

Name: Prachi Mehta

UID: 2018130025

Date: 27/10/2020

CEL 51, DCCN, Monsoon 2020

Lab 8: Socket Programming

**Aim: To implement Client Server program.**

**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.

They are the real backbones behind web browsing. In simpler terms there is a server and a client.

Socket programming is started by importing the socket library and making a simple socket.

**Server Socket Methods:**

Sr.No.	Method & Description
1	<b>s.bind()</b> This method binds address (hostname, port number pair) to socket.
2	<b>s.listen()</b> This method sets up and start TCP listener.
3	<b>s.accept()</b> This passively accept TCP client connection, waiting until connection arrives (blocking).

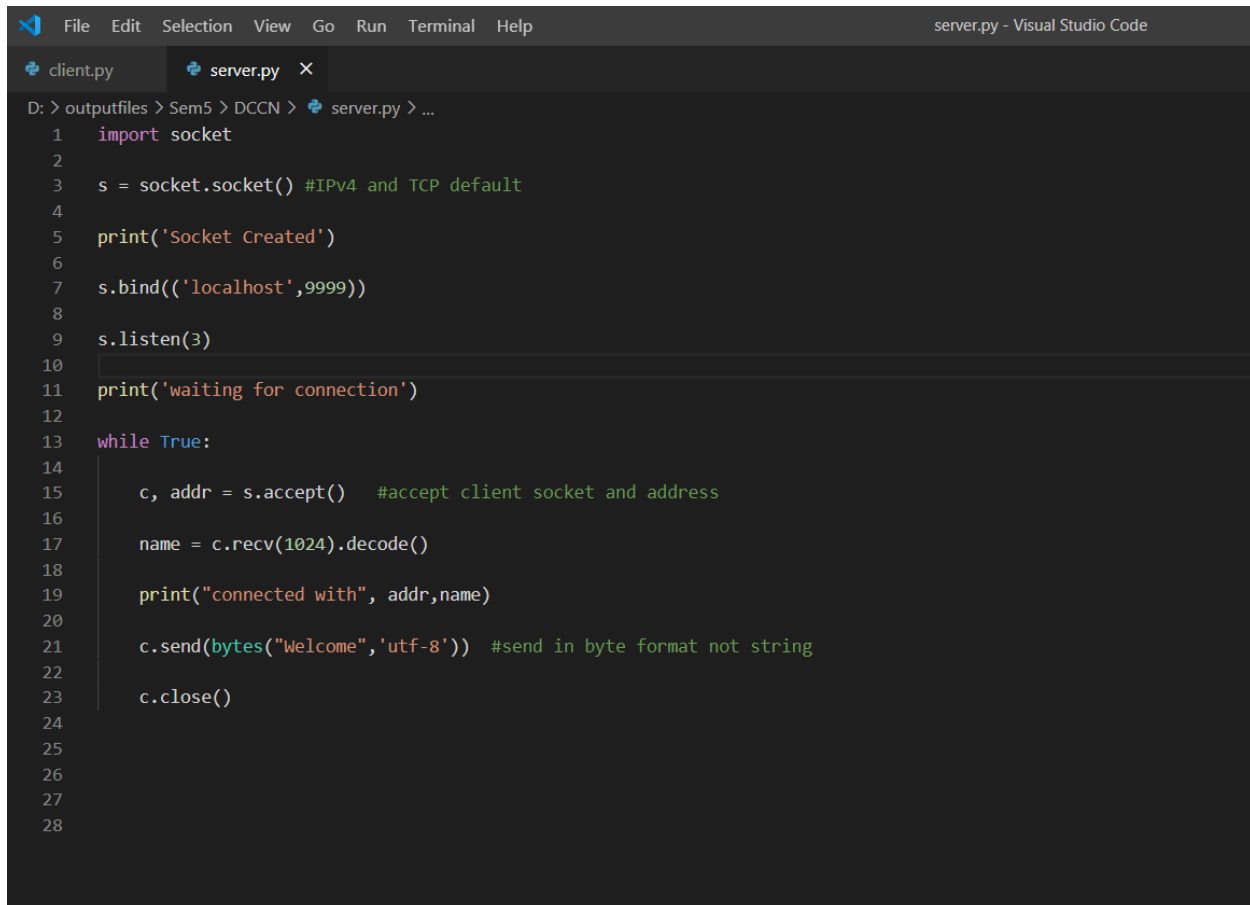
### Client Socket Methods:

Sr.No.	Method & Description
1	<b>s.connect()</b> This method actively initiates TCP server connection.

