

# **Client Server Socket Programming**

## **(One Client Program and four Server Program)**

**MOHAMMAD SHOAIB ANSARI**  
**BT21CSE063**  
**COMPUTER NETWORKS**

### Usage:

Use the following command to run each server script:

```
python3 server1.py <SERVER_IP> <SERVER_PORT>
```

Replace <SERVER\_IP> with the IP address on which you want the server to listen (e.g., 127.0.0.1), and <SERVER\_PORT> with the port number you want to use for communication (e.g., 5000).

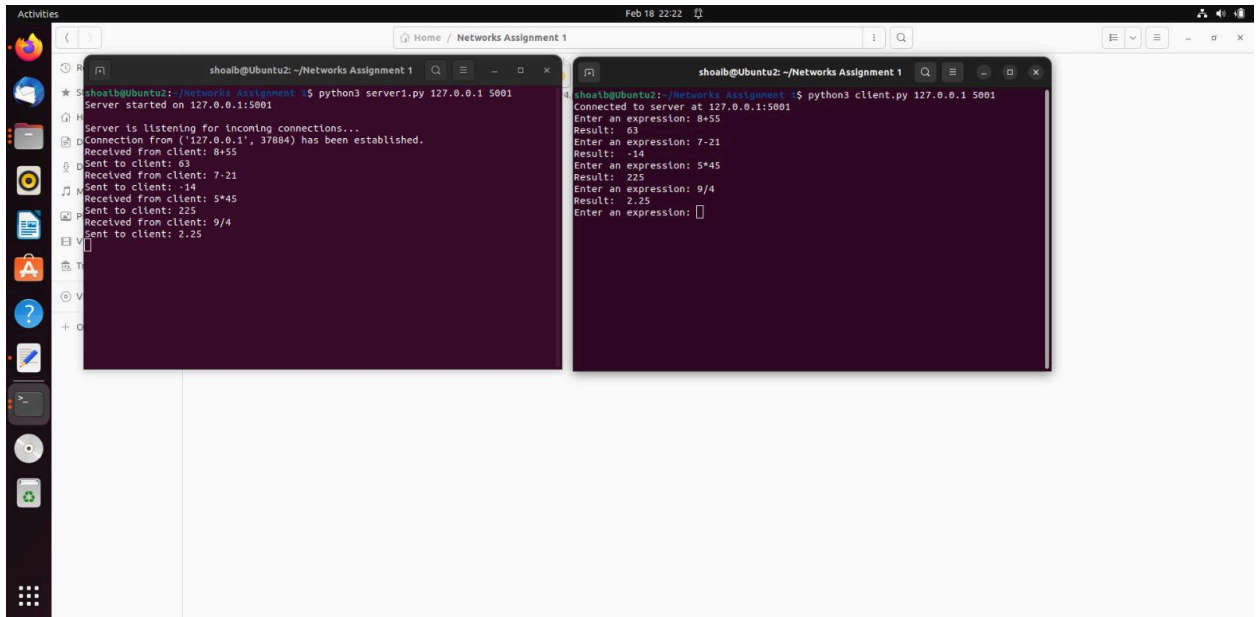
Use the following command to run the client script:

```
python3 client.py <SERVER_IP> <SERVER_PORT>
```

Replace <SERVER\_IP> with the IP address of the server you want to connect to, and <SERVER\_PORT> with the port number on which the server is listening.

