

CSCI 4311/5311

Socket Programming

Programming Assignment 1

Due Date: Wednesday, October 14, 2020, 11:59 PM

Goal of the assignment

In this assignment, we build a simple group chat application. The protocol between the client and the server is as follows.

- You can choose to use either TCP or UDP in your implementation. (TCP preferred)
- The server is first started on a known port.
- The client program is started (server IP and port are provided on the command line).
- The client connects to the server. The server asks the user for input. The user types the username in the following format "username = ComNet" message on the terminal.
- If the user doesn't provide a username, the server doesn't accept the user's messages.
- After the user enters a username, the server broadcast to everyone "Server: Welcome username".
- After that, the user can send messages (e.g., "Hi", "Bye", "How are you"). The user's input is sent to the server via the connected socket.
- The server reads the user's input from the client's socket. If the user has typed "Bye", the server must broadcast to everyone with "Goodbye username" e.g. "Server: Goodbye ComNet".
- If a user enters "AllUsers", the server needs to send all active users to that user.

Rules:

- The server must be able to monitor and handle messages from multiple clients simultaneously. Therefore, your server needs to be multithreaded.
- Do NOT implement a peer-to-peer application. E.g., clients talk directly to each other.
- The architecture here is server-client. All messages from the clients go through the server. The server distributes the messages to all other clients who have a connection to the server.
- Implement the assignment with Java

- You need to write a report to explain your code, put some screenshots for your outputs, etc. Save your report in PDF format.
- Without the report, you don't get any points.
- GUI is optional with 20 bonus points. If you implement the GUI version well, you will get an additional 20 points. Example:

Step 1: Execute Server code

e.g. `java Server 8989`

This starts the server listening on the port number 8989

Step 2: Execute Client code

`java Client localhost 8989`

It tells the client to connect to the server at localhost on port 8989. Then you see the following message in the server's console:

The program asks for the username:

e.g. Enter your username:

Let say client 1 enters UNO.

Server prints "Welcome UNO"

Now, UNO can enter any message

Step 3: Execute Client code one more time to get a new user

The program asks for the username:

e.g. Enter your username:

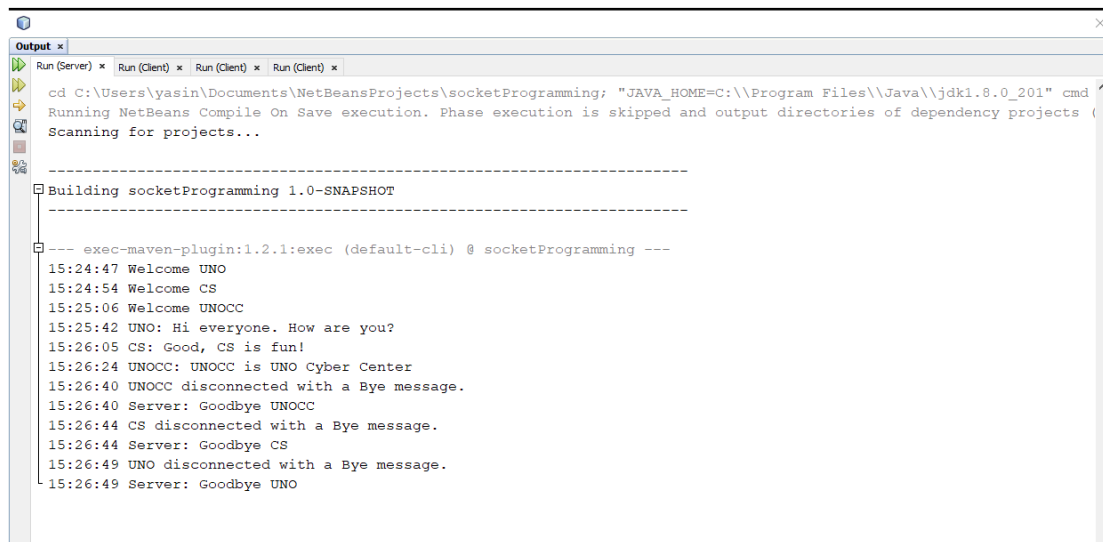
Let say client 2 enters CS.

