Experiment 8

Harsh Sandesara
Batch C, 49
UID: 2018130045

Aim:

Implementing socket programming in python

Theory:

Socket programming [1] is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server. They are the real backbones behind web browsing. In simpler terms there is a server and a client.
Sockets are the endpoints of a bidirectional communications channel. Sockets may communicate within a process, between processes on the same machine, or between processes on different continents.

A simple server-client program [1][2]:
Server:
A server has a bind() method which binds it to a specific ip and port so that it can listen to incoming requests on that ip and port. A server has a listen() method which puts the server into listen mode. This allows the server to listen to incoming connections. And last a server has an accept() and close() method. The accept method initiates a connection with the client and the close method closes the connection with the client.

Client:
The socket.connect(hosname, port) opens a TCP connection to hostname on the port.And socket.receive() receives the TCP message.

Code:

server.py

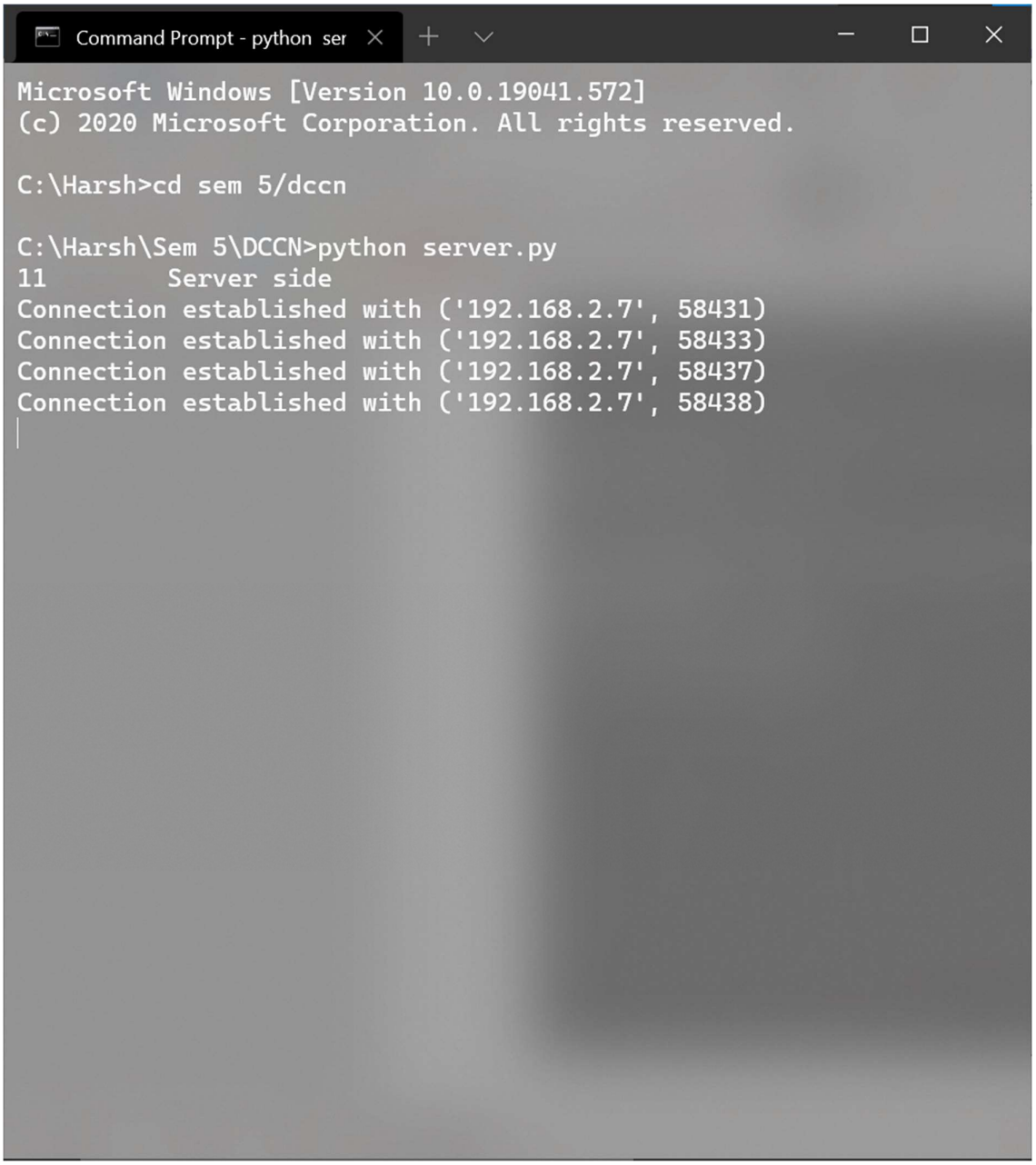
```
import socket
msg = 'Server side'
print(f'{len(msg):<10}'+msg)
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.bind((socket.gethostname(), 8000))
s.listen(5)
while True:
        client_socket, address = s.accept()
        print(f'Connection established with {address}')
        client_socket.send(bytes('Client Side', 'utf-8'))
```