### Testing:

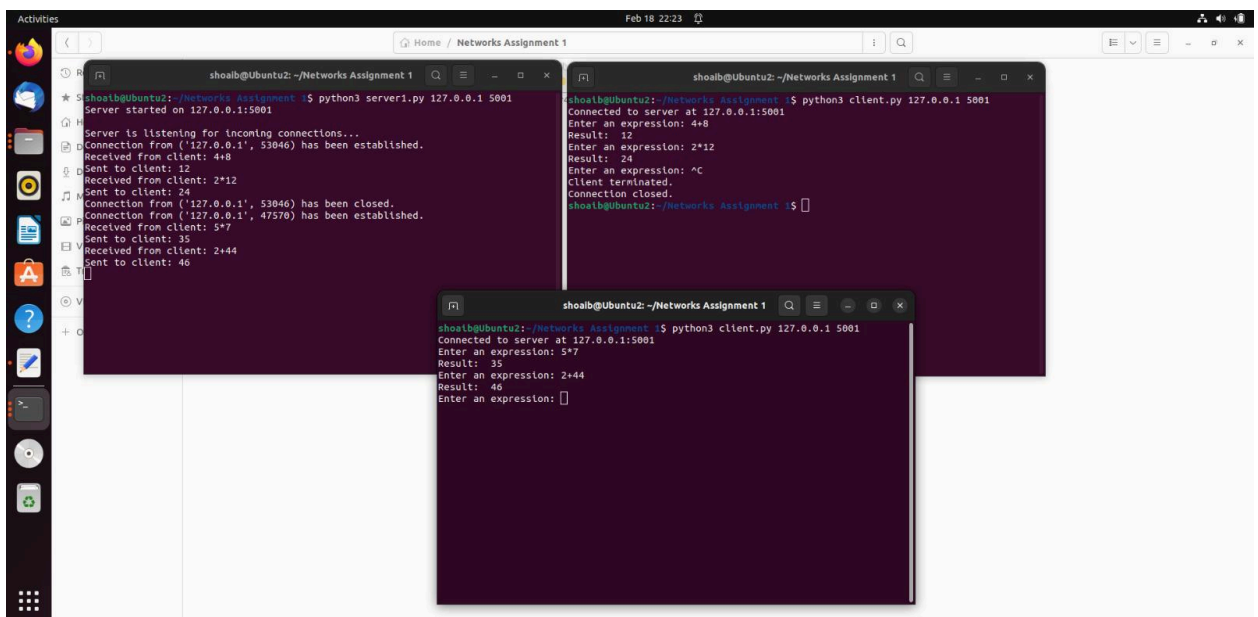
- [TEST1]: For server 1, we will start a single client, connect to server, and test all 4 arithmetic operations (+, -, \*, /) with two operands each. We will check that the results returned by the server to the client are correct.



```
shoalb@Ubuntu2: ~/Networks Assignment 1
Server started on 127.0.0.1:5001
Server is listening for incoming connections...
Connection from ('127.0.0.1', 37884) has been established.
Received from client: 8+55
Sent to client: 63
Received from client: 7-21
Sent to client: -14
Received from client: 5*45
Sent to client: 225
Received from client: 9/4
Sent to client: 2.25

shoalb@Ubuntu2: ~/Networks Assignment 1
Connected to server at 127.0.0.1:5001
Enter an expression: 8+55
Result: 63
Enter an expression: 7-21
Result: -14
Enter an expression: 5*45
Result: 225
Enter an expression: 9/4
Result: 2.25
Enter an expression: 
```

- [TEST2]: For server1, we will start a client, do some math operations (like TEST1), then terminate the client, start a second client, and check that the second client can chat with the server as well.

