# Laukik CN 5,6,7,8,11, 12

Artificial Intelligence and Machine Learning (Savitribai Phule Pune University)

**Name** : Laukik Nitin Marathe
**Roll No** : TEAD21153
**Subject** : Computer Networks
**Class** : TE
**Branch** : AI&DS
**Assignment No :**5

Socket Programming using Python/C/C++/Java.
a. TCP Client, TCP Server.
b. UDP Client, UDP Server.

CODE:

a.TCP Client, TCP Server.

@server.py

```
import socket
import time
# create a socket object
serversocket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# get local machine name
host = socket.gethostname()
port = 9999
# bind to the port
serversocket.bind((host, port))
# queue up to 5 requests
serversocket.listen(5)
while True:
   # establish a connection
   clientsocket,addr = serversocket.accept()

   print("Got a connection from %s" % str(addr))
   currentTime = time.ctime(time.time()) + "\r\n"
   clientsocket.send(currentTime.encode('ascii'))
   clientsocket.close()
```

@client.py

```
import socket
# create a socket object
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
# get local machine name
host = socket.gethostname()
port = 9999
# connection to hostname on the port.
s.connect((host, port))
# Receive no more than 1024 bytes
tm = s.recv(1024)
```

```
s.close()
print("The time got from the server is %s" % tm.decode('ascii'))
```

b.UDP Client, UDP Server.

@Server.py

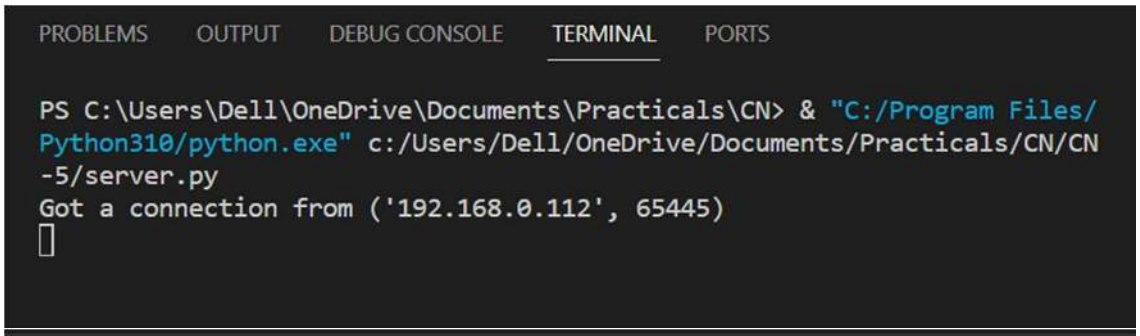
```
import socket
localIP     = "127.0.0.1"
localPort   = 20001
bufferSize  = 1024
msgFromServer       = "Hello UDP Client"
bytesToSend         = str.encode(msgFromServer)

# Create a datagram socket
UDPServerSocket = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)
# Bind to address and ip
UDPServerSocket.bind((localIP, localPort))
print("UDP server up and listening")
# Listen for incoming datagrams
while(True):
    bytesAddressPair = UDPServerSocket.recvfrom(bufferSize)
    message = bytesAddressPair[0]
    address = bytesAddressPair[1]
    clientMsg = "Message from Client:{}".format(message)
    clientIP  = "Client IP Address:{}".format(address)
    print(clientMsg)
    print(clientIP)
    # Sending a reply to client
    UDPServerSocket.sendto(bytesToSend, address)
```

