

TsClient.py

TsServer.py

```
#!usr/bin/env python

from socket import *
from time import ctime

HOST = ''
PORT = 21567
BUFSIZ = 1024
ADDR = (HOST, PORT)

tcpSerSock = socket(AF_INET, SOCK_STREAM)
tcpSerSock.bind(ADDR)
tcpSerSock.listen(5)

while True:
    print('waiting for connection...')
    tcpCliSock, addr = tcpSerSock.accept()
    print('...connected from:', addr)

    while True:
        data = input('> ')
        if not data:
            break
        tcpCliSock.send(('[%s] %s' % (ctime(), data)).encode())

        datal = tcpCliSock.recv(BUFSIZ).decode()
        if not datal:
            break
        print(datal)

    tcpCliSock.close()
tcpSerSock.close()
```

TsClient.py :-

```
#!/usr/bin/env python
```

```
from socket import *  
from time import ctime
```

```
HOST = 'localhost'
```

```
PORT = 21567
```

```
BUFSIZ = 1024
```

```
ADDR = (HOST, PORT)
```

```
tcpCliSock = socket(AF_INET, SOCK_STREAM)
```

```
tcpCliSock.connect(ADDR)
```

```
while True:
```

```
    data1 = tcpCliSock.recv(BUFSIZ).decode()
```

```
    if not data1:
```

```
        break
```

```
    print (data1)
```

```
    data = input('> ')
```

```
    if not data:
```

```
        break
```

```
    tcpCliSock.send(('[%s] %s' % (ctime(), data)).encode())
```

```
tcpCliSock.close()
```

TsServer.py :-

```
#!/usr/bin/env python

from socket import *
from time import ctime

HOST = ''
PORT = 21567
BUFSIZ = 1024
ADDR = (HOST, PORT)

tcpSerSock = socket(AF_INET, SOCK_STREAM)
tcpSerSock.bind(ADDR)
tcpSerSock.listen(5)

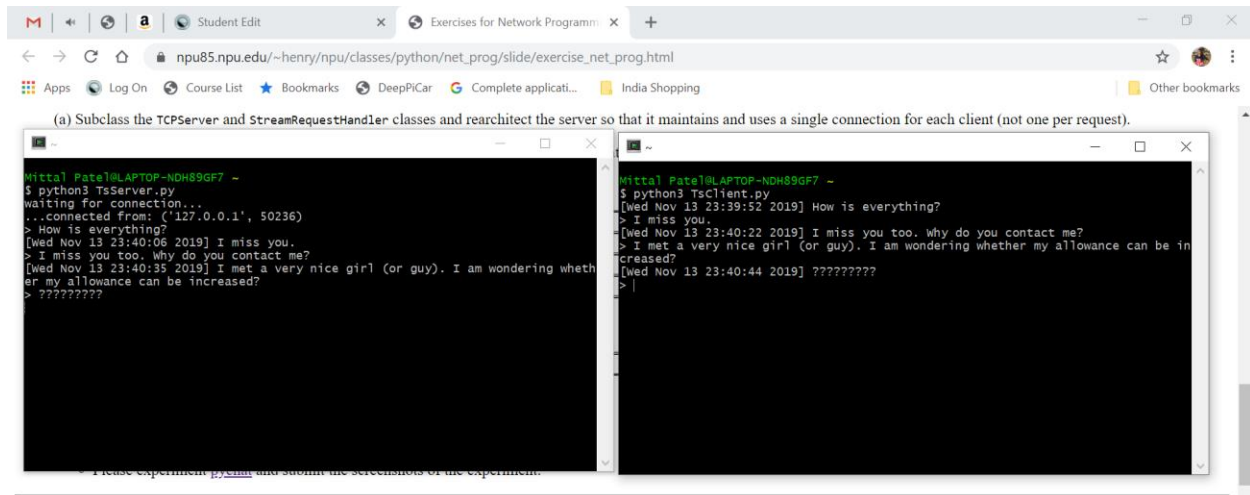
while True:
    print ('waiting for connection...')
    tcpCliSock, addr = tcpSerSock.accept()
    print ('...connected from:', addr)

    while True:
        data = input('> ')
        if not data:
            break
        tcpCliSock.send(('[%s] %s' % (ctime(), data)).encode())

        data1 = tcpCliSock.recv(BUFSIZ).decode()
        if not data1:
            break
        print (data1)

    tcpCliSock.close()
tcpSerSock.close()
```

Execution Result



The screenshot shows a web browser window with the address bar displaying `npu85.npu.edu/~henry/npu/classes/python/net_prog/slide/exercise_net_prog.html`. The page content includes a subtask instruction: (a) Subclass the `TCPServer` and `StreamRequestHandler` classes and rearchitect the server so that it maintains and uses a single connection for each client (not one per request).

Below the instruction are two terminal windows. The left terminal shows the execution of `TsServer.py`, which starts listening for connections. It receives a connection from `127.0.0.1` at `50236`. The conversation log shows:
- User input: `> How is everything?`
- Server output: `[Wed Nov 13 23:40:06 2019] I miss you.`
- User input: `> I miss you too. Why do you contact me?`
- Server output: `[Wed Nov 13 23:40:35 2019] I met a very nice girl (or guy). I am wondering whether my allowance can be increased?`
- User input: `> ??????????`

The right terminal shows the execution of `TsClient.py`, which sends the same messages to the server:
- User input: `> How is everything?`
- Server output: `[Wed Nov 13 23:39:52 2019] How is everything?`
- User input: `> I miss you.`
- Server output: `[Wed Nov 13 23:40:22 2019] I miss you too. Why do you contact me?`
- User input: `> I met a very nice girl (or guy). I am wondering whether my allowance can be increased?`
- Server output: `[Wed Nov 13 23:40:44 2019] ??????????`

Last modified on: 10/22/2018 12:31:18