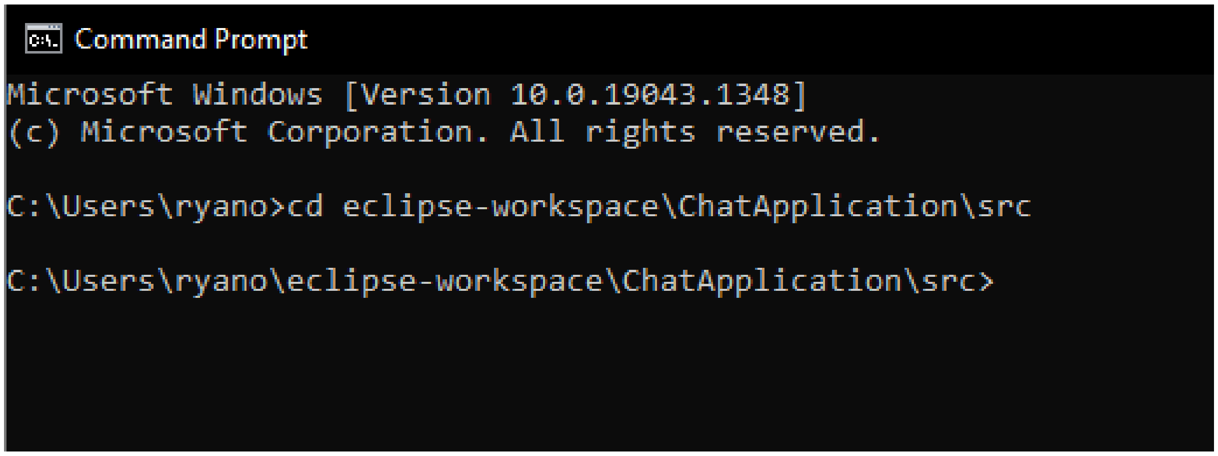
Instructions to use this chat application:

I have provided a zip file with three java files inside (ChatClient.java, ChatMessage.java, and ChatServer.java). The general idea of these instructions is that everything will be run on command prompt windows, and the server must be set up first before setting up clients.

1. Download the zip file and make sure to extract all the files into a folder.

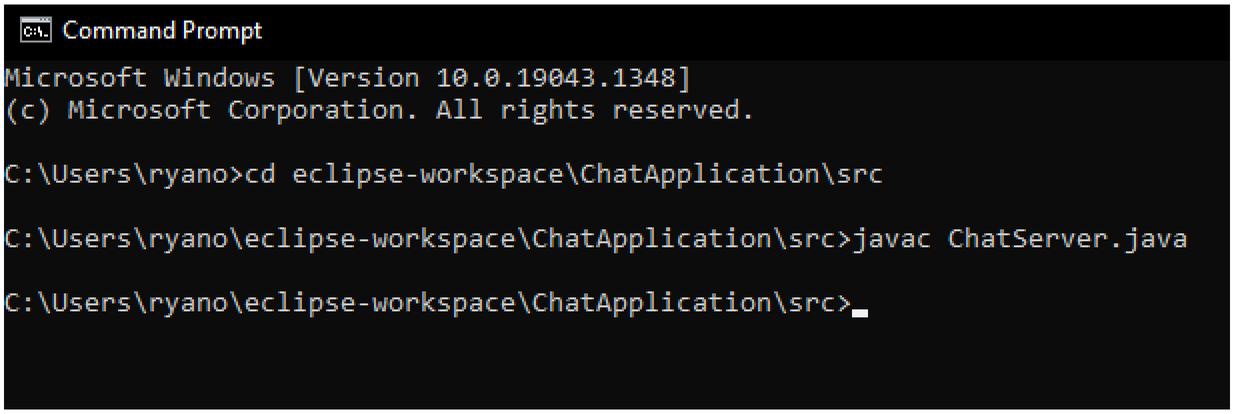
   Steps 2 to 4 will involve setting up the server.

