**Practical-12: Networking and Multithreading using Python**

# Create server.py and client.py file and create chat applications for continuous communication between both.

## CODE :-

* **Server.py :**

import socket localhost="127.0.0.1" port=1234

server=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM) server.bind((localhost,port))

server.listen(1)

print("Server is ready ; And waiting for client ...")

cc,addr=server.accept() msg=''

while TRUE: msg=cc.recv(1024) in\_msg=msg.decode() if in\_msg=='bye':

break

print("From client : ",in\_msg) out\_msg=input() cc.sendall(bytes(out\_msg,'UTF-8'))

print("Connection Terminated")

## Client.py :

import socket localhost="127.0.0.1" port=1234

client=socket.socket(socket.AF\_INET,socket.SOCK\_STREAM) client.connect((localhost,port))

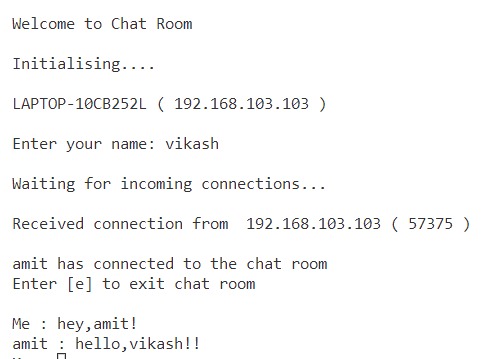
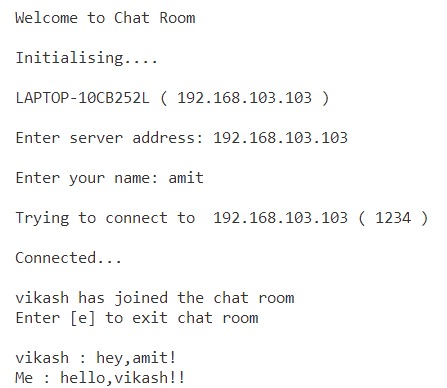
client.sendall(bytes("This is client",'UTF-8')) while TRUE:

msg=client.recv(1024) in\_msg=msg.decode() print("From server : ",in\_msg) out\_msg=input()

client.sendall(bytes(out\_msg,'UTF-8')) if out\_msg=='bye':

break client.close()

**Output:**



# Write a python program to demonstrate usage of multi threading.

## Code:

from threading import \* from time import \*

print("By Tirth Patel(20012011129)") class mythread1(Thread):

def run(self):

for i in range(3): print("Processing Thread-1") sleep(1)

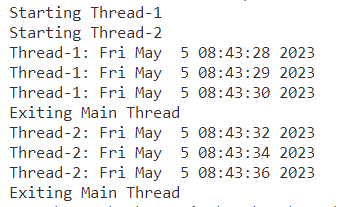
class mythread2(Thread): def run(self):

for i in range(3): print("Processing Thread-2")

obj1=mythread1() obj2=mythread2()

obj1.start() obj2.start()

**Output:**



# Write a python code to send email by passing a content.

## Code:

from email import message

import smtplib as s

obj=s.SMTP("smtp.gmail.com",587)

# connecting object to server

obj.starttls()

#establishing secure environment.It will enable secure socket layer.

obj.login("amitgoswami21@gnu.ac.in","uvpce@123")

sender=" amitgoswami21@gnu.ac.in "

receiver=["amitgoswami698@gmail.com"]

subject="Testing SMTP protocol"

body=" It is no-reply mail"

msg="Subject:{}\n\n{}".format(subject,body)

obj.sendmail(sender,receiver,msg)

obj.quit()

## Output:

