LAB WEEK 11

3. 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.

Client.py

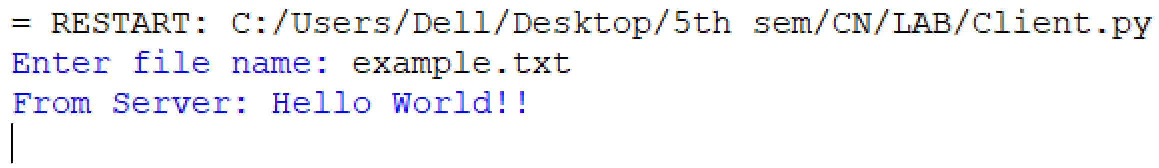
```
from socket import *
serverName = “127.0.0.1”
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_STREAM)
clientSocket.connect((serverName,serverPort))
sentence = input("Enter file name")
clientSocket.send(sentence.encode())
filecontents = clientSocket.recv(1024).decode()
print ('From Server:',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

Server.py

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

Output

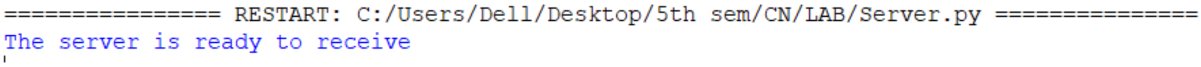
Client.py



```
= RESTART: C:/Users/Dell/Desktop/5th sem/CN/LAB/Client.py
Enter file name: example.txt
From Server: Hello World!!
```

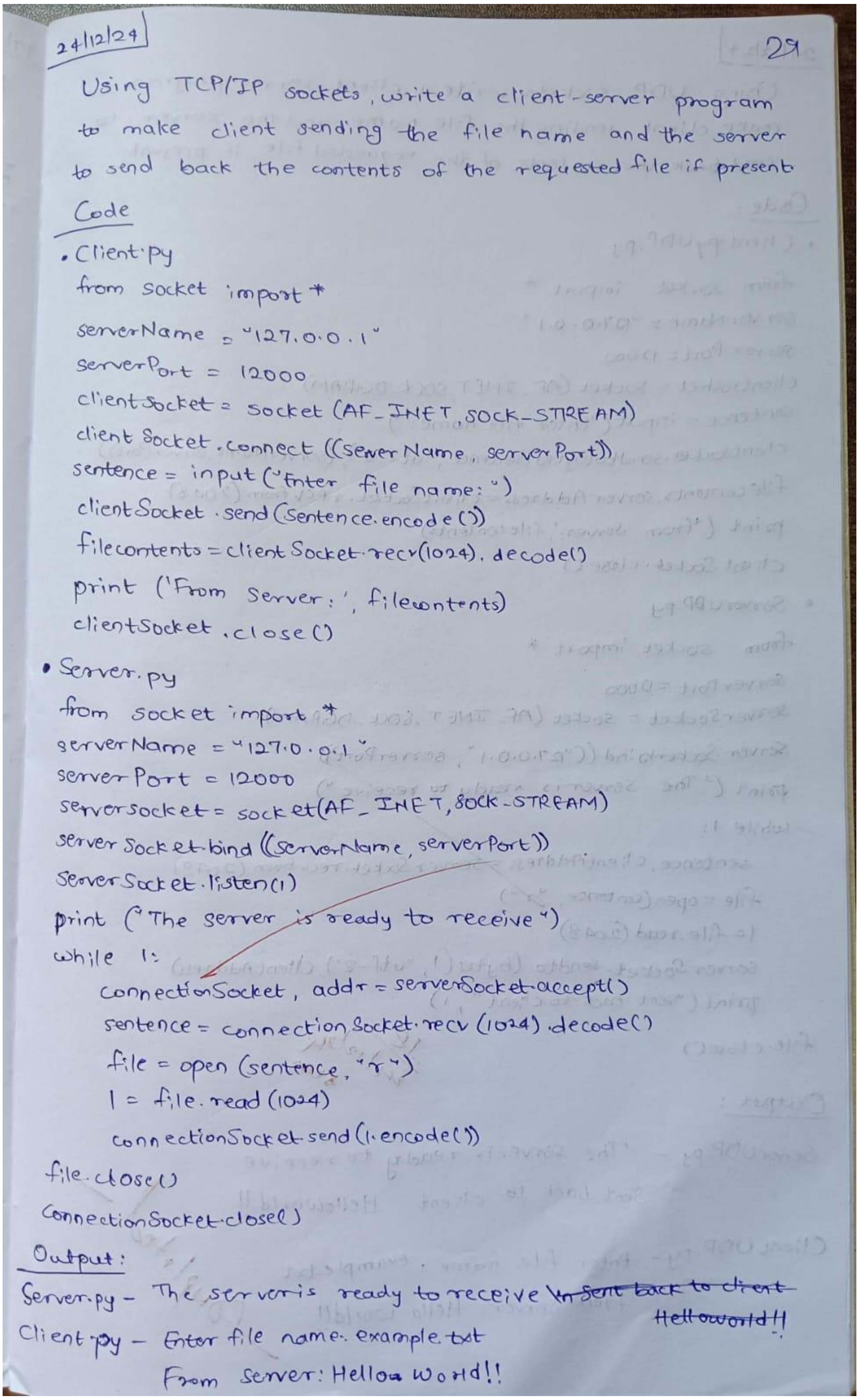
Figure 1: Output of Client.py

Server.py