2. Open up the command prompt and use the change directory command to whatever location the folder with three files is stored in.
   (ex: cd eclipse-workspace\ChatApplication\src)



```
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ryano>cd eclipse-workspace\ChatApplication\src

C:\Users\ryano\eclipse-workspace\ChatApplication\src>
```

3. Type “javac ChatServer.java“ in the command prompt and press Enter.

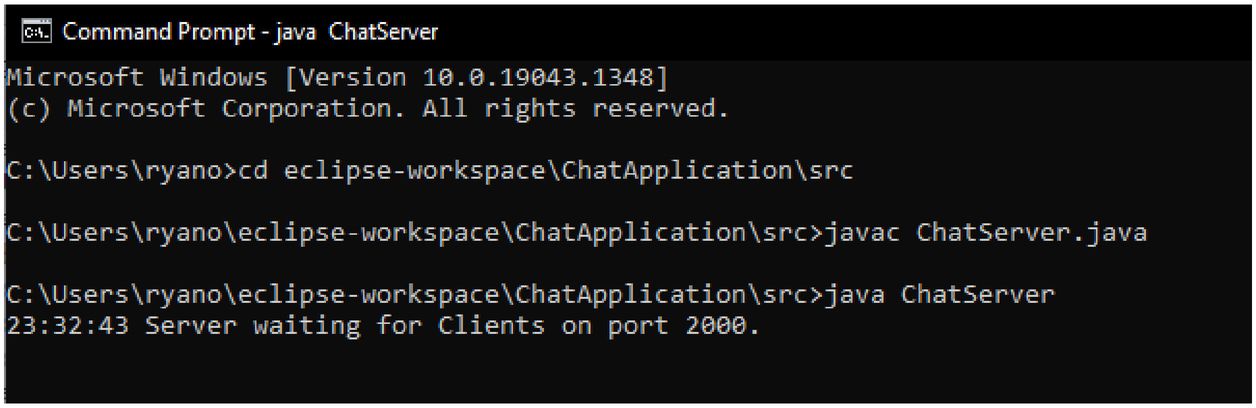


```
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ryano>cd eclipse-workspace\ChatApplication\src

C:\Users\ryano\eclipse-workspace\ChatApplication\src>javac ChatServer.java

C:\Users\ryano\eclipse-workspace\ChatApplication\src>_
```

Note: If the “javac” command is not working for you, check if you have JDK installed and please install it if JDK is not yet on your computer (link to the JDK downloads: https://www.oracle.com/java/technologies/downloads/#jdk17-windows).

4. Type “java ChatServer“ in the command prompt and press Enter in order to run the server.

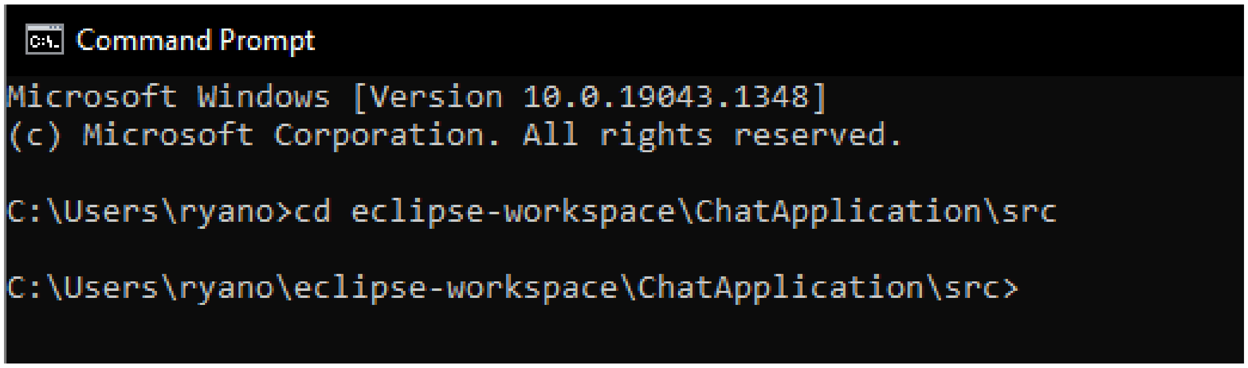
   If the below message shows up, this means the server should be working properly.



```
Command Prompt - java  ChatServer
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ryano>cd eclipse-workspace\ChatApplication\src

C:\Users\ryano\eclipse-workspace\ChatApplication\src>javac ChatServer.java

C:\Users\ryano\eclipse-workspace\ChatApplication\src>java ChatServer
23:32:43 Server waiting for Clients on port 2000.
```