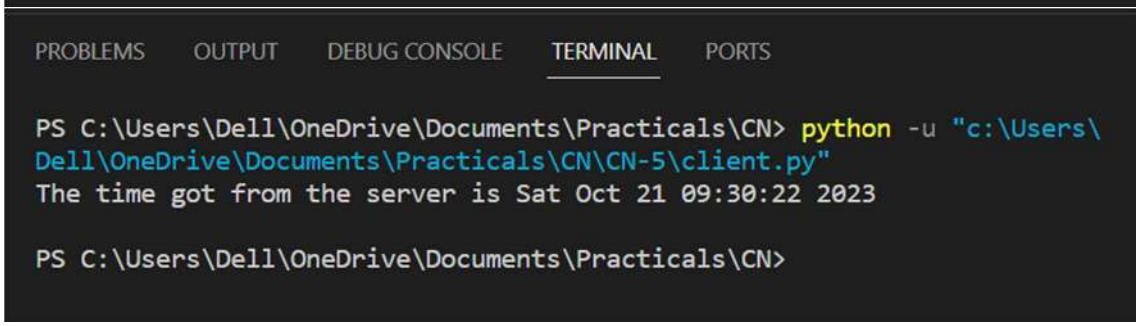
@client.py

```
import socket

msgFromClient       = "Hello UDP Server"
bytesToSend         = str.encode(msgFromClient)
serverAddressPort   = ("127.0.0.1", 20001)
bufferSize          = 1024
# Create a UDP socket at client side
UDPClientSocket = socket.socket(family=socket.AF_INET, type=socket.SOCK_DGRAM)
# Send to server using created UDP socket
UDPClientSocket.sendto(bytesToSend, serverAddressPort)
msgFromServer = UDPClientSocket.recvfrom(bufferSize)
msg = "Message from Server {}".format(msgFromServer[0])
print(msg)
```

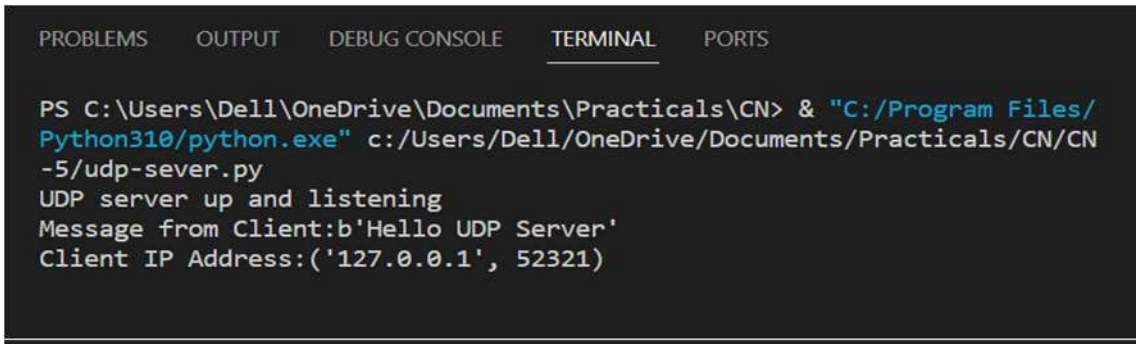
OUTPUT:

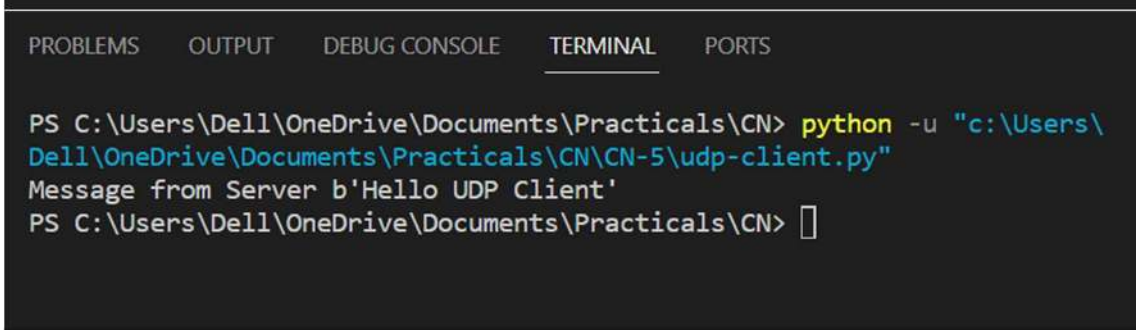
a.TCP Client, TCP Server.




## b.UDP Client, UDP Server.



**Name** : Laukik Nitin Marathe
**Roll No** : TEAD21153
**Subject** : Computer Networks
**Class** : TE
**Branch** : AI&DS
**Assignment No :**6

Write a program using TCP socket for wired network for following
a. Say Hello to Each other.
b. File transfer.

