memory

exception NameError

Raised when local or global name is not found

exception NotImplemented

Raised when required derived classes to override the method derived from Runtime Error

3 exception OSError [arg] [forg]

exception OSError([errno, strerror[, filename], winerror [, filename2]]])

This exception is raised when a system function returns a system related error

4 exception OverflowError

Raised when result of arithmetic operation is too large to represent

5 exception RecursionError

Raised when interpreter detects max recursion depth. derived from Runtime Error

17 exception ReferenceError

Raised when weak reference proxy is used to access an attribute after garbage has been collected

exception Runtime Error

Raised when error is detected doesn't fall in any other categories

exception StopIteration

Raised when next and \_\_iter\_\_ method has no items to be produced

exception StopAsyncIteration

Raised when \_\_next\_\_ method of asynchronous object stop the iteration

exception SyntaxError

Raised when parser encounters syntax error.

exception IndentationError

Base class for syntax related to incorrect indentation

exception TabError

Raised when indentation contains use of tabs and spaces

exception SystemError

Raised when interpreter finds an internal error, but situation does not look serious

exception SystemExit

- Exception raised by sys.exit()

exception TypeError

Raised when operation or function is applied to object of inappropriate type

exception UnboundLocalError

Raised when reference to local variable in a function or method, has no value

exception UnicodeError

Raised when a Unicode related encoding or decoding error occurs

exception UnicodeEncodeError

Raised when a Unicode errors occurs during encoding

exception UnicodeDecodeError

Raised when a Unicode error occurs during decoding

exception ValueError

Raised when an operation or function receives argument of inappropriate value



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALMIRADA

Raised when

exception ZeroDivisionError

Raised when argument of division  
or modulo operation is zero

exception EnvironmentError

exception IOError

exception WindowsError

OS exceptions

exception BlockingIOError

Raised when an operation would block  
of an object set for non-blocking operation

exception ChildProcessError

Raised when an operation on a child  
process failed

exception ConnectionError

A base class for connection-related issues

exception BrokenPipeError

Raised when trying to write on a pipe  
while the other end is closed

exception ConnectionAbortedError

Raised when attempt is aborted by peer

exception ConnectionRefusedError

Raised when attempt is refused by peer

exception ConnectionResetError

Raised when connection is reset by peer

exception FileNotFoundError

Raised when trying to create a file or  
directory that already exists

FileNotFoundException

exception FileNotFoundError Raised when file or directory requested doesn't exists

exception IsADirectoryError Raised when operation like os.remove is requested requested in directory

exception NotADirectoryError Raised when directory operation like os.listdir() is requested when something is not directory

exception PermissionError Raised when trying to run operation without any adequate rights

exception ProcessLookupError Raised when given process doesn't exist.

exception TimeoutError Raised when system function timed out of system level



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALNRAO

Warning

exception Warning

Base class for warning categories

exception UserWarning

Base class for warning generated by user

exception DeprecationWarning Base class for warning about deprecated features

exception SyntaxWarning Base class for dubious syntax

exception RuntimeWarning Base class for dubious runtime behaviour

exception FutureWarning Base class for warning about deprecated features

exception ImportWarning Base class for mistakes in import

exception UnicodeWarning Base class for warning related to unicode

exception BytesWarning Base class for bytes and bytearray

exception ResourceWarning Base class for warnings related to resource usage.



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

PLTIPAO

Practical - 06

Aim: Exception Handling

777 (10/(1/0))

Traceback (most recent call last):  
file "<stdin>", line 1 in <module>

ZeroDivisionError: division by zero

777 R 4 + spam\*3

Traceback (most recent call last):

file "<stdin>", line 1 in <module>

NameError: name spam is not defined

777 '2' + 28

Traceback (most recent call last):

file "<stdin>", line 1 in <module>

TypeError:

TypeError: can't convert 'int' object to str implicitly



SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE  
Department of Computer Science B.Tech

try:

```
p = open('win.txt')
p.write("lose")
except FileNotFoundError:
 print("file not found")
finally:
 p.close()
```

```
import
from math import win
```

try

```
main()
except ImportError:
 print("Module not found")
```

A STREAM TO THE END OF THE FILE

ALNRAO

28/11/14 Output file not found

Module not found

(C:\1D\el)\el

(file not found) module  
calculator and combinator  
were not much used in this

Strange file

(file not found) module

calculator and combinator

and both were quite used



# SHETH L.U.J. COLLEGE OF ARTS & SIR M.V. COLLEGE OF SCIENCE & COMMERCE

Department of Computer Science

ALNRAE

try:

$$a = 3$$

if  $a < 4$ :

$$b = a / (a - 3)$$

print "Value of b: ", b

except (ZeroDivisionError, NameError):

print "An Error Occurred and Handled"

x = "hello"

if not type(x) is int:

raise TypeError ("Only integers are allowed")

ALMIRAO

Print

Error Occurred and Handled

Only Integers are allowed



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALN RAO

```
while True:
 try:
 x = int(input("Please enter number:"))
 break
 except ValueError:
 print("Only There is no valid number")
```

```
while True:
 try:
 x = int(input("Enter number"))
 y = int(input("Enter number"))
 z = x + y
 except OverflowError:
 print("Value is too big")
```

```
while True:
 try:
 x = int(input("Enter number"))
 x + '8'
 except TypeError:
 print("Can't convert string to int")
```

ALNRAO

Output:

Please enter Number : K

There is no valid number

is invalid

Enter number: 2568987468214372

Enter number = 547821561928765

Value is too big

Enter number: 2

Convert string to int



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**  
Department of Computer Science

ALVNR

IO Error

try

file\_ = open ("file.txt", "r")

except IO Error

print ("file not found")

else:

print ("The file opened successfully")

file.close()

ALNIRAO

Output:

File not found F



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALNRAO

Key Error

try:  
array = {"a": 1, "b": 2}  
print array['c']  
  
except KeyError:  
 print("KeyError: There is no 'c'")

else  
 print("Successfully Done!")

ALNRAO

Output

There is no C



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ACNAPAC

Name Error

my  
print(x)

except NameError:

print("Variable x is not defined")

except:

print("Some thing else went wrong")

ALNRA

Output -

Variable x is not defined



# SHETH L.U.J. COLLEGE OF ARTS & SIR M.V. COLLEGE OF SCIENCE & COMMERCE

Department of Computer Science

ALNRAO

Zero Division Error.

def divide(x,y)

try:

result = x/y

print ("Your answer: ", result)

except ZeroDivisionError:

print ("Sorry you cannot divide by zero")

divide(3,0)

PINTER

Output

Sorry you cannot divide by 200



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**  
Department of Computer Science

AI-N-202

Type Error:

try:

x = hello!

if not type(x) is int:

raise TypeError:

else

print('Done')

except TypeError:

print("Only Integers are allowed")

ALNRAU

Output

Only Integers are allowed

(Hence float is not allowed)



# **SHETH L.U.J. COLLEGE OF ARTS & SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALNRAO

Syntax Error

by  
y = "Hello World!"

print z

except Syntax Error:

print("There is wrong Syntax is print")

else:

print ("Success")

SCHOOL OF COMPUTER SCIENCE & ENGINEERING  
SCHOOL OF SCIENCE & COMPUTER SCIENCE  
DEPARTMENT OF COMPUTER SCIENCE

ANURBU

Output

There is wrong syntax in print

'allah'

on 21 (x) apt  
Date 1995 06 01



SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE  
Department of Computer Science

PLAT#80

Index Error

try:

array = [0, 1, 2]

print array(3)

except IndexError:

print ('list index out of range')

else:

print ('Done!')

ALNRAU

Output

List Index is , out of range  
of interval [0, 10].  
Value = 11.

21. (below) print a list which rotates 4 times  
and prints it.

(below) print



**SHETH L.U.J. COLLEGE OF ARTS &  
SIR M.V. COLLEGE OF SCIENCE & COMMERCE**

Department of Computer Science

ALN

Multiple exception Error

try

a = 10 / b

except ArithmeticError

ALNRA

Output

Arithmatic Exception

(6.10) word  
(i) your last