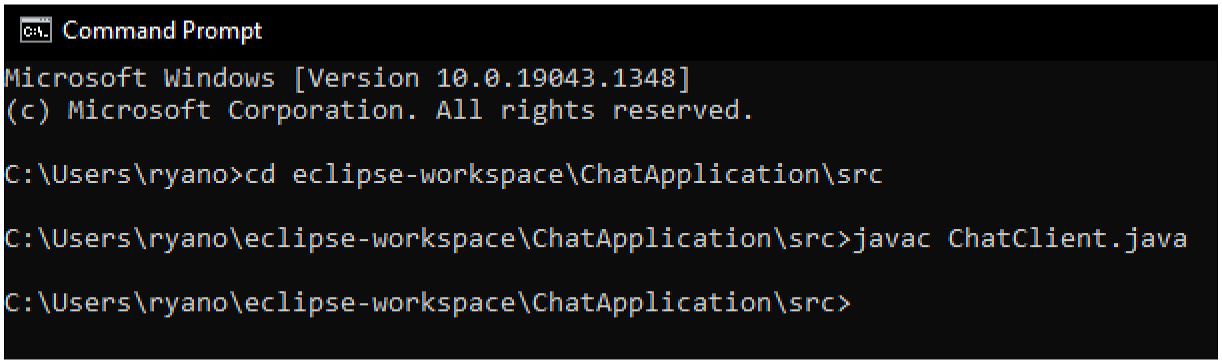
   Steps 5 to 8 will involve setting up the client(s).

5. Repeat Step 2 to have another command prompt window with the directory where the folder with the three java files is stored in.



```
Command Prompt
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ryano>cd eclipse-workspace\ChatApplication\src

C:\Users\ryano\eclipse-workspace\ChatApplication\src>
```

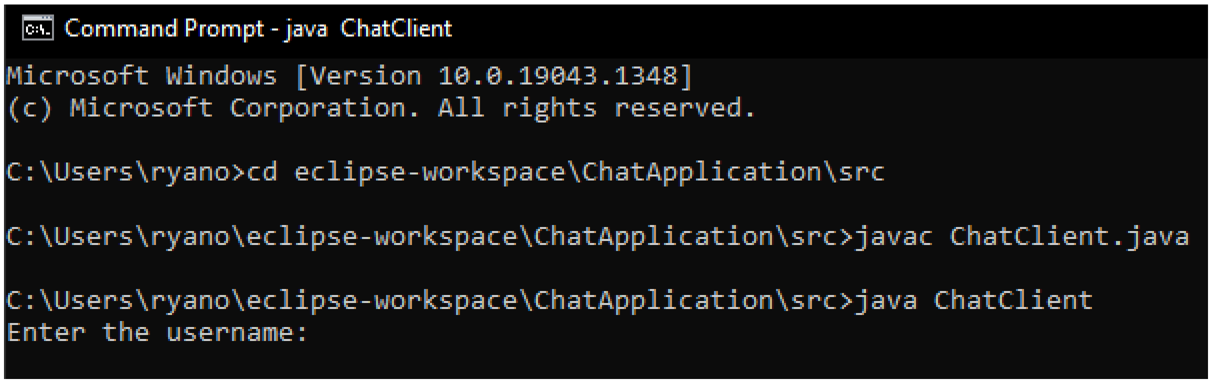
6. Type “javac ChatClient.java” and press Enter.



```
Command Prompt
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ryano>cd eclipse-workspace\ChatApplication\src

C:\Users\ryano\eclipse-workspace\ChatApplication\src>javac ChatClient.java

C:\Users\ryano\eclipse-workspace\ChatApplication\src>
```

7. Type “java ChatClient” and press Enter to run the client.

If the below message shows up, this means the client is currently looking for a username.



Command Prompt - java ChatClient

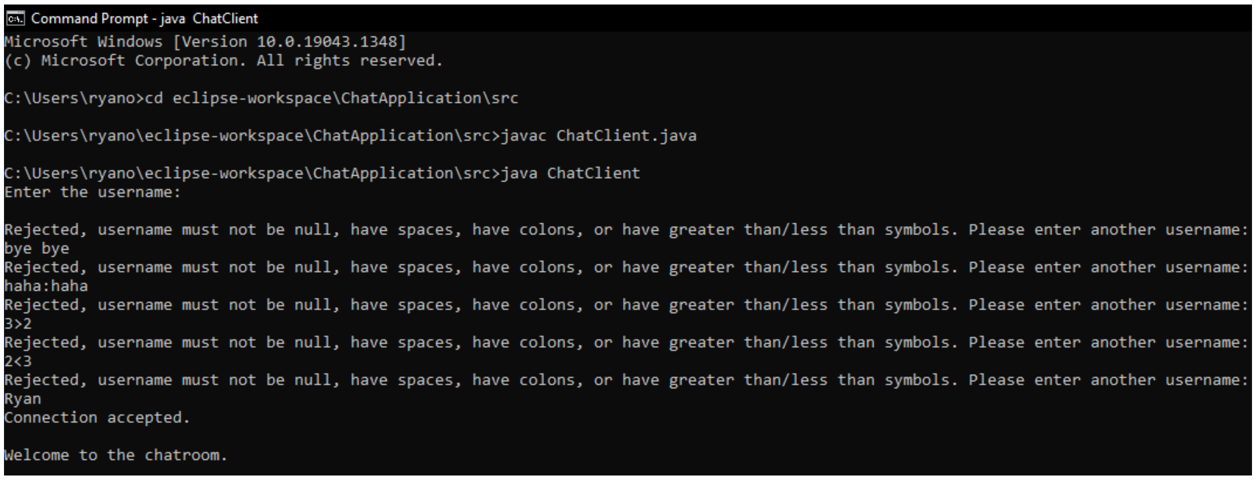
```
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ryano>cd eclipse-workspace\ChatApplication\src

C:\Users\ryano\eclipse-workspace\ChatApplication\src>javac ChatClient.java

C:\Users\ryano\eclipse-workspace\ChatApplication\src>java ChatClient
Enter the username:
```