```
================ RESTART: C:/Users/Dell/Desktop/5th sem/CN/LAB/Server.py ===============
The server is ready to receive
```

Figure 2: Output of Server.py



24/12/24

29

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

Code

- Client.py

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

- Server.py

```
from socket import *
serverName = "127.0.0.1"
server Port = 12000
serversocket = socket(AF_INET, SOCK_STREAM)
server Socket.bind ((serverName, serverPort))
ServerSocket.listen(1)
print ("The server is ready to receive")
while 1:
    connectionSocket, addr = serverSocket.accept()
    sentence = connection Socket.recv (1024).decode()
    file = open (sentence, "r")
    l = file.read (1024)
    connectionSocket.send (l.encode())
file.close()
ConnectionSocket.close()
```

Output:

Server.py - The server is ready to receive ~~\n Sent back to client Helloworld!!~~

Client.py - Enter file name: example.txt

From Server: Hello World!!

Figure 3: Observation Book

4. **Using UDP 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.**

<u>ClientUDP.py</u>

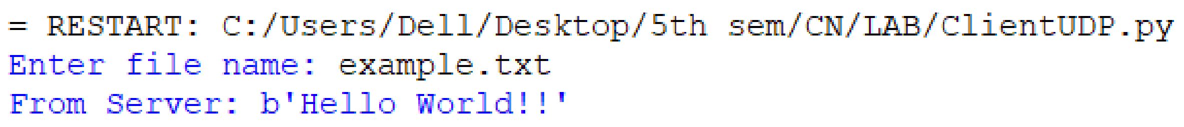
```
from socket import *
serverName = "127.0.0.1"
serverPort = 12000
clientSocket = socket(AF_INET, SOCK_DGRAM)
sentence = input("Enter file name")
clientSocket.sendto(bytes(sentence,"utf-8"),(serverName, serverPort))
filecontents,serverAddress = clientSocket.recvfrom(2048)
print ('From Server:', filecontents)
clientSocket.close()
```

<u>ServerUDP.py</u>