### General Socket Methods:

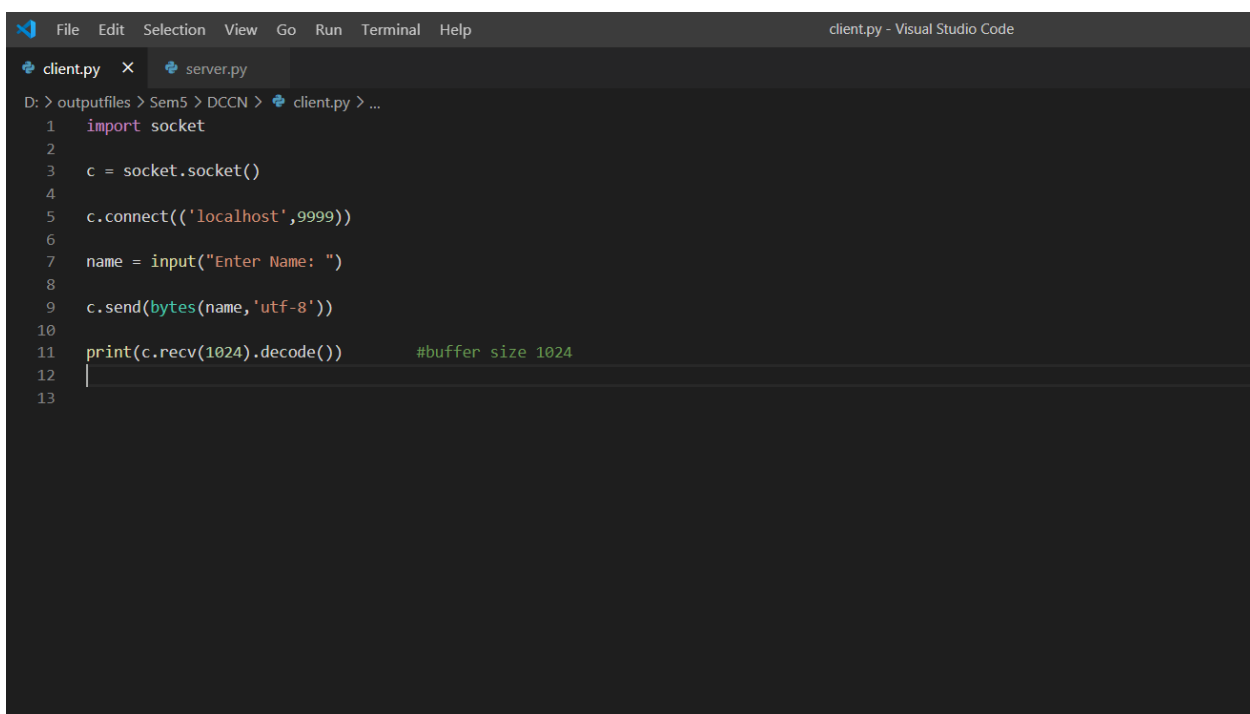
Sr.No.	Method & Description
1	<b>s.recv()</b> This method receives TCP message
2	<b>s.send()</b> This method transmits TCP message
3	<b>s.recvfrom()</b> This method receives UDP message
4	<b>s.sendto()</b> This method transmits UDP message
5	<b>s.close()</b> This method closes socket
6	<b>socket.gethostname()</b> Returns the hostname.