CODE:

a. Say Hello to Each other.

@server.py

```
import socket

HOST = '127.0.0.1'  # Standard loopback interface address (localhost)
PORT = 3333

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    conn, addr = s.accept()
    with conn:
        print('Connected by', addr)
        while True:
            data = conn.recv(1024).decode()
            print('Client says:', data)
            if data == 'stop':
                break
            str2 = input("Enter your message: ")
            conn.sendall(str2.encode())
```

@client.py

```
import socket

HOST = '127.0.0.1'  # The server's hostname or IP address
PORT = 3333

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.connect((HOST, PORT))
    while True:
        message = input("Enter your message: ")
        s.sendall(message.encode())
        if message == 'stop':
```

```
            break
        data = s.recv(1024).decode()
        print('Server says:', data)
```

b. File transfer.

@Server.py

```
import socket
host = "127.0.0.1"
port = 12000

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((host,port))

f= open('Myfile2.txt','wb')
print('New file created')
data, addr = sock.recvfrom(1024)

while(data):
    print(data)
    if data.decode("utf-8")=="Now":
        break
    f.write(data)
    data, addr = sock.recvfrom(1024)

print('File is successfully received!!!')
f.close()
f = open('Myfile2.txt','r')
print(f.read)

f.close()
sock.close()
print('Connection closed!')
```

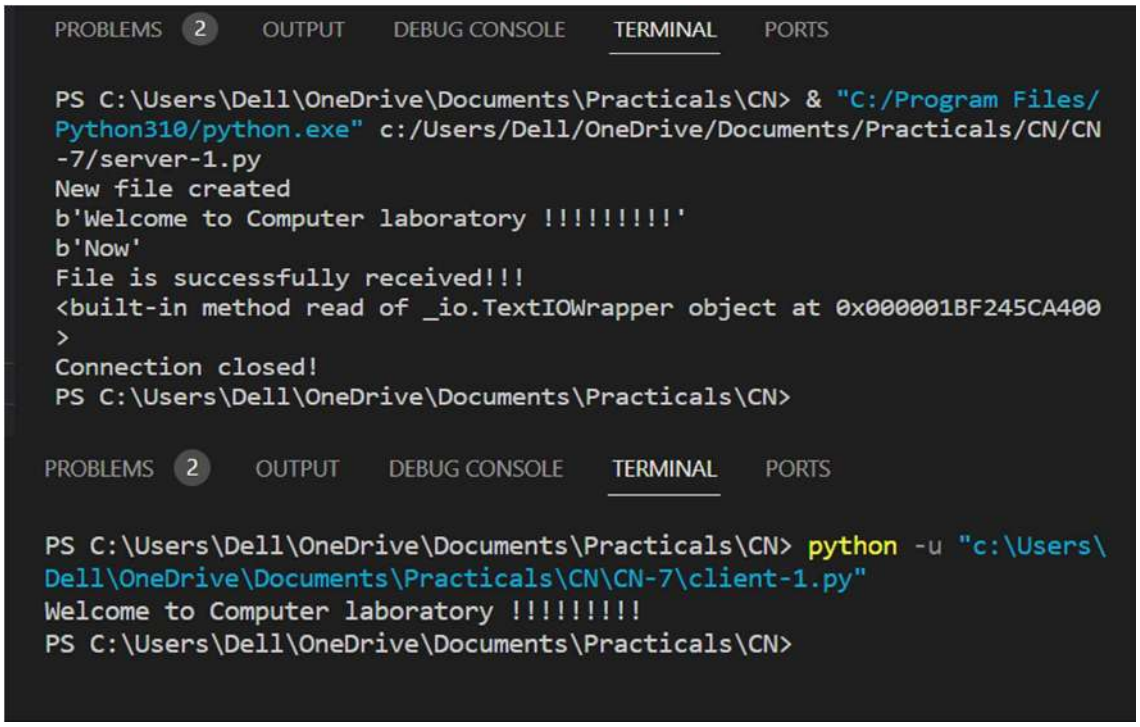
@client.py

```
import socket

host = "127.0.0.1"
port = 12000
buffer_size = 1024
file_name = 'Myfile.txt'

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

f = open("Myfile.txt", "r")
data = f.read(buffer_size)

while data:
    print(data)
```

```
    if(sock.sendto(str.encode(data), (host, port))):
        data = f.read(buffer_size)
sock.sendto(str.encode("Now"),(host, port))
sock.close()
f.close()
```

OUTPUT:

b. File transfer.



