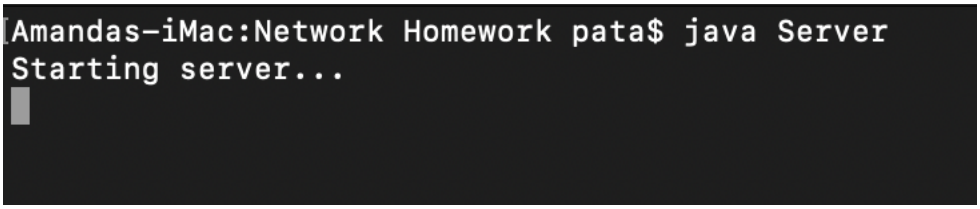
Patrick Griffin
CSCI 4311 – Socket Programming Assignment Report

The first step of is to execute the Server class. In my server class I instantiate a server socket at the port 5000. The server then listens on port 5000 for any client sockets to accept.

Server:



After the server class has been launched, you can now execute the client class. The client class is allowed to be given a manual IP address and port number. In this case you would input:

```
java Client localhost 5000
```

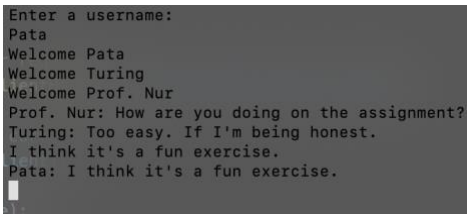
If an IP address or port number is given the client will attempt to connect to default setting, ip = localhost and port = 5000. The client is then immediately asked for a username. Once the user has input a username they are logged into the server and allowed to chat with other clients.

Client 1:

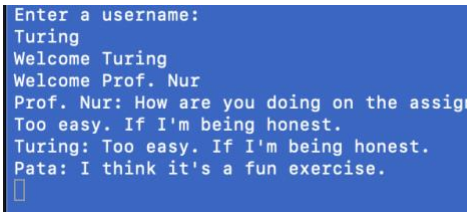
Server:

Once the client has been accepted by the server class and a username is entered the client establishes a input and output stream and is given to a client handler on the server side. On the client side, the client is now listening in by using separate threads for input and output streams. Back in the server, the clientHandler creates an object that has the following properties: DataInputStream, DataOutputStream, name, socket and a Boolean indicating whether the client is still logged in or not. This ClientHandler object also implements the run() function for when this object is given its own thread. Which is what immediately following the instantiation of the ClientHandler object. In the server's main function, the client is given a greeting message and the client is added to an arraylist of clients to maintain communication.

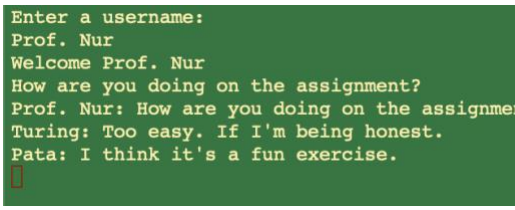
Client 1:



Client 2:



Client 3:



Server:

```
New client request received : Socket[addr=/127.0.0.1,port=58893,localport=5000]
Creating client handler
Welcome Pata
New client request received : Socket[addr=/127.0.0.1,port=58932,localport=5000]
Creating client handler
Welcome Turing
New client request received : Socket[addr=/127.0.0.1,port=58933,localport=5000]
Creating client handler
Welcome Prof. Nur
Prof. Nur: How are you doing on the assignment?
Turing: Too easy. If I'm being honest.
Pata: I think it's a fun exercise.
```

When a client types a message the sendMessage thread writes the input from that client onto the I/O stream. Server side, the message is read from the I/O stream and printed to the server which then iterates through all clients that are logged in and delivers the message to those clients over the I/O stream. This continues until a client is ready to log off by sending the message “Bye” (case-sensitive) to the server. Afterwards the other clients still logged in to the server are notified of the client leaving with a goodbye message.

Client 1:

```
I have to head out. Later.
Pata: I have to head out. Later.
Bye
```

Client 2:

```
Pata: I have to head out. Later.
Server: Goodbye Pata
```

Client 3:

```
Pata: I have to head out. Later.
Server: Goodbye Pata
```

Server:

```
Pata: I think it's a fun exercise.
Pata: I have to head out. Later.
Server: Goodbye Pata
```

The client then closes the DataInputStream and DataOutputStream and the client is flagged as logged out in the clientList.