Server prints "Welcome CS"

Now, CS can enter any message

EXAMPLE:

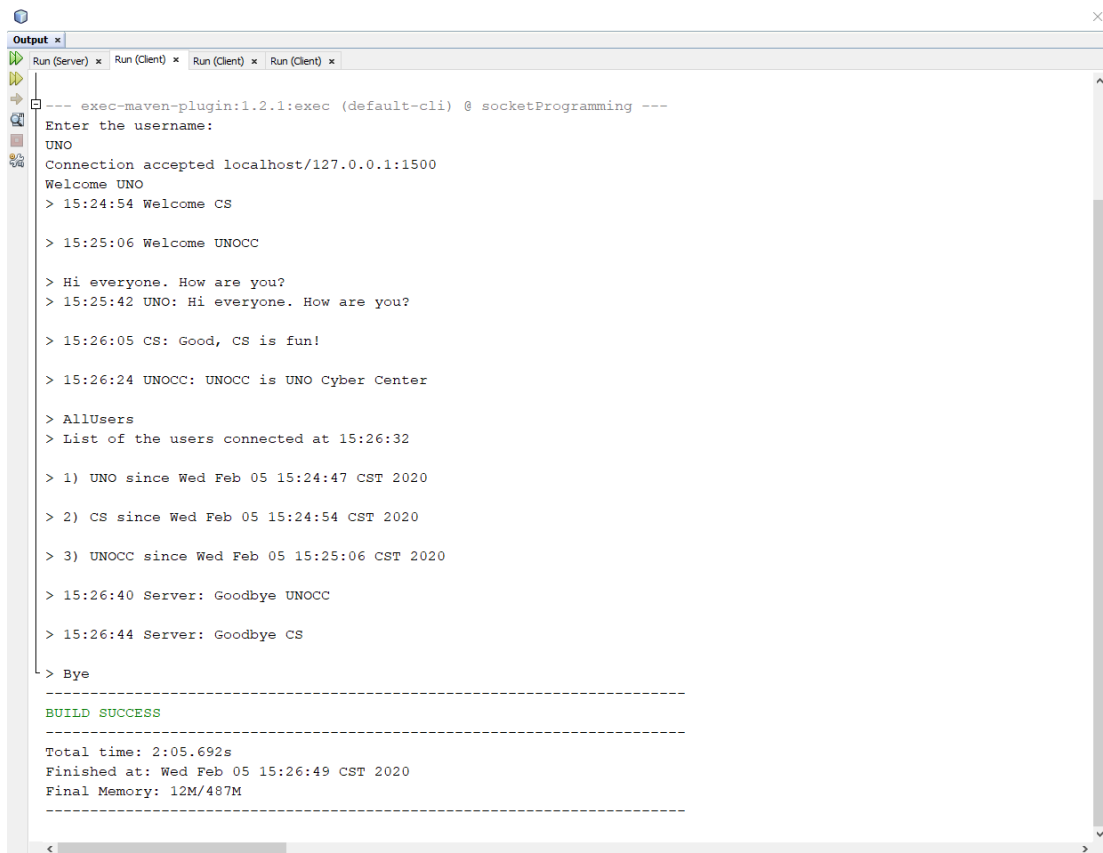
Server Output:



```
cd C:\Users\yasin\Documents\NetBeansProjects\socketProgramming; "JAVA_HOME=C:\\Program Files\\Java\\jdk1.8.0_201" cmd
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (
Scanning for projects...

-----
[ ] Building socketProgramming 1.0-SNAPSHOT
-----
[ ] --- exec-maven-plugin:1.2.1:exec (default-cli) @ socketProgramming ---
15:24:47 Welcome UNO
15:24:54 Welcome CS
15:25:06 Welcome UNOCC
15:25:42 UNO: Hi everyone. How are you?
15:26:05 CS: Good, CS is fun!
15:26:24 UNOCC: UNOCC is UNO Cyber Center
15:26:40 UNOCC disconnected with a Bye message.
15:26:40 Server: Goodbye UNOCC
15:26:44 CS disconnected with a Bye message.
15:26:44 Server: Goodbye CS
15:26:49 UNO disconnected with a Bye message.
15:26:49 Server: Goodbye UNO
```