client.py

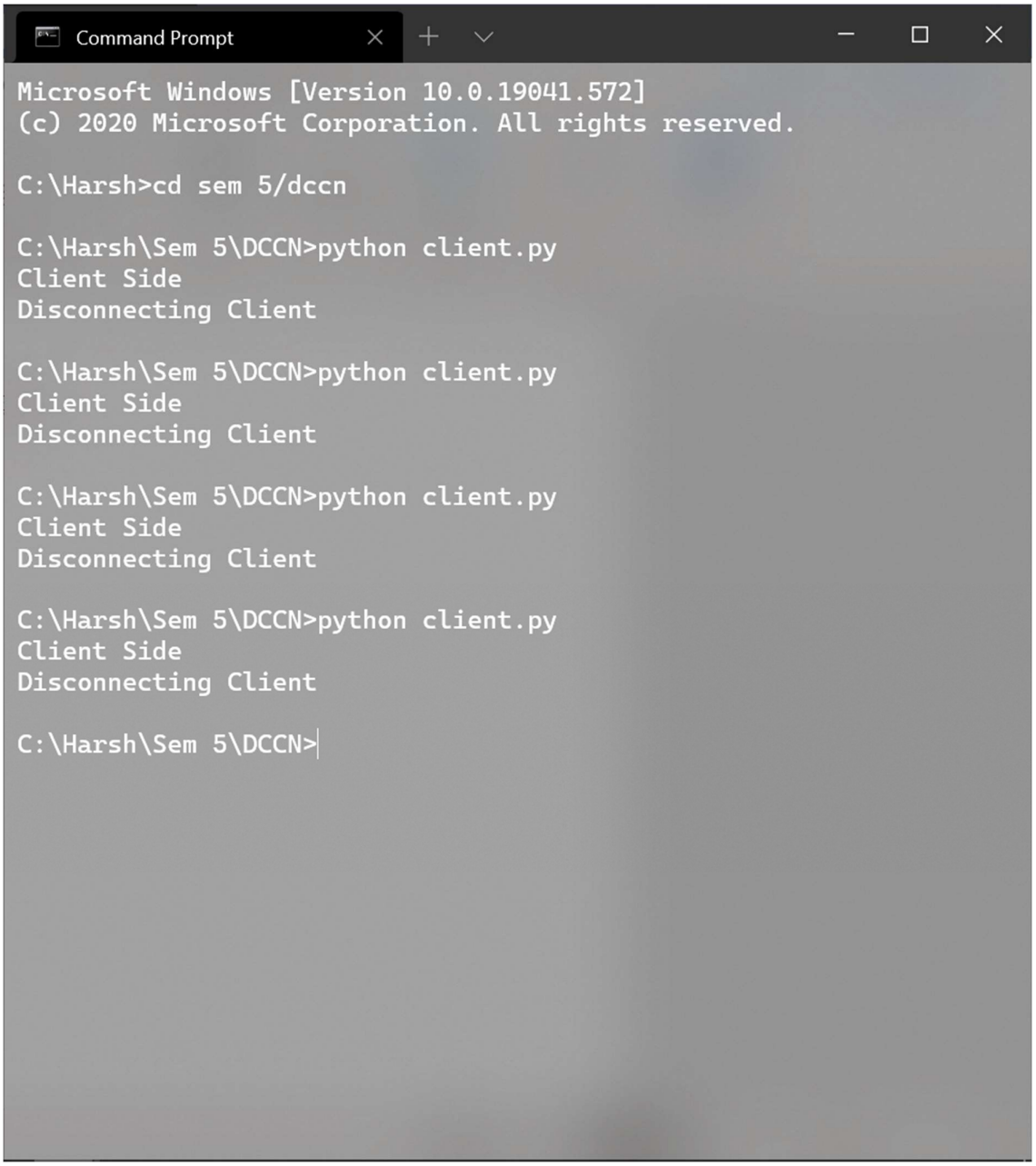
```
import socket
s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
s.connect((socket.gethostname(), 8000))
msg = s.recv(1024)
print(msg.decode('utf-8'))
print("Disconnecting Client")
```

Output:

server.py



client.py



Conclusion:

- I have learnt the basics of socket programming and how to implement a basic client-server model in python.

References:

[1] https://www.geeksforgeeks.org/socket-programming-python/
[2] https://www.tutorialspoint.com/python/python_networking.htm