**Name** : Laukik Nitin Marathe
**Roll No** : TEAD21153
**Subject** : Computer Networks
**Class** : TE
**Branch** : AI&DS
**Assignment No :**7

Write a program using UDP Sockets to enable file transfer (Script, Text, Audio and Video one file each) between two machines.

CODE:

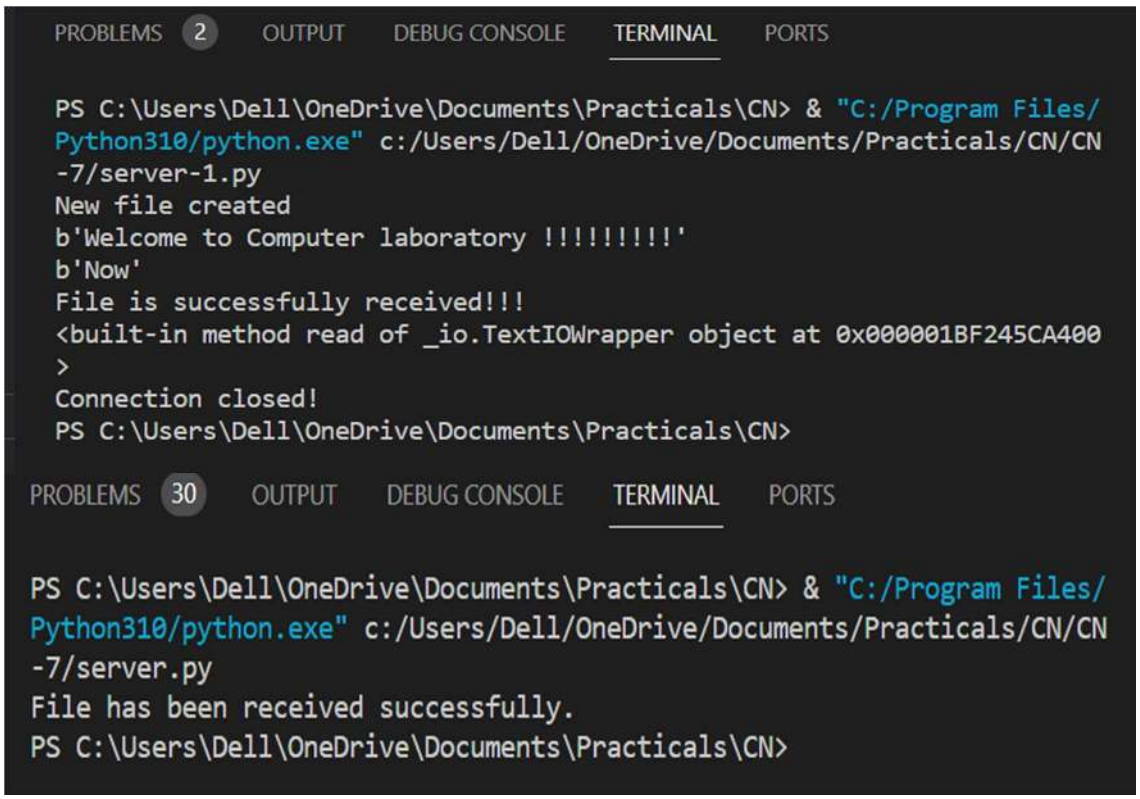
@Server.py

```
import socket

# Set up the server
HOST = '127.0.0.1'
PORT = 12345

# Create a UDP socket
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.bind((HOST, PORT))

# Receive the file
data, addr = s.recvfrom(1024)
with open('received_file.txt', 'wb') as f:
    f.write(data)

print("File has been received successfully.")
```