Client 1:



```
--- exec-maven-plugin:1.2.1:exec (default-cli) @ socketProgramming ---
Enter the username:
UNO
Connection accepted localhost/127.0.0.1:1500
Welcome UNO
> 15:24:54 Welcome CS

> 15:25:06 Welcome UNOCC

> Hi everyone. How are you?
> 15:25:42 UNO: Hi everyone. How are you?

> 15:26:05 CS: Good, CS is fun!

> 15:26:24 UNOCC: UNOCC is UNO Cyber Center

> AllUsers
> List of the users connected at 15:26:32

> 1) UNO since Wed Feb 05 15:24:47 CST 2020
> 2) CS since Wed Feb 05 15:24:54 CST 2020
> 3) UNOCC since Wed Feb 05 15:25:06 CST 2020

> 15:26:40 Server: Goodbye UNOCC

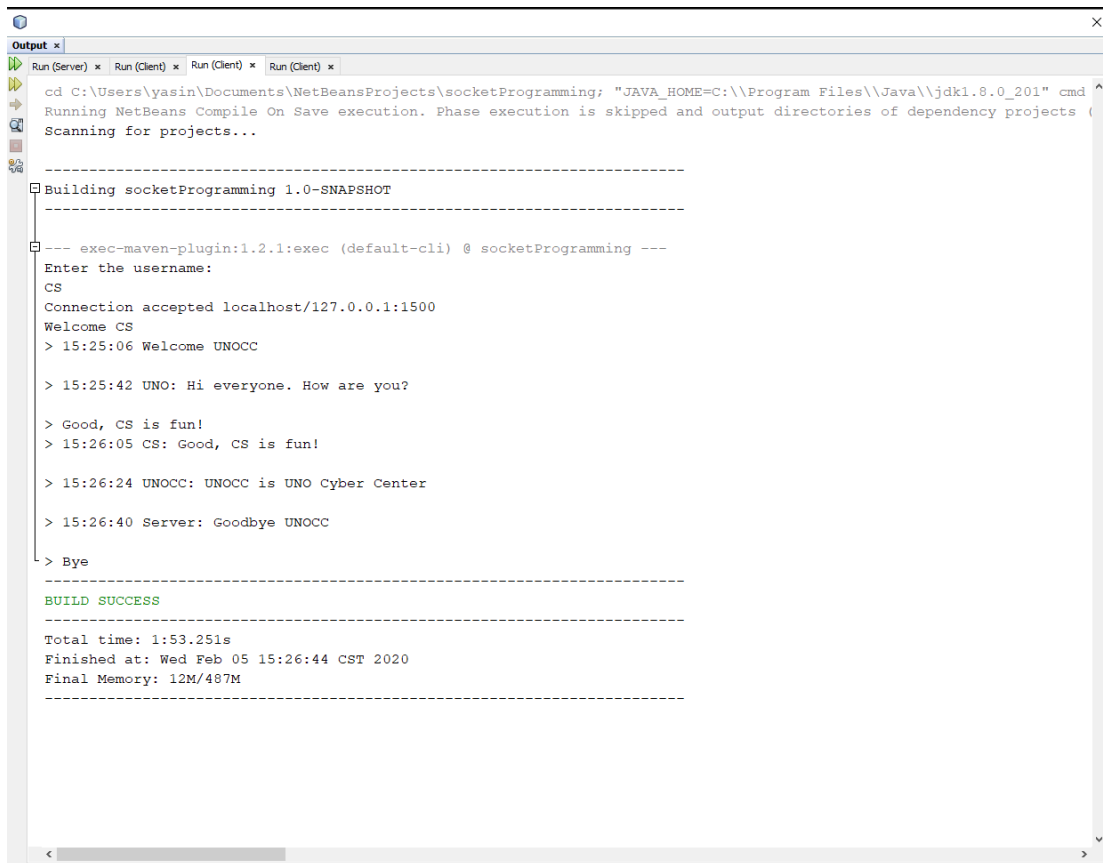
> 15:26:44 Server: Goodbye CS

> Bye

BUILD SUCCESS

Total time: 2:05.692s
Finished at: Wed Feb 05 15:26:49 CST 2020
Final Memory: 12M/487M
```

Client 2:



```
cd C:\Users\yasin\Documents\NetBeansProjects\socketProgramming; "JAVA_HOME=C:\\Program Files\\Java\\jdk1.8.0_201" cmd
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (
Scanning for projects...

-----
Building socketProgramming 1.0-SNAPSHOT
-----

--- exec-maven-plugin:1.2.1:exec (default-cli) @ socketProgramming ---
Enter the username:
CS
Connection accepted localhost/127.0.0.1:1500
Welcome CS
> 15:25:06 Welcome UNOCC

> 15:25:42 UNO: Hi everyone. How are you?

> Good, CS is fun!
> 15:26:05 CS: Good, CS is fun!

> 15:26:24 UNOCC: UNOCC is UNO Cyber Center

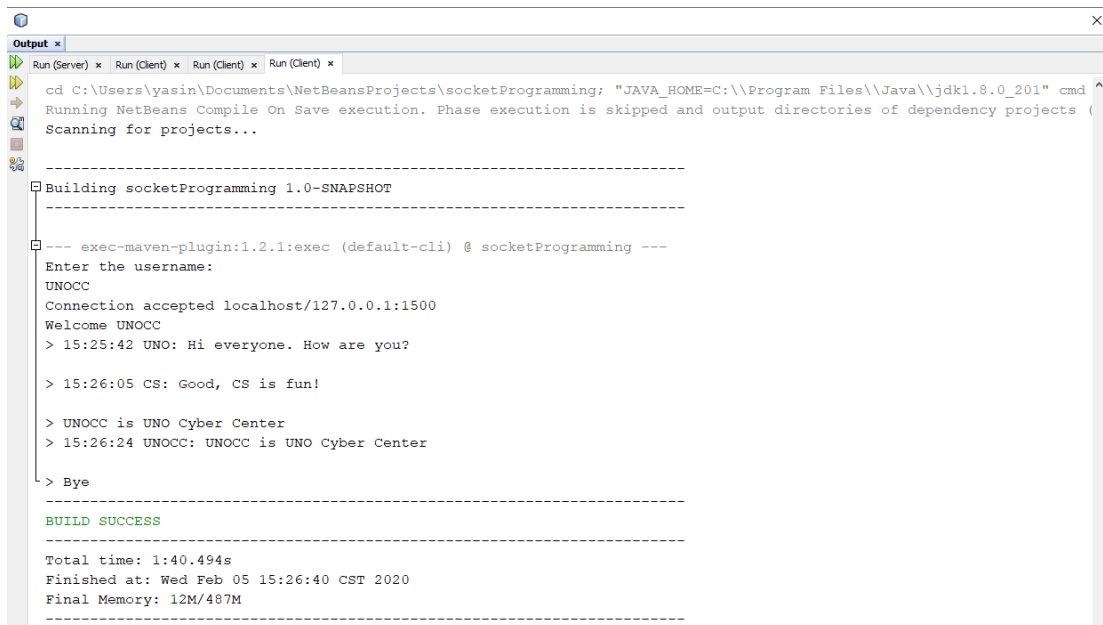
> 15:26:40 Server: Goodbye UNOCC

> Bye

-----
BUILD SUCCESS
-----

Total time: 1:53.251s
Finished at: Wed Feb 05 15:26:44 CST 2020
Final Memory: 12M/487M
-----
```

Client 3:



```
cd C:\Users\yasin\Documents\NetBeansProjects\socketProgramming; "JAVA_HOME=C:\\Program Files\\Java\\jdk1.8.0_201" cmd
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (
Scanning for projects...

-----
Building socketProgramming 1.0-SNAPSHOT
-----

--- exec-maven-plugin:1.2.1:exec (default-cli) @ socketProgramming ---
Enter the username:
UNOCC
Connection accepted localhost/127.0.0.1:1500
Welcome UNOCC
> 15:25:42 UNO: Hi everyone. How are you?

> 15:26:05 CS: Good, CS is fun!

> UNOCC is UNO Cyber Center
> 15:26:24 UNOCC: UNOCC is UNO Cyber Center

> Bye

-----
BUILD SUCCESS
-----

Total time: 1:40.494s
Finished at: Wed Feb 05 15:26:40 CST 2020
Final Memory: 12M/487M
-----
```