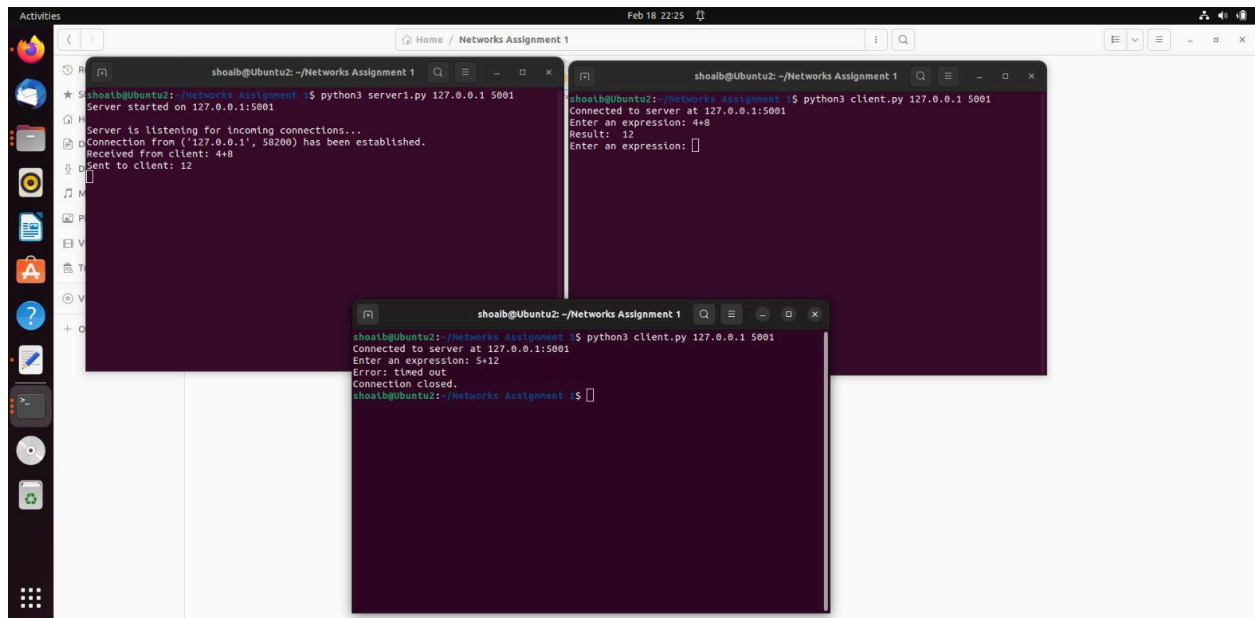


```
shoalb@Ubuntu2: ~/Networks Assignment 1
Server started on 127.0.0.1:5001
Server is listening for incoming connections...
Connection from ('127.0.0.1', 53046) has been established.
Received from client: 4+8
Sent to client: 12
Received from client: 2*12
Sent to client: 24
Connection from ('127.0.0.1', 53046) has been closed.
Connection from ('127.0.0.1', 47570) has been established.
Received from client: 5*7
Sent to client: 35
Received from client: 2+44
Sent to client: 46

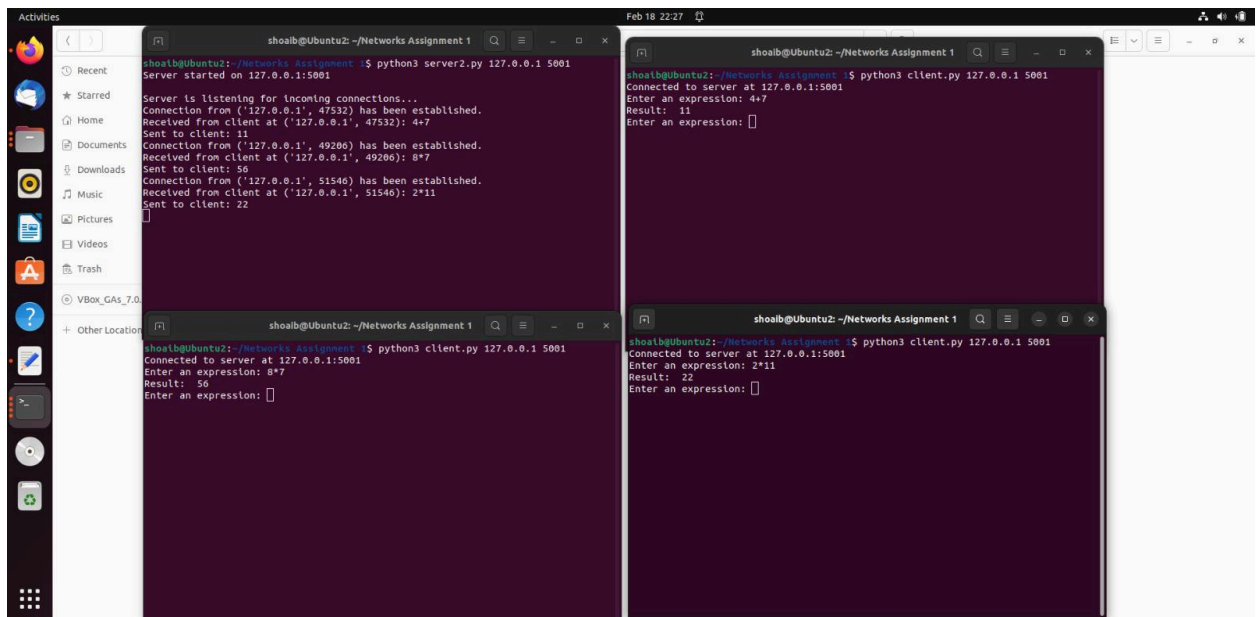
shoalb@Ubuntu2: ~/Networks Assignment 1
Connected to server at 127.0.0.1:5001
Enter an expression: 4+8
Result: 12
Enter an expression: 2*12
Result: 24
Enter an expression: ^C
Client terminated.
Connection closed.
shoalb@Ubuntu2: ~/Networks Assignment 1

shoalb@Ubuntu2: ~/Networks Assignment 1
Connected to server at 127.0.0.1:5001
Enter an expression: 5*7
Result: 35
Enter an expression: 2+44
Result: 46
Enter an expression: 
```

