**Name: Vishal S. Salvi**

**UID:2019230069**

**Batch – C**

**Aim: Implementing** **Client server programming**

**Theory:**

**What is Sockets?[1]**

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.

Sockets may be implemented over a number of different channel types: Unix domain sockets, TCP, UDP, and so on. The *socket* library provides specific classes for handling the common transports as well as a generic interface for handling the rest.

## Client[1]

client program which opens a connection to a given port 12345 and given host. This is very simple to create a socket client using Python's *socket* module function.

The **socket.connect(hosname, port )** opens a TCP connection to *hostname* on the *port*. Once you have a socket open, you can read from it like any IO object. When done, remember to close it, as you would close a file.

The following code is a very simple client that connects to a given host and port, reads any available data from the socket, and then exits −

**Client :**

import socket *# Import socket module*s = socket.socket() *# Create a socket object*host = socket.gethostname() *# Get local machine name*port = 12345 *# Reserve a port for your service.*s.connect((host, port))  
print (s.recv(8000))

## Server[1]

To write Internet servers, we use the **socket** function available in socket module to create a socket object. A socket object is then used to call other functions to setup a socket server.

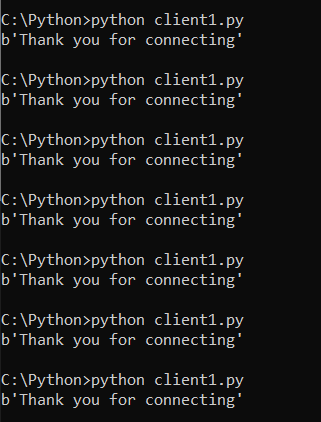
Now call **bind(hostname, port)** function to specify a *port* for your service on the given host.

Next, call the *accept* method of the returned object. This method waits until a client connects to the port you specified, and then returns a *connection* object that represents the connection to that client.

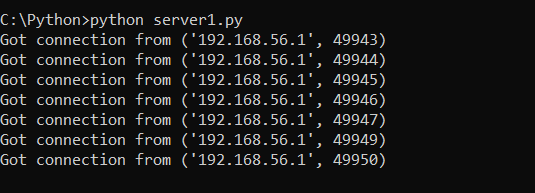
**Server:**

import socket *# Import socket module*s = socket.socket() *# Create a socket object*host = socket.gethostname() *# Get local machine name*port = 8000 *# Reserve a port for your service.*s.bind((host, port)) *# Bind to the port*while True:  
 c, addr = s.accept() *# Establish connection with client.* print (**'Got connection from'**, addr)  
 c.send((bytes(**'Thank you for connecting'**,**'utf-8'**)))

**Client:**



**Server:**



**Conclusion:**

Thus, from this experiment I successfully completed connection between client and server

**Reference:**

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