@Client.py

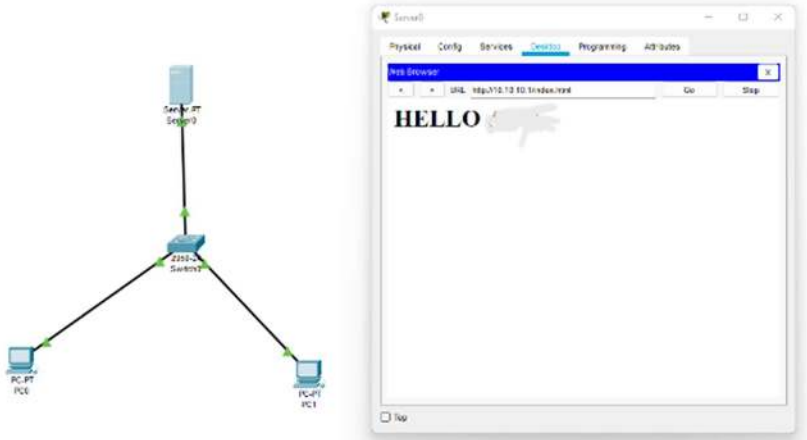
```
import socket

# Set up the client
HOST = '127.0.0.1'
PORT = 12345
# Create a UDP socket
s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Read the file
with open('your_file.txt', 'rb') as f:
    data = f.read()

# Send the file
s.sendto(data, (HOST, PORT))
print("File has been sent successfully.")
```

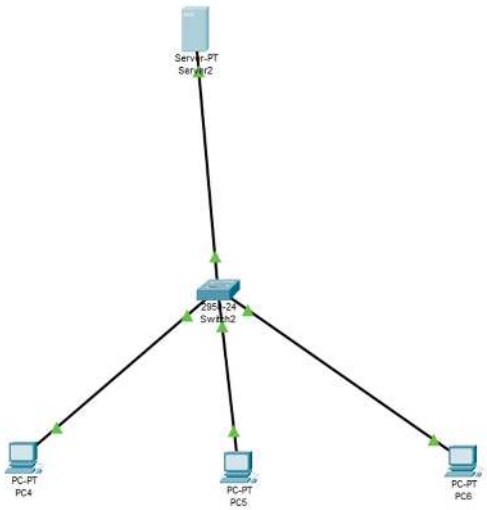
OUTPUT:



**Name** : Laukik Nitin Marathe
**Roll No** : TEAD21153
**Subject** : Computer Networks
**Class** : TE
**Branch** : AI&DS
**Assignment No :**8

Study and Analyze the performance of HTTP, HTTPS and FTP protocol using Packet tracer tool.

