Name: Namrata Bhorade
TE COMPS
UID: 2019130004

CEL 51, DCCN, Monsoon 2020
Lab 8: Socket Programming

Aim: To implement Socket Programming and establish connection between a client and server.

Theory:

Socket Programming

- Socket programming 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.

Socket creation

```
int sockfd = socket(domain, type, protocol)
```

- **sockfd:** socket descriptor, an integer (like a file-handle)
- **domain:** integer, communication domain e.g., AF_INET (IPv4 protocol) , AF_INET6 (IPv6 protocol)
- **type:** communication type
  SOCK_STREAM: TCP(reliable, connection oriented)
  SOCK_DGRAM: UDP(unreliable, connectionless)
- **protocol:** Protocol value for Internet Protocol(IP), which is 0. This is the same number which appears on protocol field in the IP header of a packet.(man protocols for more details)

Bind

```
int bind(int sockfd, const struct sockaddr *addr, socklen_t addrlen);
```

- After creation of the socket, bind function binds the socket to the address and port number specified in addr(custom data structure).

Listen

```
int listen(int sockfd, int backlog);
```

- It puts the server socket in a passive mode, where it waits for the client to approach the server to make a connection.
- The backlog, defines the maximum length to which the queue of pending connections for sockfd may grow.
- If a connection request arrives when the queue is full, the client may receive an error with an indication of ECONNREFUSED.

Accept

```
int new_socket= accept(int sockfd, struct sockaddr *addr, socklen_t *addrlen);
```

- It extracts the first connection request on the queue of pending connections for the listening socket, sockfd, creates a new connected socket, and returns a new file descriptor referring to that socket.
- At this point, connection is established between client and server, and they are ready to transfer data.

Connect

int connect(int sockfd, const struct sockaddr *addr, socklen_t addrlen);

- The connect() system call connects the socket referred to by the file descriptor sockfd to the address specified by addr. Server's address and port is specified in addr.

Code:

server.py

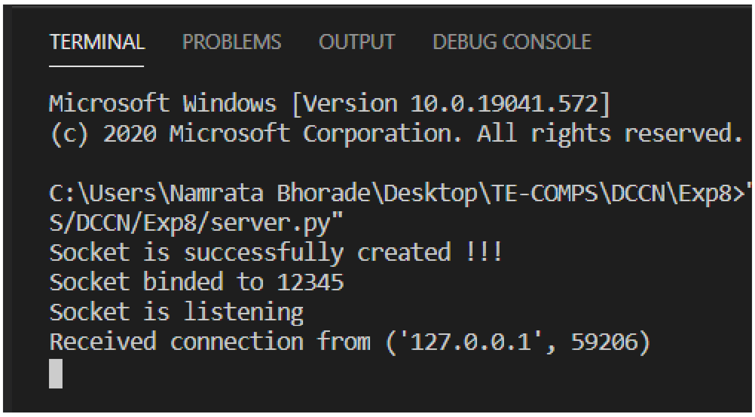
```
import socket

s = socket.socket()
print("Socket is successfully created !!!")

port = 12345

s.bind(('', port))
print(f"Socket binded to {port}")

s.listen(5)
print("Socket is listening")

while True:
    c, address = s.accept()
    print(f"Received connection from {address}")

    c.sendall(b"Thank you for connecting !")
```

client.py

```
import socket

s= socket.socket()
port = 12345

s.connect(('127.0.0.1', port))

print(s.recv(1024))

s.close()
```

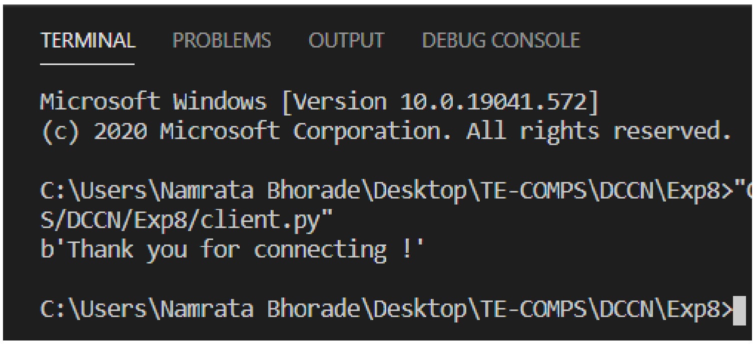
Output:

server.py



```
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Namrata Bhorade\Desktop\TE-COMPS\DCCN\Exp8>'
S/DCCN/Exp8/server.py"
Socket is successfully created !!!
Socket binded to 12345
Socket is listening
Received connection from ('127.0.0.1', 59206)
```

client.py



```
Microsoft Windows [Version 10.0.19041.572]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\Namrata Bhorade\Desktop\TE-COMPS\DCCN\Exp8>"
S/DCCN/Exp8/client.py"
b'Thank you for connecting !'

C:\Users\Namrata Bhorade\Desktop\TE-COMPS\DCCN\Exp8>
```

Conclusion:

1. In this experiment, I learned about Socket Programming.
2. I established connection between server and client by socket programming.

References:

1. Socket Programming
   https://www.geeksforgeeks.org/socket-programming-cc/