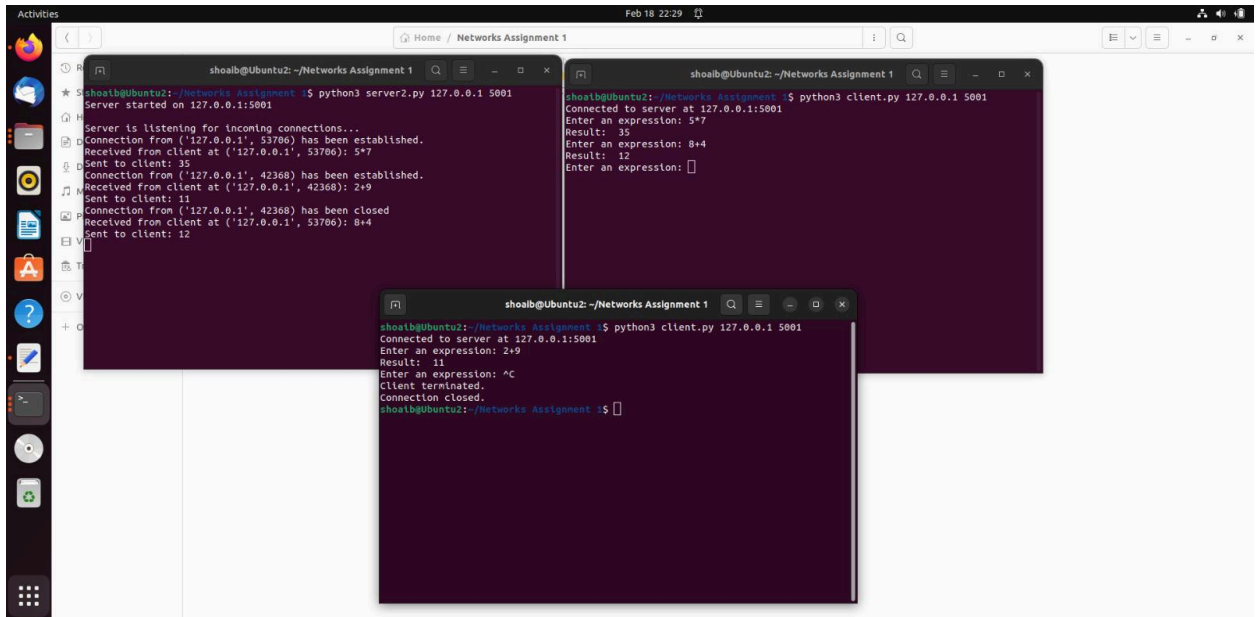
- [TEST3]: For server 1, we will try to connect a second client when the first one is still connected, and check that its socket operations fail.



- [TEST4]: For server2, we will test that multiple clients can simultaneously connect and chat with the server correctly.



- [TEST6]: For server2, we will connect a client, then connect and disconnect a second client. The first client should continue to function correctly.

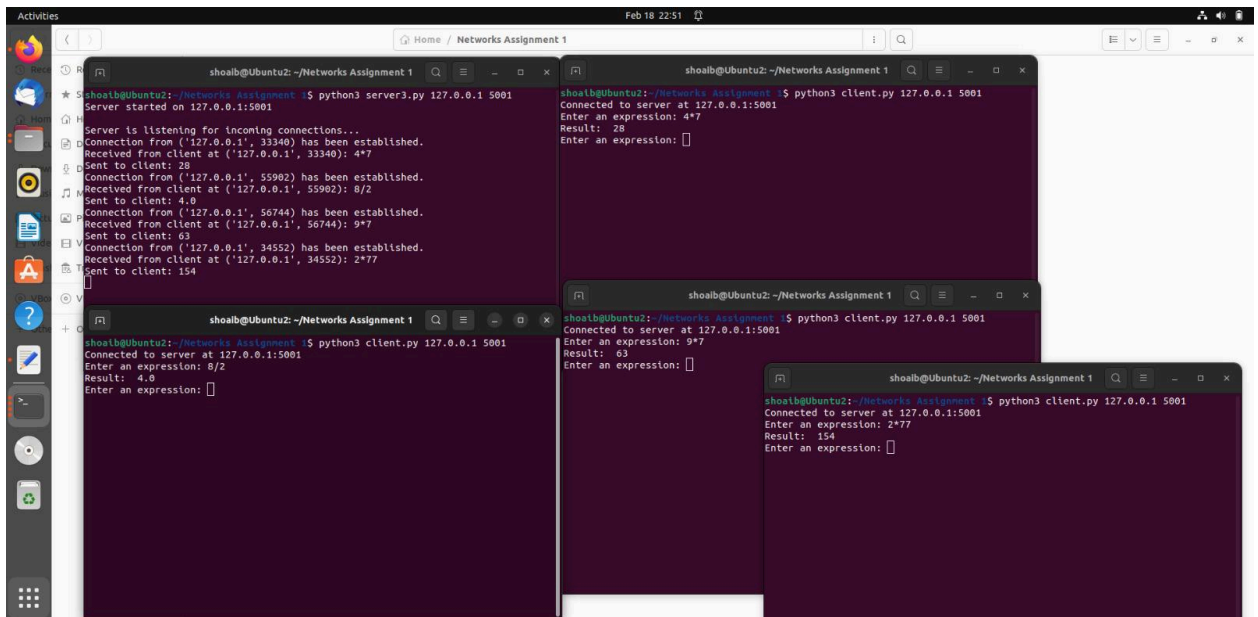


```
shoalb@Ubuntu2: ~/Networks Assignment 1
shoalb@Ubuntu2:~/Networks Assignment 1 $ python3 server2.py 127.0.0.1 5001
Server started on 127.0.0.1:5001
Server is listening for incoming connections...
Connection from ('127.0.0.1', 53706) has been established.
Received from client at ('127.0.0.1', 53706): 5*7
Sent to client: 35
Connection from ('127.0.0.1', 42368) has been established.
Received from client at ('127.0.0.1', 42368): 2+9
Sent to client: 11
Connection from ('127.0.0.1', 42368) has been closed
Received from client at ('127.0.0.1', 53706): 8+4
Sent to client: 12

shoalb@Ubuntu2:~/Networks Assignment 1 $ python3 client.py 127.0.0.1 5001
Connected to server at 127.0.0.1:5001
Enter an expression: 5*7
Result: 35
Enter an expression: 8+4
Result: 12
Enter an expression: 

