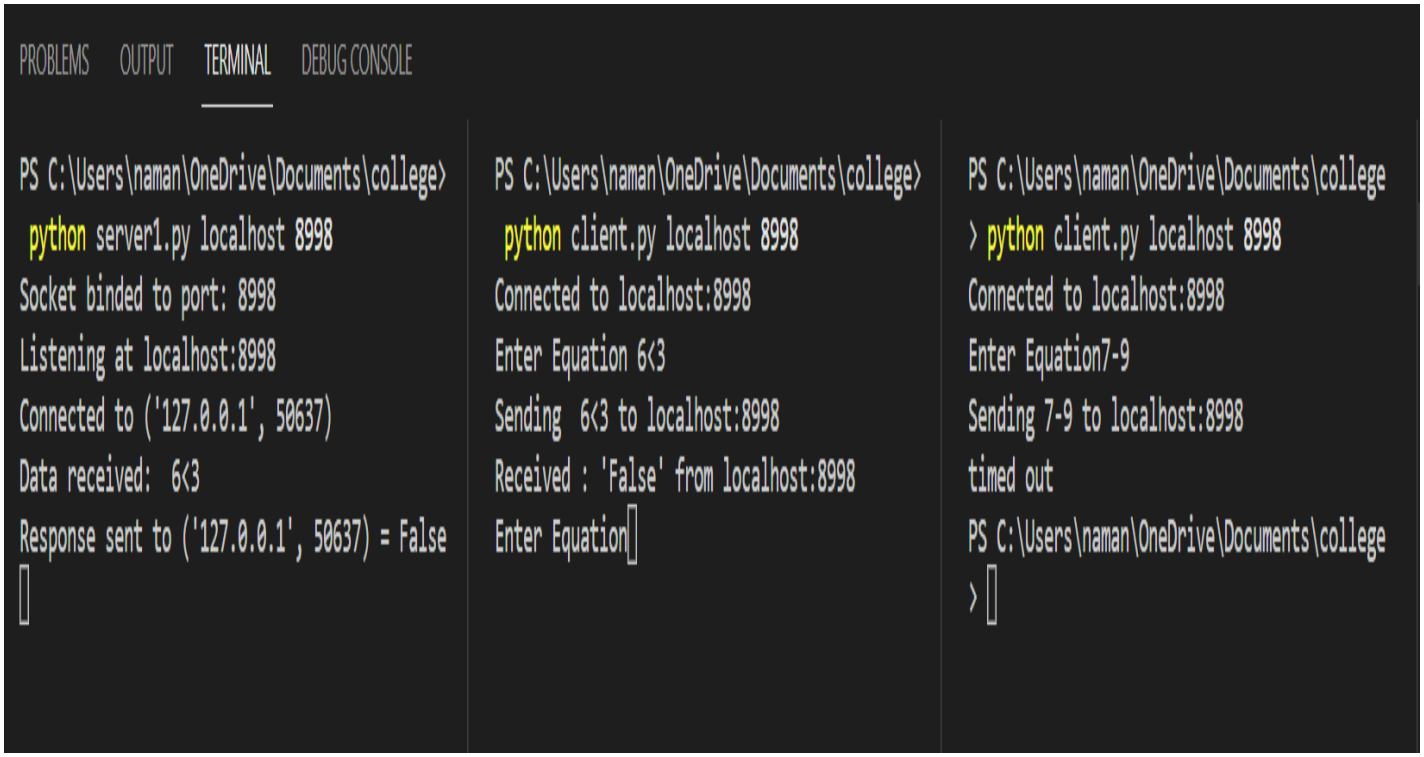
ASSIGNMENT-2
Client Server Socket Programming

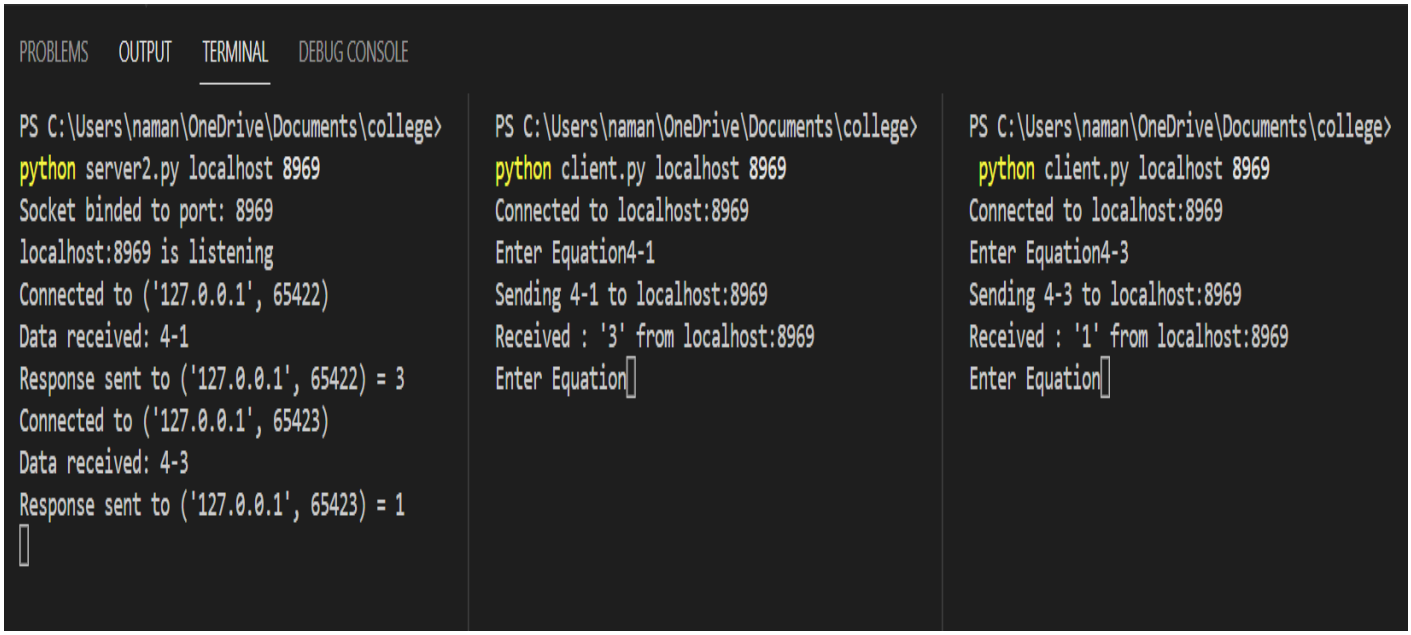
- Part1: Server handling single connection at a time
- Part2: Server handling multiple connections simultaneously
- The sockets must use TCP and IPV4 protocols.
- All the messages of connection details and information passed should be displayed on the command line with respect to servers and clients.
- Python3 must be installed.

- How To Run :
  - Each of the server files can be run using:
    -> python serverx.py "host" "port"
  - Client files can be run using:
    -> python client.py "host" "port"

1. Your server program "server1.py " will be a single process server that can handle only one client at a time. If a second client tries to chat with the server while some other client's session is already in progress, the second client's socket operations should see an error. After the first client closes the connection, the server should then accept connections from the other client.
server1.py :

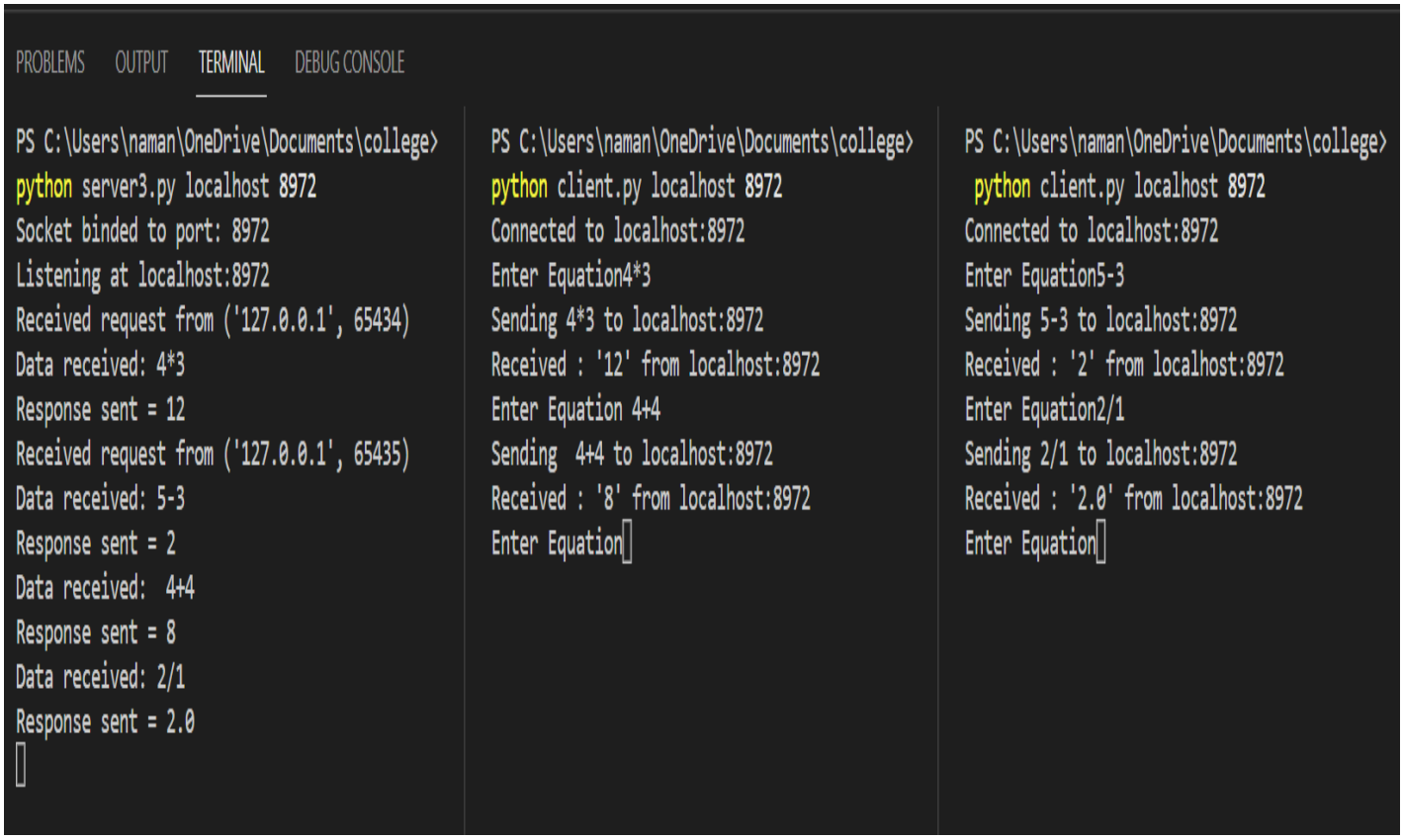


2. Your server program "server2.py " will be a multi-threaded server that will create a new thread for every new client request it receives. Multiple clients should be able to simultaneously chat with the server
server2.py :

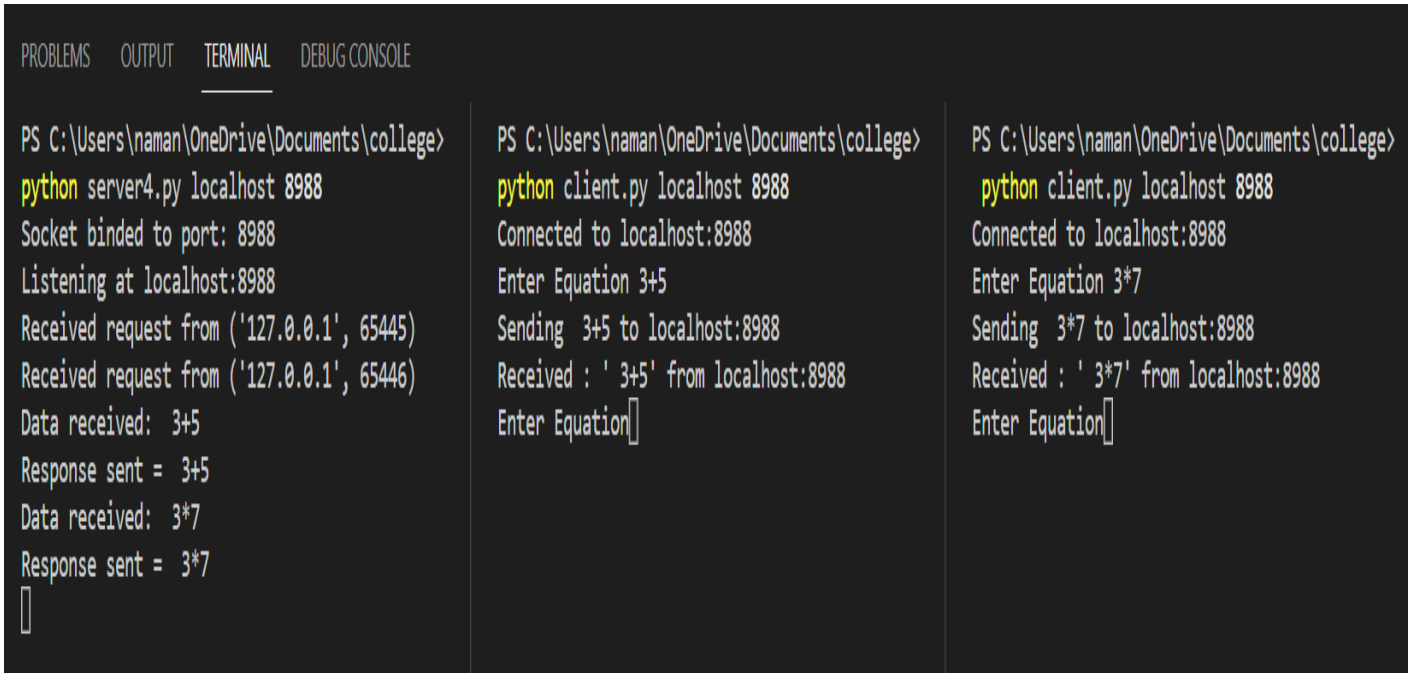


Each connection spawns its own thread and engages the client.
Connection is closed when the client closes the connection.

3. Your server program "server3.py " will be a single process server that uses the "select" method to handle multiple clients concurrently
server3.py :

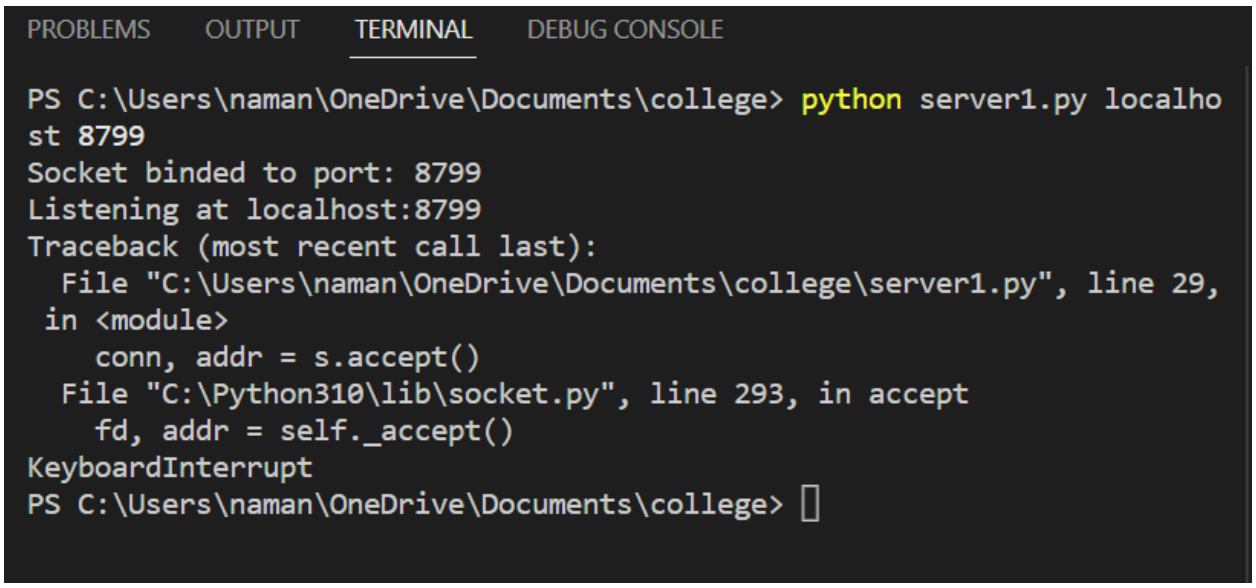


4. Your server program "server4.py" will be an echo server (that replies the same message to the client that was received from the same client); it will be a single process server that uses the "select" method to handle multiple clients concurrently
server4.py :

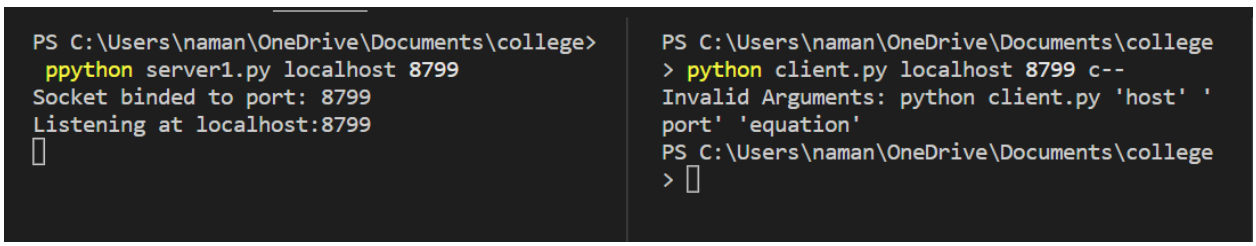


Edge cases:

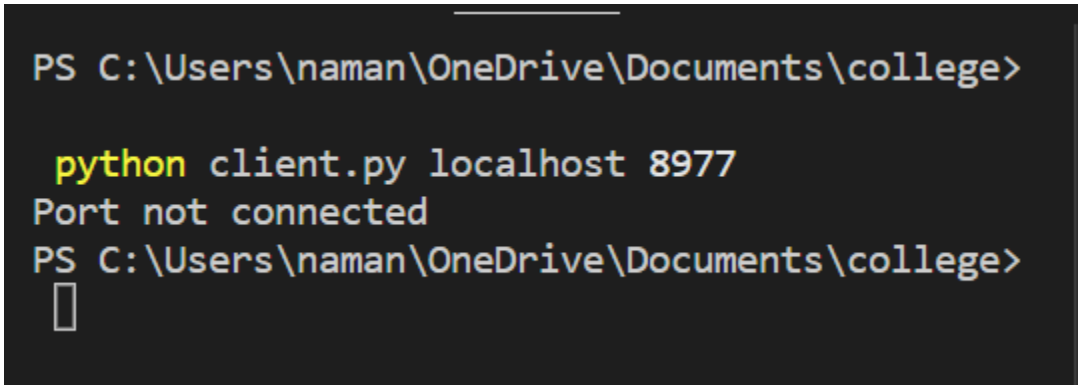
Keyboard interrupt ctrl+c



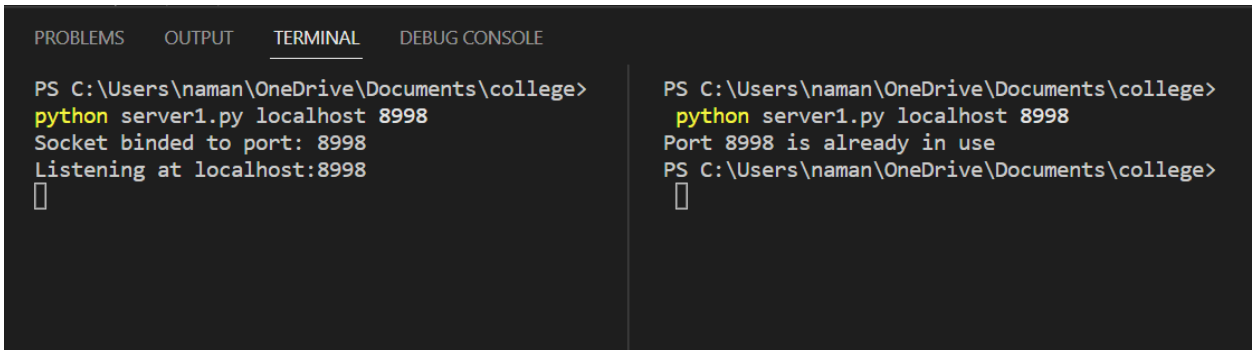
Invalid argument



Connecting to an unopened port



Connecting to an already opened port



THANK YOU,
Naman Jain
BT20CSE087