

To run both the client-server programs on VS code, see below :

Single Process Server :

Output:

Server:

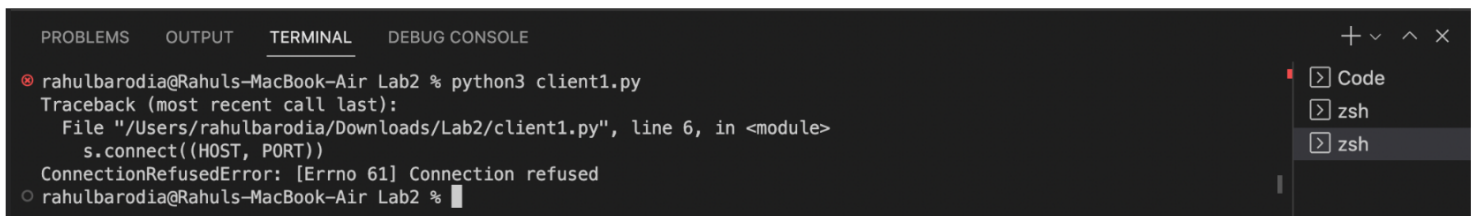


```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
rahulbarodia@Rahuls-MacBook-Air Lab2 % python3 server1.py
Socket is listening..
```

Client:



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
rahulbarodia@Rahuls-MacBook-Air Lab2 % python3 client1.py
Enter the expression:8-3
5
rahulbarodia@Rahuls-MacBook-Air Lab2 %
```



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
rahulbarodia@Rahuls-MacBook-Air Lab2 % python3 client1.py
Traceback (most recent call last):
  File "/Users/rahulbarodia/Downloads/Lab2/client1.py", line 6, in <module>
    s.connect((HOST, PORT))
ConnectionRefusedError: [Errno 61] Connection refused
rahulbarodia@Rahuls-MacBook-Air Lab2 %
```

As we can see from the above images that when a 2nd client tries to access the server, the connection is refused. Thus server1 can handle only a single client at a time.

Multiprocess Server :

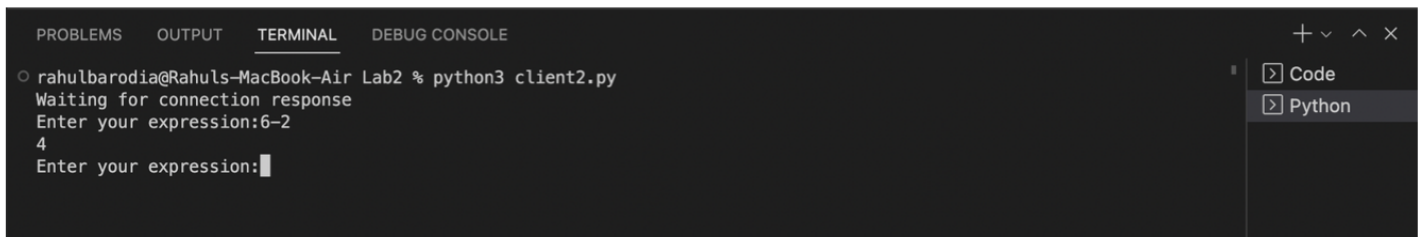
Output:

Server:

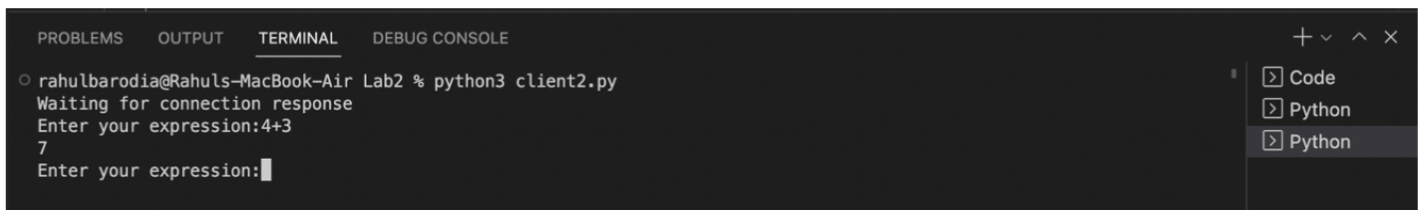


```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
rahulbarodia@Rahuls-MacBook-Air Lab2 % python3 server2.py
Socket is listening..
```

Client:



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
rahulbarodia@Rahuls-MacBook-Air Lab2 % python3 client2.py
Waiting for connection response
Enter your expression:6-2
4
Enter your expression:
```



```
PROBLEMS OUTPUT TERMINAL DEBUG CONSOLE
rahulbarodia@Rahuls-MacBook-Air Lab2 % python3 client2.py
Waiting for connection response
Enter your expression:4+3
7
Enter your expression:
```

As we can see from the above images, server2 allows multiple clients to access at the same time. The above example shows two clients accessing server2 and getting the correct response.