```

- [TEST7] For server3, we will test that multiple clients can simultaneously connect and chat with the server correctly.



```
shoalb@Ubuntu2:~/Networks Assignment 1
shoalb@Ubuntu2:~/Networks Assignment 1 $ python3 server3.py 127.0.0.1 5001
Server started on 127.0.0.1:5001
Server is listening for incoming connections...
Connection from ('127.0.0.1', 33340) has been established.
Received from client at ('127.0.0.1', 33340): 4*7
Sent to client: 28
Connection from ('127.0.0.1', 55902) has been established.
Received from client at ('127.0.0.1', 55902): 8/2
Sent to client: 4.0
Connection from ('127.0.0.1', 56744) has been established.
Received from client at ('127.0.0.1', 56744): 9*7
Sent to client: 63
Connection from ('127.0.0.1', 34552) has been established.
Received from client at ('127.0.0.1', 34552): 2*77
Sent to client: 154

shoalb@Ubuntu2:~/Networks Assignment 1 $ python3 client.py 127.0.0.1 5001
Connected to server at 127.0.0.1:5001
Enter an expression: 8/2
Result: 4.0
Enter an expression: 

shoalb@Ubuntu2:~/Networks Assignment 1 $ python3 client.py 127.0.0.1 5001
Connected to server at 127.0.0.1:5001
Enter an expression: 9*7
Result: 63
Enter an expression: 

shoalb@Ubuntu2:~/Networks Assignment 1 $ python3 client.py 127.0.0.1 5001
Connected to server at 127.0.0.1:5001
Enter an expression: 2*77
Result: 154
Enter an expression: 

```

- [TEST8] For server3, we will connect a client, then connect and disconnect a second client. The first client should continue to function correctly.

The screenshot shows a terminal window titled "shoalb@Ubuntu2: ~/Networks Assignment 1". The server is running `python3 server3.py 127.0.0.1 5001`. It logs incoming connections from `127.0.0.1` at ports `34148` and `55648`. The server sends data to the client and receives responses. The client is running `python3 client.py 127.0.0.1 5001` and sends expressions like `7*11`, `25*5`, and `125`, which the server evaluates and returns.

- [TEST9] For server 4, we will test that multiple clients can simultaneously connect and chat with the server correctly.

The screenshot shows a terminal window titled "shoalb@Ubuntu2: ~/Networks Assignment 1". The server is running `python3 server4.py 127.0.0.1 5001`. It logs incoming connections from `127.0.0.1` at ports `34020`, `41050`, `41962`, and `55834`. The server receives data from multiple clients and sends responses. The clients are running `python3 client.py 127.0.0.1 5001` and send expressions like `client1`, `hi this is client2`, `hi this is client3 !!`, and `hi client4 here`, which the server echoes back.