

The screenshot shows the VS Code interface with the Explorer sidebar on the left displaying a project named 'CN LAB'. The files listed are 'ClientTCP.py', 'ServerTCP.py', 'VLAN.pkt', and 'WLAN.pkt'. The 'ServerTCP.py' file is selected and its code is displayed in the main editor. The code is a Python script that sets up a TCP server on 127.0.0.1 at port 12000. It listens for a connection, receives a message, reads the file 'ServerTCP.py', and sends its contents back to the client. The terminal at the bottom shows the command to run the script and its output: 'The server is ready to receive'.

```
1 from socket import *
2 serverName='127.0.0.1'
3 serverPort = 12000
4 serverSocket = socket(AF_INET,SOCK_STREAM)
5 serverSocket.bind((serverName,serverPort))
6 serverSocket.listen(1)
7 while 1:
8     print ("The server is ready to receive")
9     connectionSocket, addr = serverSocket.accept()
10    sentence = connectionSocket.recv(1024).decode()
11    file=open(sentence,"r")
12
13    l=file.read(1024)
14    connectionSocket.send(l.encode())
15    print ("\nSent contents of" + sentence)
16    file.close()
17    connectionSocket.close()
18
```

Python: ServerTCP

PS C:\Users\de11\Desktop\CN_LAB> & C:/Users/de11/AppData/Local/Programs/Python/Python312/python.exe c:/Users/de11/Desktop/CN_LAB/ServerTCP.py

The server is ready to receive

Sent contents ofServerTCP.py

The server is ready to receive

The screenshot shows the VS Code interface with the Explorer sidebar on the left displaying a project named 'CN LAB'. The files listed are 'ClientTCP.py', 'ServerTCP.png', 'ServerTCP.py', 'VLAN.pkt', and 'WLAN.pkt'. The 'ClientTCP.py' file is selected and its code is displayed in the main editor. The code is a Python script that sets up a TCP client that connects to 127.0.0.1 at port 12000. It sends the contents of 'ServerTCP.py' to the server. The terminal at the bottom shows the command to run the script and its output: 'Enter file name: ServerTCP.py'.

```
1 from socket import *
2 serverName='127.0.0.1'
3 serverPort = 12000
4 serverSocket = socket(AF_INET,SOCK_STREAM)
5 serverSocket.bind((serverName,serverPort))
6 serverSocket.listen(1)
7 while 1:
8     print ("The server is ready to receive")
9     connectionSocket, addr = serverSocket.accept()
10    sentence = connectionSocket.recv(1024).decode()
11    file=open(sentence,"r")
12
13    l=file.read(1024)
14    connectionSocket.send(l.encode())
15    print ("\nSent contents of" + sentence)
16    file.close()
17    connectionSocket.close()
18
```

Python: ClientTCP

PS C:\Users\de11\Desktop\CN_LAB> & C:/Users/de11/AppData/Local/Programs/Python/Python312/python.exe c:/Users/de11/Desktop/CN_LAB/ClientTCP.py

Enter file name: ServerTCP.py

From Server:

```
from socket import *
serverName='127.0.0.1'
serverPort = 12000
serverSocket = socket(AF_INET,SOCK_STREAM)
serverSocket.bind((serverName,serverPort))
serverSocket.listen(1)
while 1:
    print ("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
```


EXPERIMENT T-15

Using TCP/IP sockets write a client server program to make client sending the file name and the server to send back the contents of the requested file if present.

Programs: ClientTCP.py

```
from socket import *
serverName = '127.0.0.1'
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName, serverPort))
sentence = input("Enter file name:")
clientSocket.send(sentence.encode())
fileContents = clientSocket.recv(1024).decode()
print("From server: ")
print(fileContents)
clientSocket.close()
```

ServerTCP.py

```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_STREAM)
serverSocket.bind((serverName, serverPort))
serverSocket.listen(1)
while 1:
    print("The server is ready to receive")
    connectionSocket, addr = serverSocket.accept()
    sentence = connectionSocket.recv(1024).decode()
    file = open(sentence, "r")
    file data = file.read(1024)
    connectionSocket.send(data.encode())
    print("Sent contents of " + sentence)
    file.close()
    connectionSocket.close()
```