### Server:



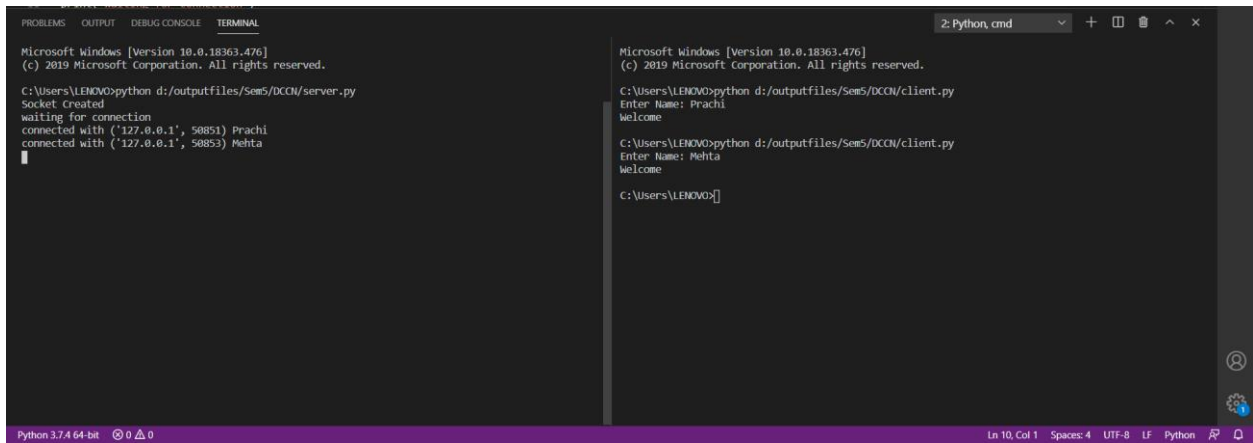
```
server.py
1  import socket
2
3  s = socket.socket() #IPv4 and TCP default
4
5  print('Socket Created')
6
7  s.bind(('localhost',9999))
8
9  s.listen(3)
10
11 print('waiting for connection')
12
13 while True:
14     c, addr = s.accept() #accept client socket and address
15
16     name = c.recv(1024).decode()
17
18     print("connected with", addr,name)
19
20     c.send(bytes("Welcome",'utf-8')) #send in byte format not string
21
22     c.close()
23
24
25
26
27
28
```

## Client:



```
client.py
1  import socket
2
3  c = socket.socket()
4
5  c.connect(('localhost',9999))
6
7  name = input("Enter Name: ")
8
9  c.send(bytes(name,'utf-8'))
10
11 print(c.recv(1024).decode()) #buffer size 1024
12
13
```

## Output:



The image shows two side-by-side terminal windows from a Windows environment. The left window is titled 'Python cmd' and shows the execution of a Python script `d:/outputfiles/Sem5/DCCN/server.py`. The output indicates that a socket was created and the program is waiting for a connection. It then shows two successful connections: one from `('127.0.0.1', 50851)` with the name 'Prachi' and another from `('127.0.0.1', 50853)` with the name 'Mehta'. The right window is also titled 'Python cmd' and shows the execution of `d:/outputfiles/Sem5/DCCN/client.py`. It prompts for a name, and the user enters 'Prachi', followed by 'Mehta'. Both entries are acknowledged with a 'Welcome' message. The status bar at the bottom of the left window indicates 'Python 3.7.4 64-bit' and the status bar of the right window shows 'Ln 10, Col 1' and 'Spaces: 4'.

```
Microsoft Windows [Version 10.0.18363.476]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\LENOVO>python d:/outputfiles/Sem5/DCCN/server.py
Socket Created
waiting for connection
connected with ('127.0.0.1', 50851) Prachi
connected with ('127.0.0.1', 50853) Mehta
█

Python 3.7.4 64-bit 0 0 0
```

```
Microsoft Windows [Version 10.0.18363.476]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\LENOVO>python d:/outputfiles/Sem5/DCCN/client.py
Enter Name: Prachi
Welcome

C:\Users\LENOVO>python d:/outputfiles/Sem5/DCCN/client.py
Enter Name: Mehta
Welcome

C:\Users\LENOVO>
```

Ln 10, Col 1 Spaces: 4 UTF-8 LF Python

**Conclusion:** After completing this experiment I understood concept of socket programming.

**Reference:**

1. [https://www.tutorialspoint.com/python/python\\_networking.html](https://www.tutorialspoint.com/python/python_networking.html)