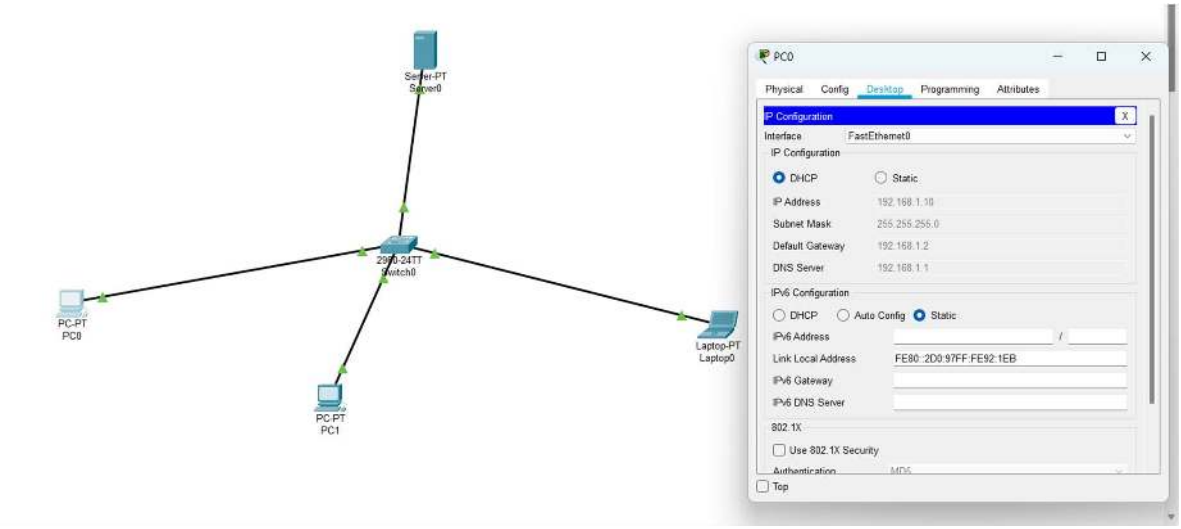


```
Packet Tracer SERVER Command Line 1.0
C:\>ipconfigs
Invalid Command.

C:\>ipconfig

FastEthernet0 Connection:(default port)

   Link-local IPv6 Address.....: FE80::2D0:D3FF:FE81:AE9
   IP Address.....: 10.10.10.4
   Subnet Mask.....: 255.0.0.0
   Default Gateway.....: 0.0.0.0

C:\>ftp10.10.10.4
Invalid Command.

C:\>ftp 10.10.10.4
Trying to connect...10.10.10.4
Connected to 10.10.10.4
220- Welcome to PT Ftp server
Username:abc
331- Username ok, need password
Password:
230- Logged in
(passive mode On)
ftp>put MyFile.txt
%Error opening c:MyFile.txt (No such file or directory)
ftp>put Sumit.txt
%Error opening c:Sumit.txt (No such file or directory)
ftp>put sumit.txt

Writing file sumit.txt to 10.10.10.4:
File transfer in progress...

[Transfer complete - 82 bytes]

02 bytes copied in 0.035 secs (2342 bytes/sec)
ftp>cd /http
ftp>
Working directory changed to /http successfully
ftp>put sumit.txt

Writing file sumit.txt to 10.10.10.4:
File transfer in progress...

[Transfer complete - 82 bytes]

82 bytes copied in 0.017 secs (4823 bytes/sec)
ftp>ftp 10.10.10.4
 Invalid or non supported command.
ftp>dir

Listing /http directory from 10.10.10.4:
0    : copyrights.html                                  14053
1    : cscoptlogo177x111.jpg                            9628
2    : helloworld.html                                  62
3    : image.html                                       253
4    : index.html                                       369
5    : sumit.txt                                        82
ftp>
```

**Name** : Laukik Nitin Marathe
**Roll No** : TEAD21153
**Subject** : Computer Networks
**Class** : TE
**Branch** : AI&DS
**Assignment No :**11

Installing and configuring DHCP server and assign IP addresses to client machines using DHCP server.



**Name** : Laukik Nitin Marathe
**Roll No** : TEAD21153
**Subject** : Computer Networks
**Class** : TE
**Branch** : AI&DS
**Assignment No :**12

Write a program for DNS lookup. Given an IP address input, it should return URL and vice-versa.

CODE:

```
import socket

# Sample data (IP to URL mapping)
ip_url_mapping = {
   '192.0.2.1': 'www.example.com',
   '192.0.2.2': 'www.openai.com',
   '192.0.2.3': 'www.google.com',
   '192.0.2.4': 'www.github.com'
}

# Function for IP to URL lookup
def ip_to_url(ip):
   return ip_url_mapping.get(ip, 'No URL found for this IP.')

# Function for URL to IP lookup
def url_to_ip(url):
   for ip, u in ip_url_mapping.items():
      if u == url:
         return ip
   return 'No IP found for this URL.'

# Sample usage
while True:
   user_input = input("Enter an IP address or URL (type 'exit' to quit): ")
   if user_input.lower() == 'exit':
      break
   if user_input.replace('.', '').isdigit():
      ip_address = user_input
      print(f"URL for {ip_address} is {ip_to_url(ip_address)}")
   else:
      url = user_input
      print(f"IP for {url} is {url_to_ip(url)}")
```

OUTPUT: