# Computer Network Lab (CS358)

## Assignment 3 - Client Server Calculator

# Report -

# Saurav Dudhate

# 2001CS62

# How to Run?

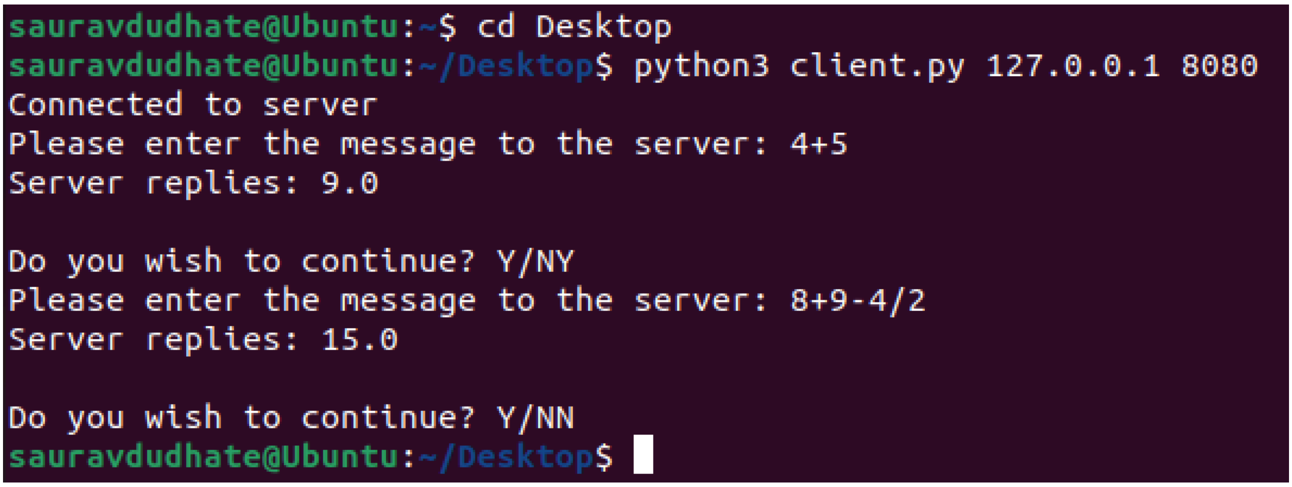
1. Open command prompt and set the directory to where your file exists.
2. Run the file using python3 command and specify the command line arguments which will be IpV4 address and port number
3. Do the same thing for both server as well as client.
4. In case of clients, enter the expression you want to compute without any spaces or brackets. For example, input can be of the format “4+5”,”5+6+7”,”4*5-7/2” but not “4 + 5”,”8+(7-3)”
5. Your expression will be computed and results will be shown in case your expression is according to the above specifies format, else an error message will be shown that states the expression is invalid.
6. The client will then be asked if he wants to continue. If yes, he/she will be able to do further calculations else program for client will be terminated.
7. Program for server can only be close with a keyboard interruption such as “Ctrl+C”

# Output

## Server 1:

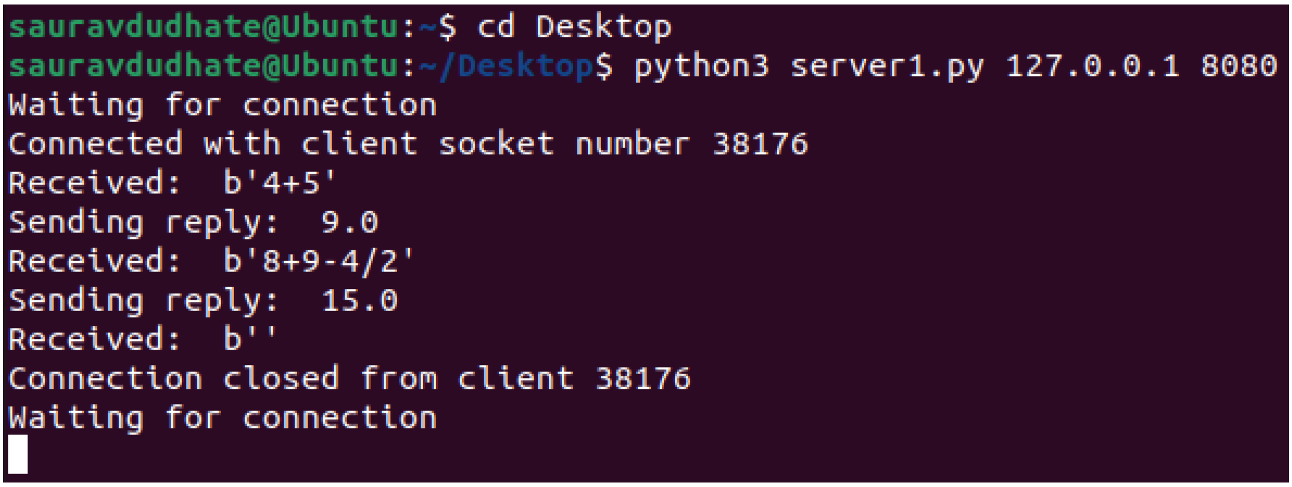
**Inputs : “4+5”, “8+9-4/2”**
**Outputs: “9.0” , “15.0”**

Client:



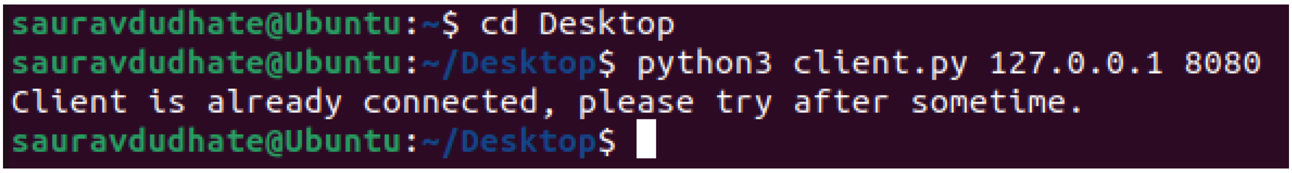
```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 client.py 127.0.0.1 8080
Connected to server
Please enter the message to the server: 4+5
Server replies: 9.0

Do you wish to continue? Y/NY
Please enter the message to the server: 8+9-4/2
Server replies: 15.0

Do you wish to continue? Y/NN
sauravdudhate@Ubuntu:~/Desktop$ 
```

Server:



```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 server1.py 127.0.0.1 8080
Waiting for connection
Connected with client socket number 38176
Received:  b'4+5'
Sending reply:  9.0
Received:  b'8+9-4/2'
Sending reply:  15.0
Received:  b''
Connection closed from client 38176
Waiting for connection
```

**Error thrown to a client when a server is already connected to some other client:**

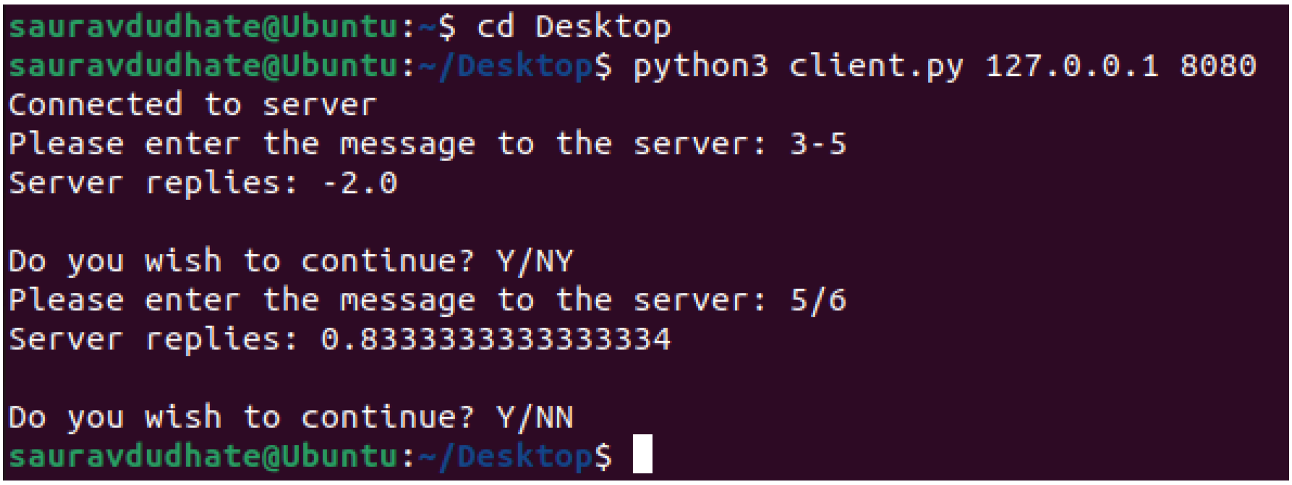


```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 client.py 127.0.0.1 8080
Client is already connected, please try after sometime.
sauravdudhate@Ubuntu:~/Desktop$ 
```

## Server 2:

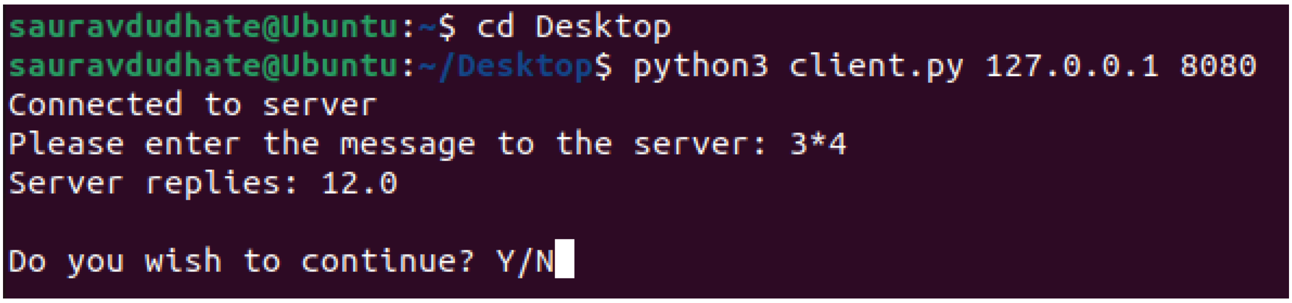
**Inputs: “3-5” (Client1) -> “3*4” (Client2) -> “5/6” (Client1)**
**Outputs: “-2.0” (Client1) -> “0.8333333333334” (Client1) -> “12.0” (Client2)**

Client1:



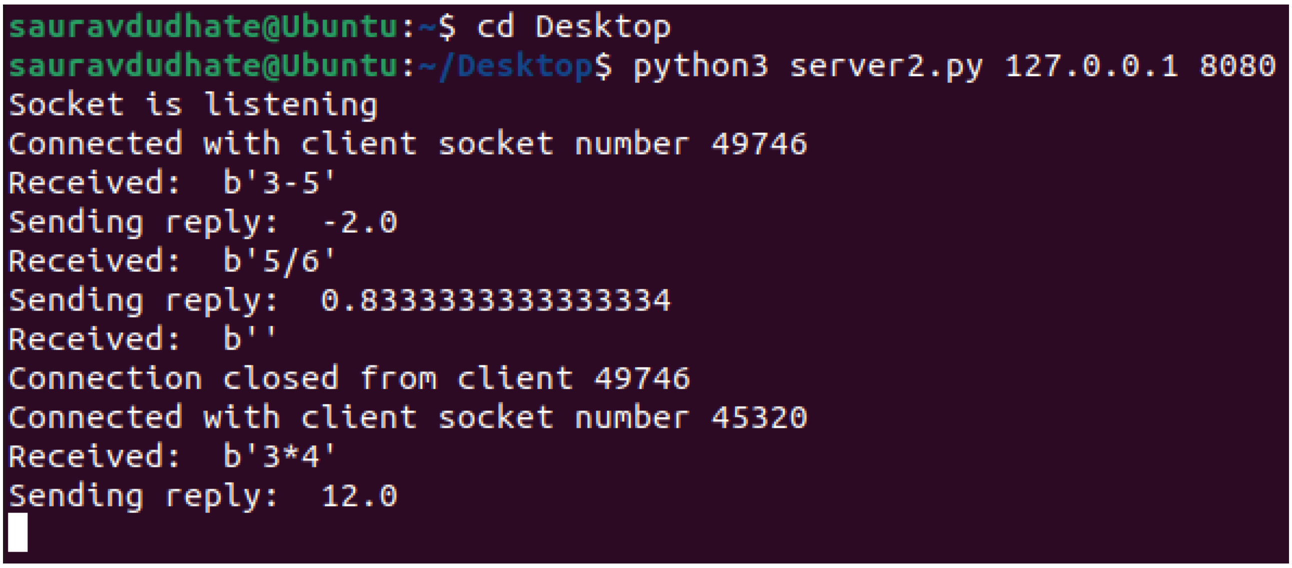
```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 client.py 127.0.0.1 8080
Connected to server
Please enter the message to the server: 3-5
Server replies: -2.0

Do you wish to continue? Y/NY
Please enter the message to the server: 5/6
Server replies: 0.8333333333333334

Do you wish to continue? Y/NN
sauravdudhate@Ubuntu:~/Desktop$ 
```

Client2:



```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 client.py 127.0.0.1 8080
Connected to server
Please enter the message to the server: 3*4
Server replies: 12.0

Do you wish to continue? Y/N
```

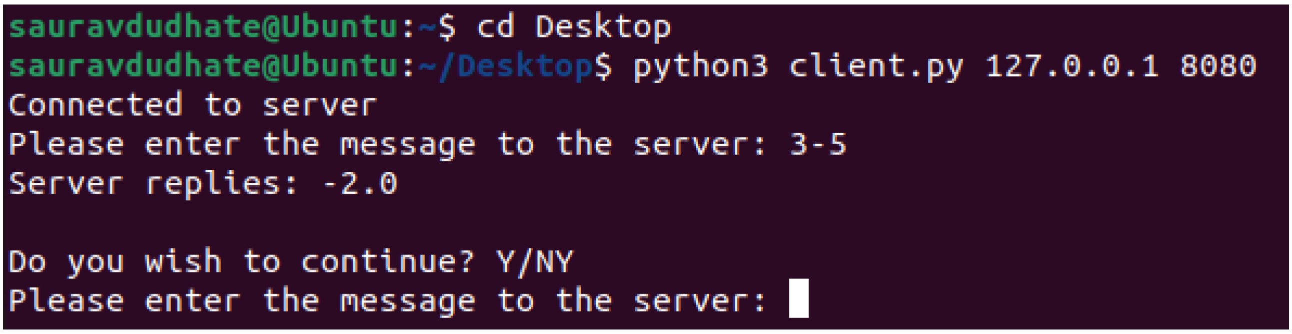
Server:



```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 server2.py 127.0.0.1 8080
Socket is listening
Connected with client socket number 49746
Received:  b'3-5'
Sending reply:  -2.0
Received:  b'5/6'
Sending reply:  0.8333333333333334
Received:  b''
Connection closed from client 49746
Connected with client socket number 45320
Received:  b'3*4'
Sending reply:  12.0
```

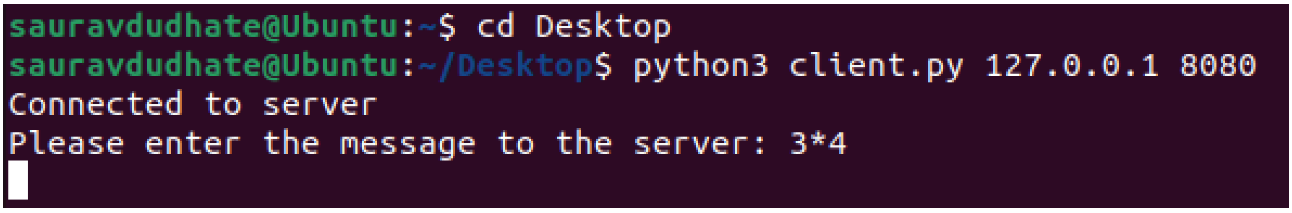
**Client2 is displayed result only after client1 exits**:

Client1:



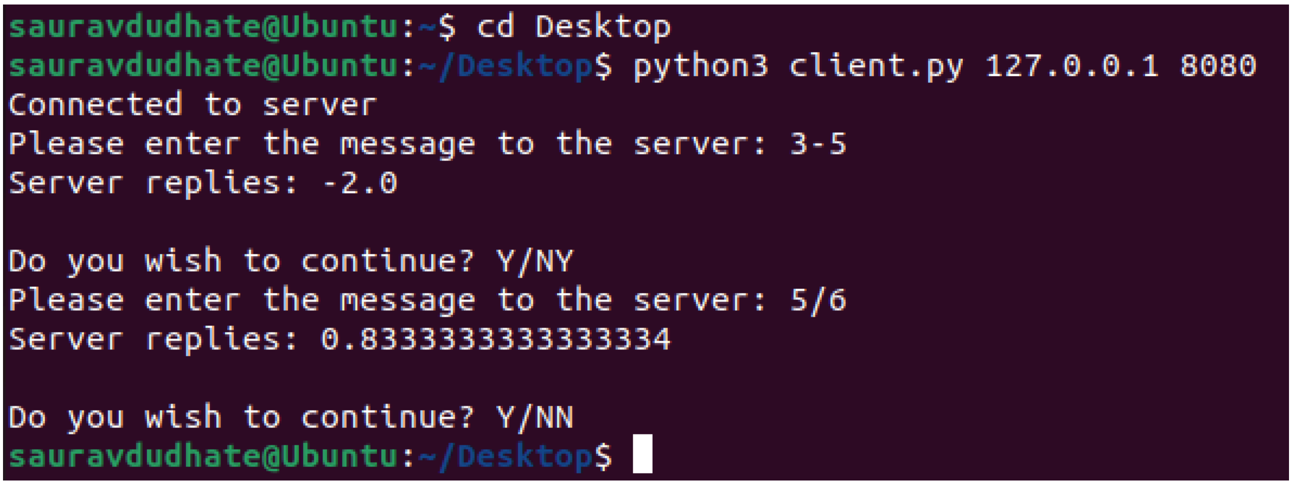
```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 client.py 127.0.0.1 8080
Connected to server
Please enter the message to the server: 3-5
Server replies: -2.0

Do you wish to continue? Y/NY
Please enter the message to the server: 
```

Client2:



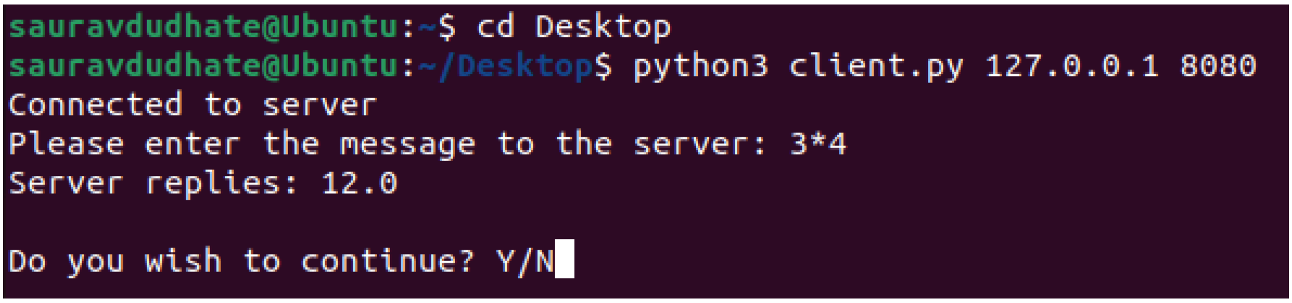
```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 client.py 127.0.0.1 8080
Connected to server
Please enter the message to the server: 3*4
```

Client1:



```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 client.py 127.0.0.1 8080
Connected to server
Please enter the message to the server: 3-5
Server replies: -2.0

Do you wish to continue? Y/NY
Please enter the message to the server: 5/6
Server replies: 0.8333333333333334

Do you wish to continue? Y/NN
sauravdudhate@Ubuntu:~/Desktop$ 
```

Client2:

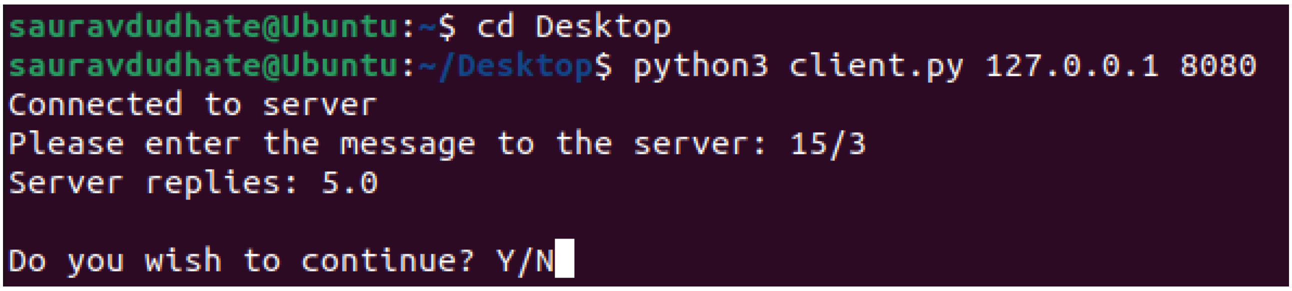


```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 client.py 127.0.0.1 8080
Connected to server
Please enter the message to the server: 3*4
Server replies: 12.0

Do you wish to continue? Y/N
```

## Server 3:

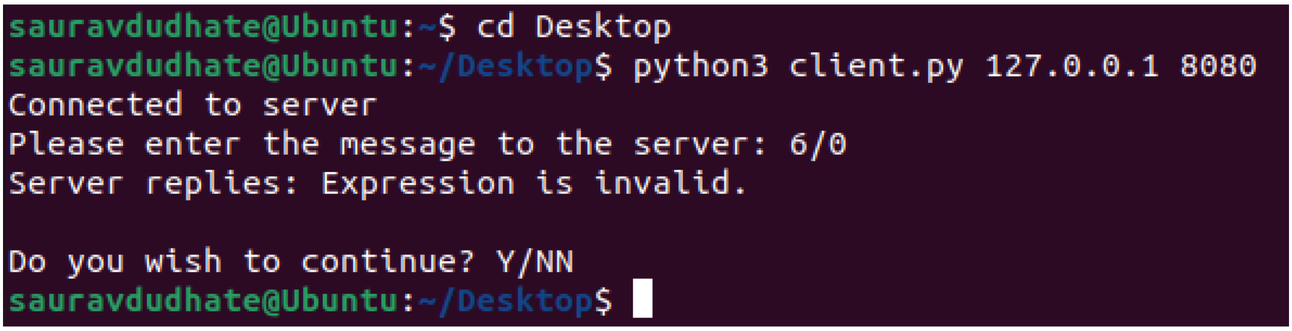
**Input : “15/3” (Client1) -> “6/0” (Client2) -> “88/4-11+9” (Client3)**
**Output: “5.0” (Client1) -> “Expression is invalid”  (Client2) -> “20.0” (Client3)**

Client1:



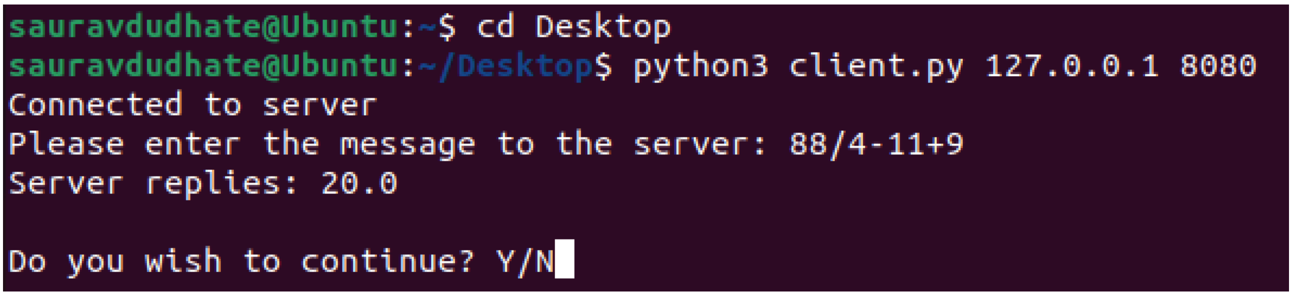
```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 client.py 127.0.0.1 8080
Connected to server
Please enter the message to the server: 15/3
Server replies: 5.0

Do you wish to continue? Y/N
```

Client2:



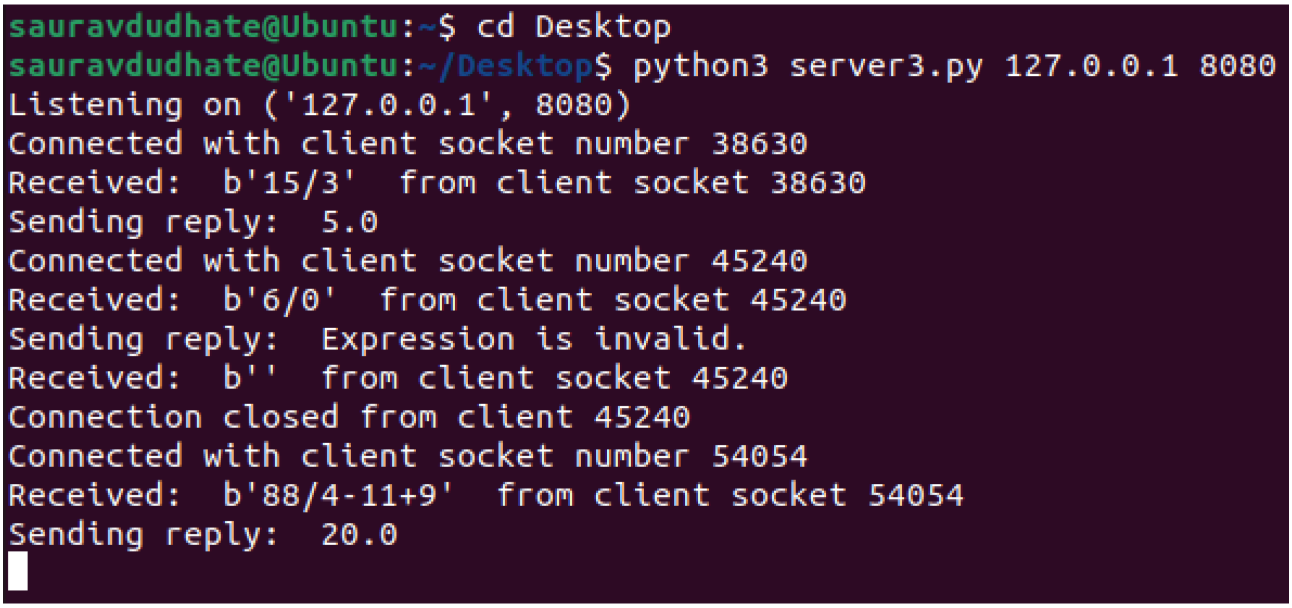
```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 client.py 127.0.0.1 8080
Connected to server
Please enter the message to the server: 6/0
Server replies: Expression is invalid.

Do you wish to continue? Y/NN
sauravdudhate@Ubuntu:~/Desktop$
```

Client3:



```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 client.py 127.0.0.1 8080
Connected to server
Please enter the message to the server: 88/4-11+9
Server replies: 20.0

Do you wish to continue? Y/N
```

Server:

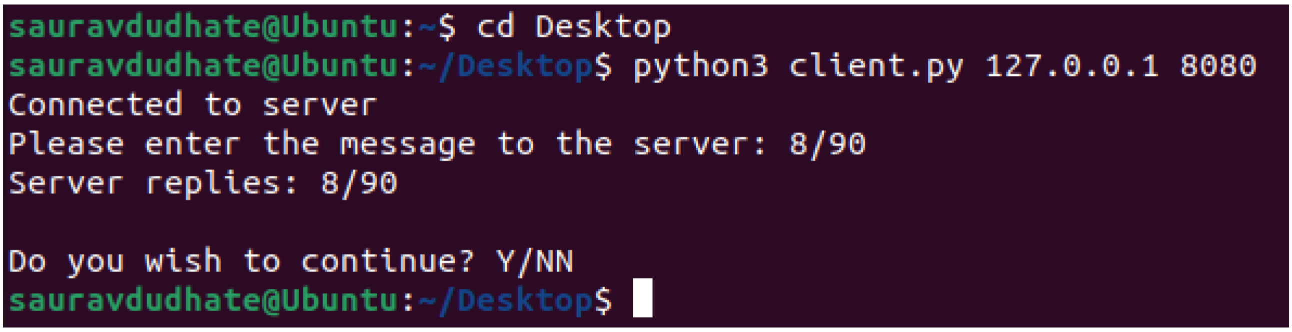


```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 server3.py 127.0.0.1 8080
Listening on ('127.0.0.1', 8080)
Connected with client socket number 38630
Received:  b'15/3'  from client socket 38630
Sending reply:  5.0
Connected with client socket number 45240
Received:  b'6/0'  from client socket 45240
Sending reply:  Expression is invalid.
Received:  b''  from client socket 45240
Connection closed from client 45240
Connected with client socket number 54054
Received:  b'88/4-11+9'  from client socket 54054
Sending reply:  20.0
```

## Server 4:

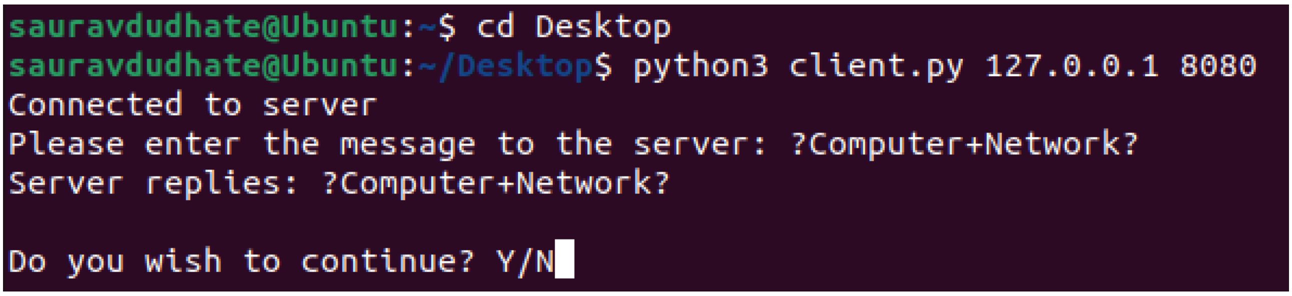
**Input: “8/90” (Client2) -> “?Computer+Network?” (Client3) -> “Saurav90” (Client1)**
**Output: “8/90” (Client2) -> “?Computer+Network?” (Client3) -> “Saurav90” (Client1)**

Client2:



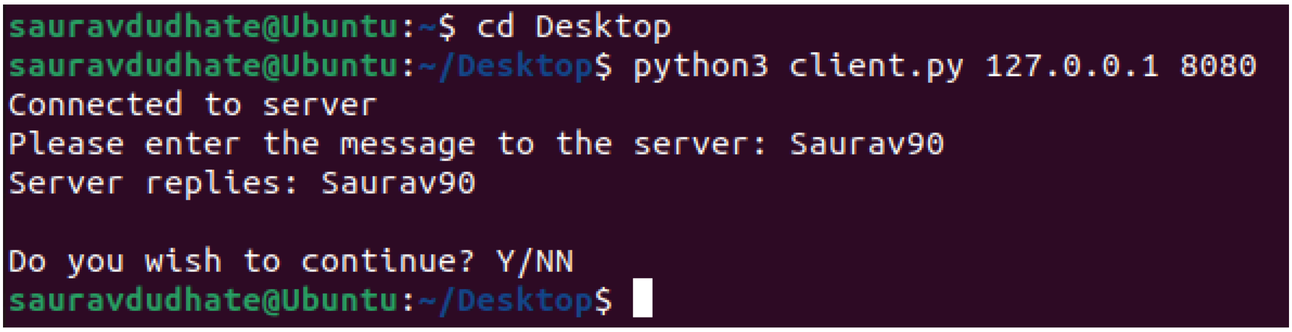
```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 client.py 127.0.0.1 8080
Connected to server
Please enter the message to the server: 8/90
Server replies: 8/90

Do you wish to continue? Y/NN
sauravdudhate@Ubuntu:~/Desktop$
```

Client3:



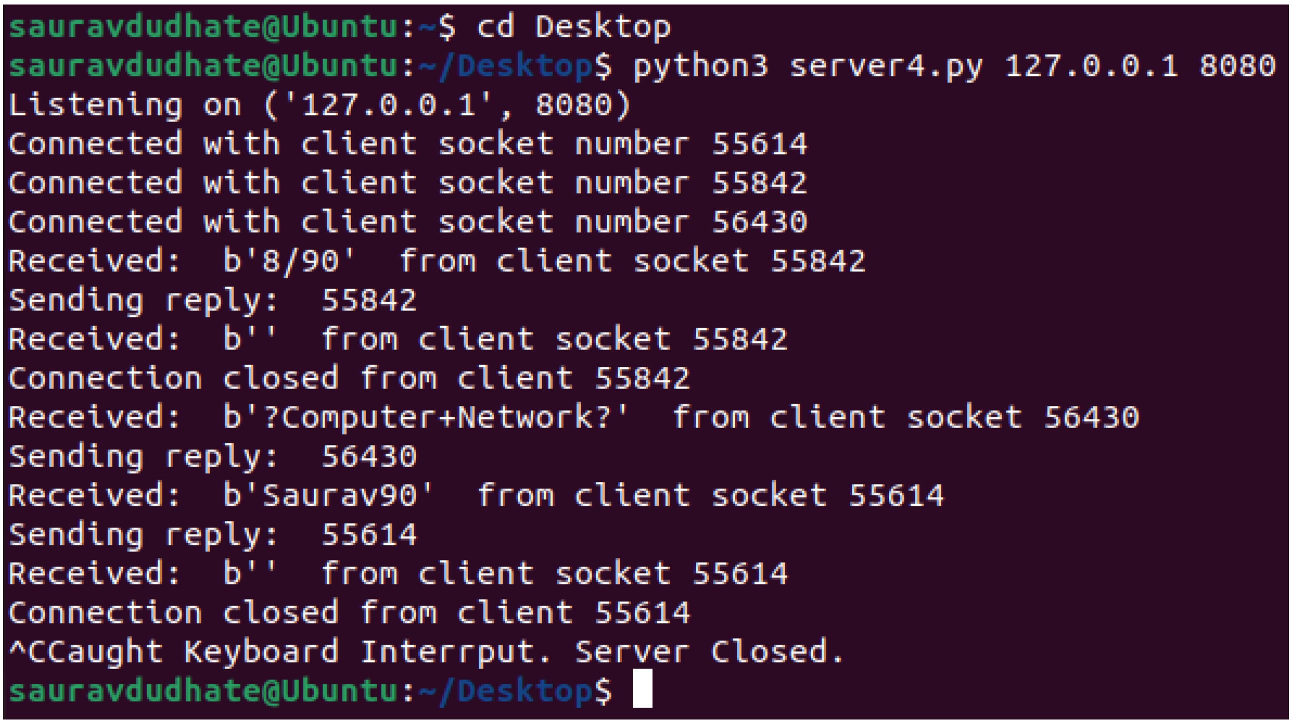
```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 client.py 127.0.0.1 8080
Connected to server
Please enter the message to the server: ?Computer+Network?
Server replies: ?Computer+Network?

Do you wish to continue? Y/N
```

Client1:



```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 client.py 127.0.0.1 8080
Connected to server
Please enter the message to the server: Saurav90
Server replies: Saurav90

Do you wish to continue? Y/NN
sauravdudhate@Ubuntu:~/Desktop$ 
```

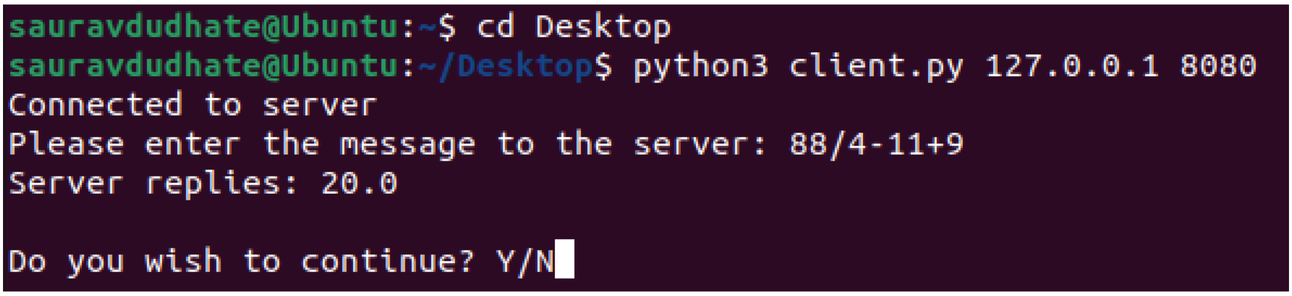
Server:



```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 server4.py 127.0.0.1 8080
Listening on ('127.0.0.1', 8080)
Connected with client socket number 55614
Connected with client socket number 55842
Connected with client socket number 56430
Received:  b'8/90'  from client socket 55842
Sending reply:  55842
Received:  b''  from client socket 55842
Connection closed from client 55842
Received:  b'?Computer+Network?'  from client socket 56430
Sending reply:  56430
Received:  b'Saurav90'  from client socket 55614
Sending reply:  55614
Received:  b''  from client socket 55614
Connection closed from client 55614
^CCaught Keyboard Interrput. Server Closed.
sauravdudhate@Ubuntu:~/Desktop$ 
```

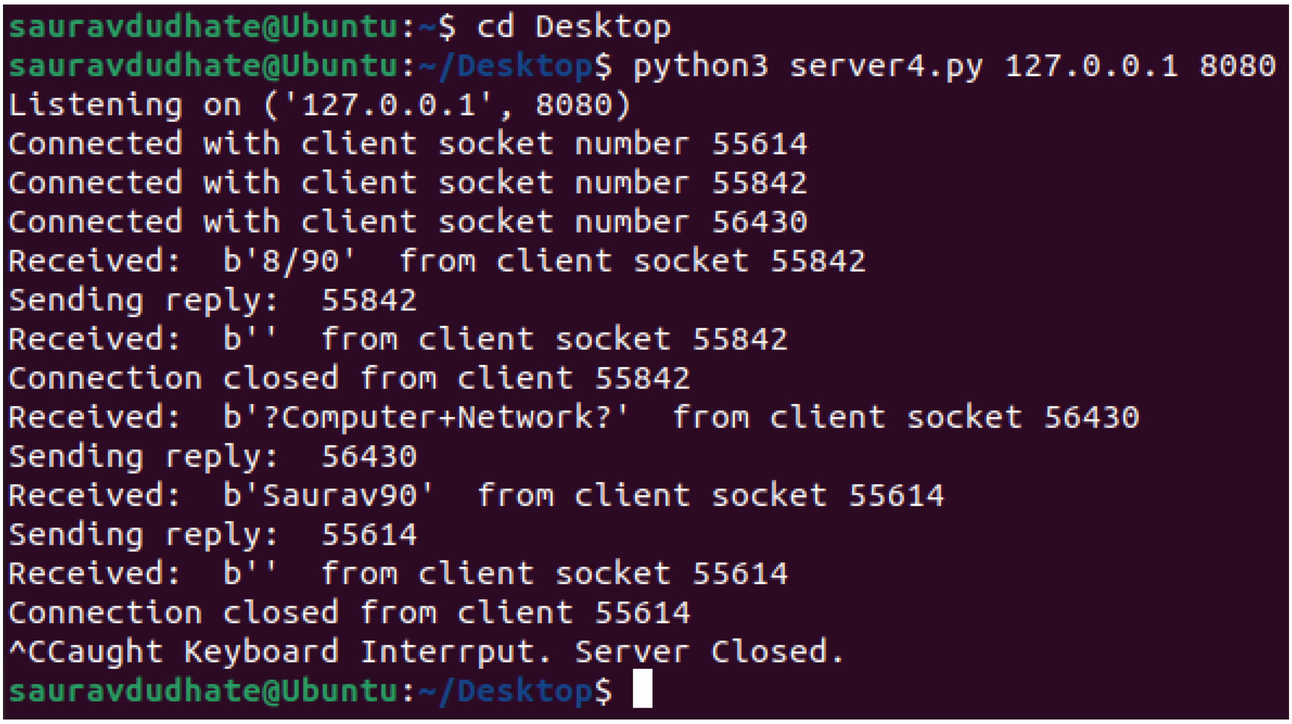
# Additional Features

1. Complex calculations involving any number of operands can be done rather than just 2 operands.



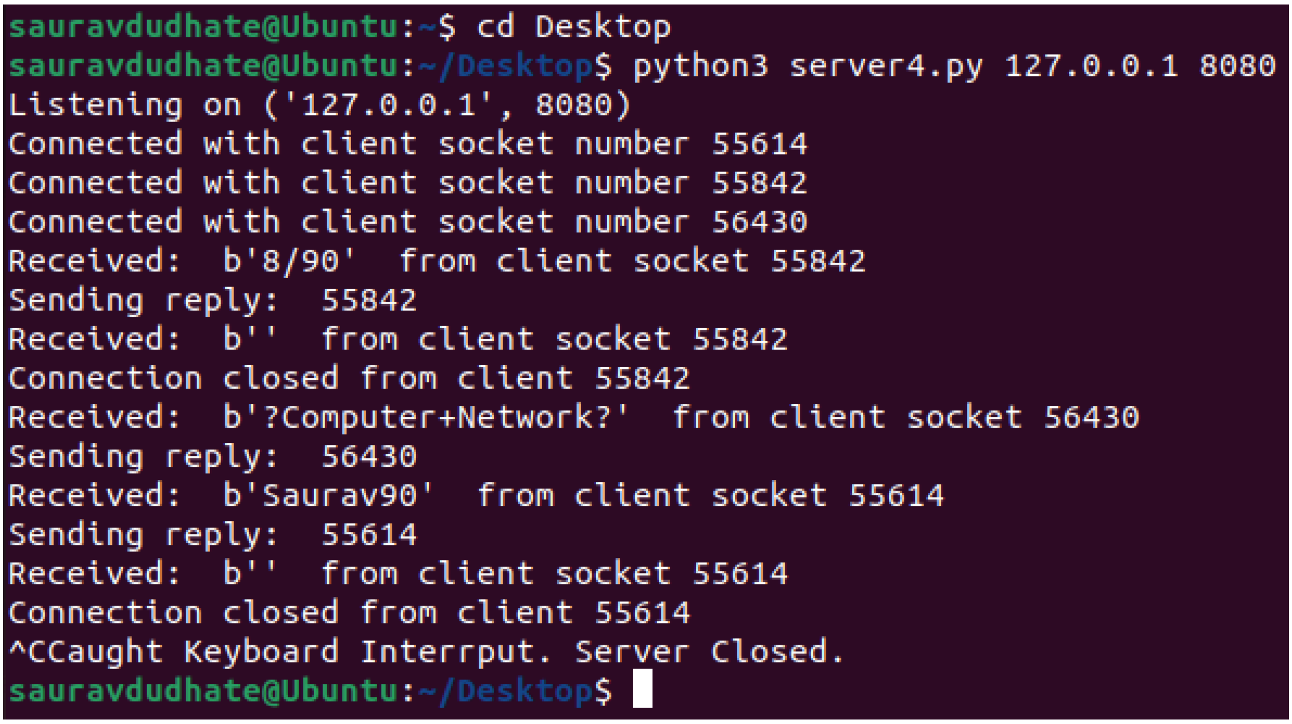
```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 client.py 127.0.0.1 8080
Connected to server
Please enter the message to the server: 88/4-11+9
Server replies: 20.0

Do you wish to continue? Y/N
```

2. Server displays log regarding the socket number of the client in case of an incoming or outgoing message or opening or closing connection.



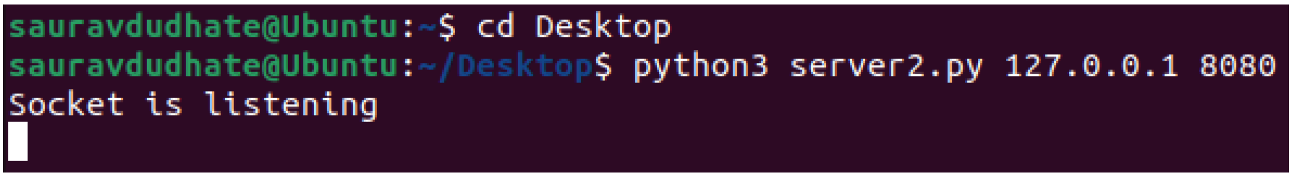
```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 server4.py 127.0.0.1 8080
Listening on ('127.0.0.1', 8080)
Connected with client socket number 55614
Connected with client socket number 55842
Connected with client socket number 56430
Received:  b'8/90'  from client socket 55842
Sending reply:  55842
Received:  b''  from client socket 55842
Connection closed from client 55842
Received:  b'?Computer+Network?'  from client socket 56430
Sending reply:  56430
Received:  b'Saurav90'  from client socket 55614
Sending reply:  55614
Received:  b''  from client socket 55614
Connection closed from client 55614
^CCaught Keyboard Interrput. Server Closed.
sauravdudhate@Ubuntu:~/Desktop$ 
```

3. Server handles the Keyboard Interrupt error and displays a message before closing.

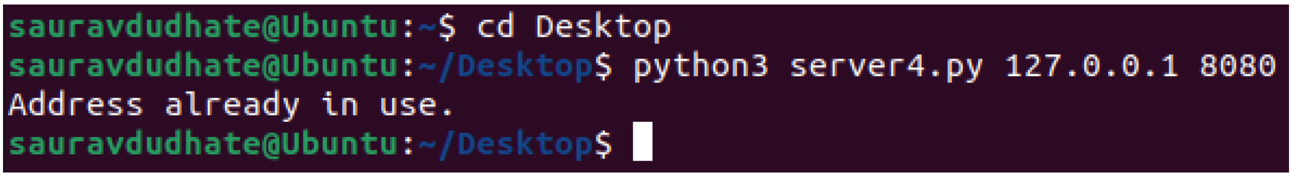


```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 server4.py 127.0.0.1 8080
Listening on ('127.0.0.1', 8080)
Connected with client socket number 55614
Connected with client socket number 55842
Connected with client socket number 56430
Received:  b'8/90'  from client socket 55842
Sending reply:  55842
Received:  b''  from client socket 55842
Connection closed from client 55842
Received:  b'?Computer+Network?'  from client socket 56430
Sending reply:  56430
Received:  b'Saurav90'  from client socket 55614
Sending reply:  55614
Received:  b''  from client socket 55614
Connection closed from client 55614
^CCaught Keyboard Interrput. Server Closed.
sauravdudhate@Ubuntu:~/Desktop$ 
```

4. Handling of exception when a server tries to connect to an address which is already connected to a socket.

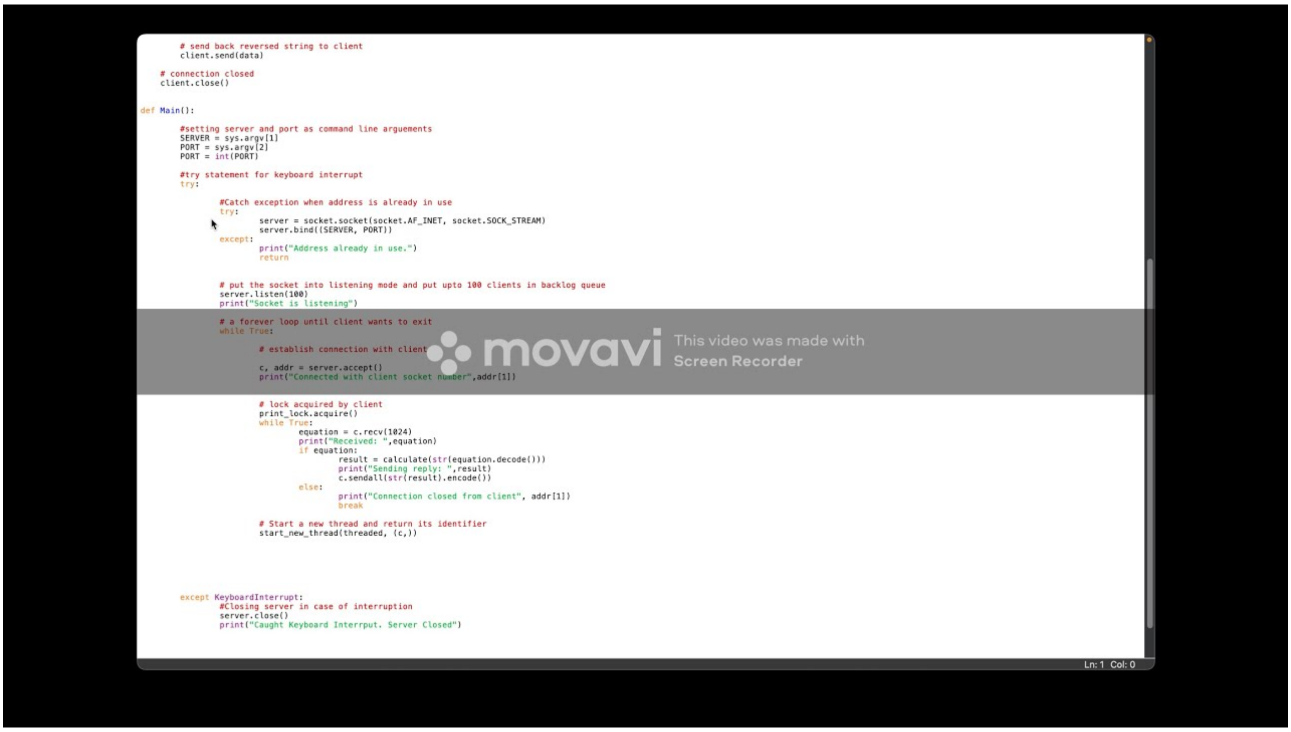


```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 server2.py 127.0.0.1 8080
Socket is listening
```



```
sauravdudhate@Ubuntu:~$ cd Desktop
sauravdudhate@Ubuntu:~/Desktop$ python3 server4.py 127.0.0.1 8080
Address already in use.
sauravdudhate@Ubuntu:~/Desktop$
```

# Youtube Video



```
        # send back reversed string to client
        client.send(data)

    # connection closed
    client.close()

def Main():

        #setting server and port as command line arguements
        SERVER = sys.argv[1]
        PORT = sys.argv[2]
        PORT = int(PORT)

        #try statement for keyboard interrupt
        try:

                #Catch exception when address is already in use
                try:
                        server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
                        server.bind((SERVER, PORT))
                except:
                        print("Address already in use.")
                        return


                # put the socket into listening mode and put upto 100 clients in backlog queue
                server.listen(100)
                print("Socket is listening")

                # a forever loop until client wants to exit
                while True:

                        # establish connection with client

                        c, addr = server.accept()
                        print("Connected with client socket number",addr[1])

                        # lock acquired by client
                        print_lock.acquire()
                        while True:
                                equation = c.recv(1024)
                                print("Received: ",equation)
                                if equation:
                                        result = calculate(str(equation.decode()))
                                        print("Sending reply: ",result)
                                        c.sendall(str(result).encode())
                                else:
                                        print("Connection closed from client", addr[1])
                                        break

                        # Start a new thread and return its identifier
                        start_new_thread(threaded, (c,))



        except KeyboardInterrupt:
                #Closing server in case of interruption
                server.close()
                print("Caught Keyboard Interrput. Server Closed")
```