```
from socket import *
serverPort = 12000
serverSocket = socket(AF_INET, SOCK_DGRAM)
serverSocket.bind(("127.0.0.1", serverPort))
print ("The server is ready to receive")
while 1:
    sentence,clientAddress = serverSocket.recvfrom(2048)
    file=open(sentence,"r")
    l=file.read(2048)
    serverSocket.sendto(bytes(l,"utf-8"),clientAddress)
    print("sent back to client",l)
    file.close()
```

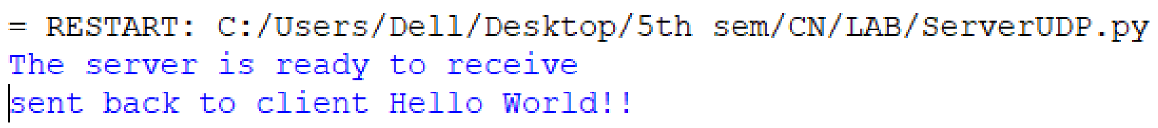
Output

<u>ClientUDP.py</u>



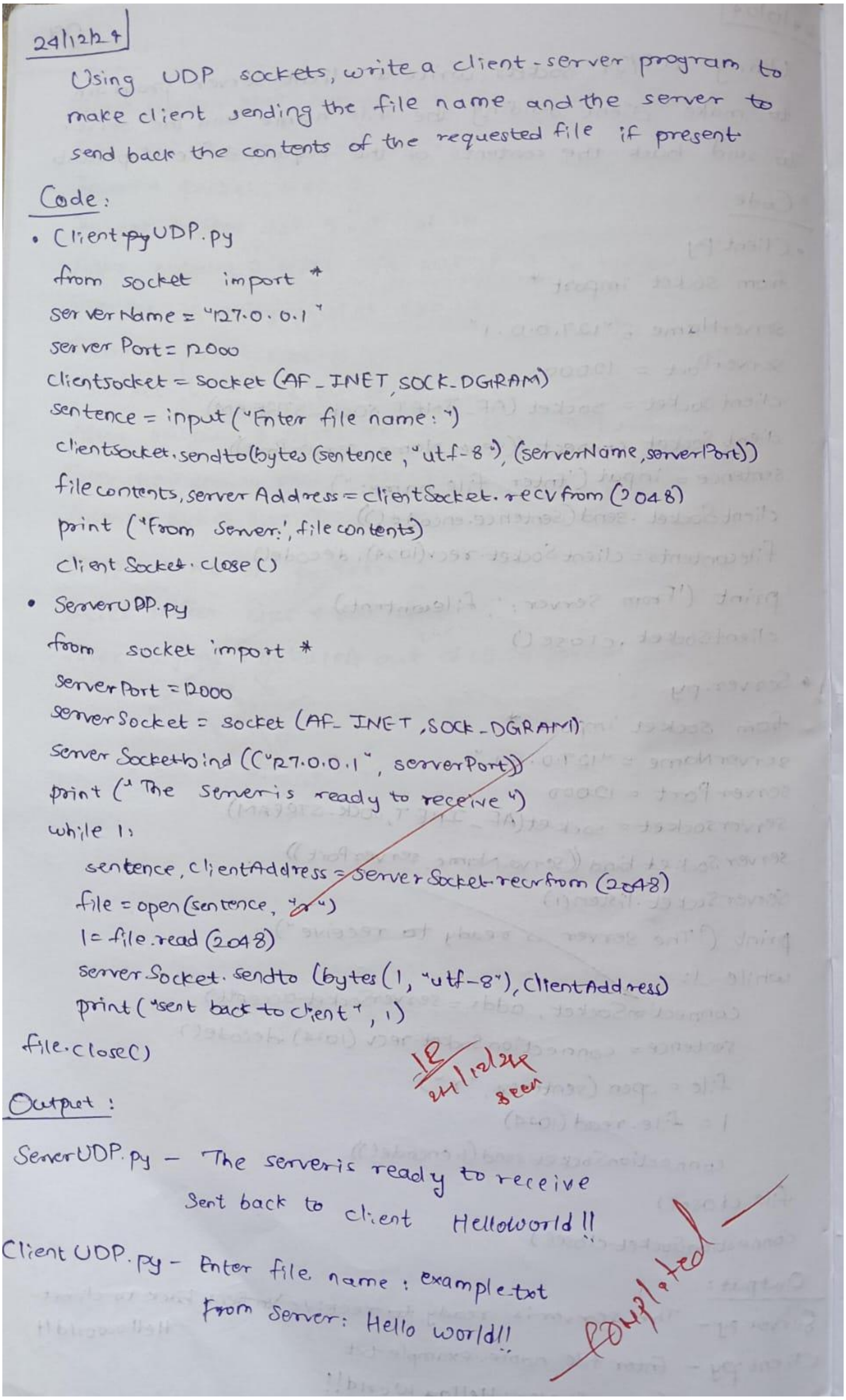
```
= RESTART: C:/Users/Dell/Desktop/5th sem/CN/LAB/ClientUDP.py
Enter file name: example.txt
From Server: b'Hello World!!'
```

Figure 4: Output of ClientUDP.py



```
= RESTART: C:/Users/Dell/Desktop/5th sem/CN/LAB/ServerUDP.py
The server is ready to receive
sent back to client Hello World!!
```

Figure 5: Output of ServerUDP.py



24/12/24

Using UDP 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

Code:

- Client~~py~~UDP.py

```
from socket import *
serverName = "127.0.0.1"
server Port= 12000
clientsocket = socket (AF_INET, SOCK_DGRAM)
sentence = input ("Enter file name: ")
clientsocket.sendto(bytes (sentence, "utf-8"), (serverName, serverPort))
filecontents, server Address = clientSocket. recvfrom (2048)
print ('From Server:', file contents)
client Socket. close ()
```

- ServerUDP.py

```
from socket import *
serverPort = 12000
serverSocket = socket (AF_INET, SOCK_DGRAM)
server Socketbind (("127.0.0.1", serverPort))
print ("The serveris ready to receive")
while 1:
    sentence, clientAddress = serverSocket.recvfrom (2048)
    file = open(sentence, "r")
    l= file.read (2048)
    serverSocket. sendto (bytes (l, "utf-8"), clientAddress)
    print ("sent back to client", l)
file.close()
```

12
24/12/24
seen

Output:

```
ServerUDP.py - The serveris ready to receive
               Sent back to client   Helloworld !!
Client UDP.py - Enter file name : example.txt
                From Server: Hello world!!
```

Completed

Figure 6: Observation Book