8. Repeat steps 5 to 7 until you have as many clients as you desire for testing.

   Note: If one or more of the clients do not seem to be responding or updating when testing multiple clients, try clicking on the corresponding command prompt window and typing anything in.

   Steps 9 to 11 will involve the testing of the code

9. Type in a username and press enter. If the username is invalid, it will prompt the user to input another one.

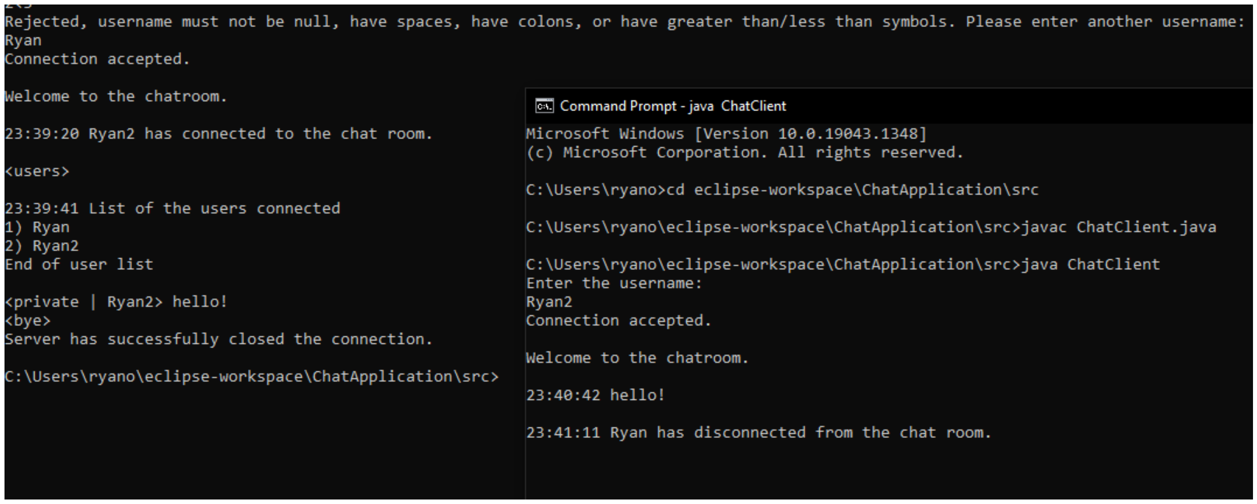


Command Prompt - java ChatClient

```
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ryano>cd eclipse-workspace\ChatApplication\src

C:\Users\ryano\eclipse-workspace\ChatApplication\src>javac ChatClient.java

C:\Users\ryano\eclipse-workspace\ChatApplication\src>java ChatClient
Enter the username:

Rejected, username must not be null, have spaces, have colons, or have greater than/less than symbols. Please enter another username:
bye bye
Rejected, username must not be null, have spaces, have colons, or have greater than/less than symbols. Please enter another username:
haha:haha
Rejected, username must not be null, have spaces, have colons, or have greater than/less than symbols. Please enter another username:
3>2
Rejected, username must not be null, have spaces, have colons, or have greater than/less than symbols. Please enter another username:
2<3
Rejected, username must not be null, have spaces, have colons, or have greater than/less than symbols. Please enter another username:
Ryan
Connection accepted.

Welcome to the chatroom.
```

Note: I unfortunately have not been able to successfully implement the proper rejecting of duplicate usernames. I set a requirement to the username of not being able to have spaces, have colons, have greater than/less than symbols, and not being able to be null or “”. This is because setting these restrictions allow my code to run more consistently and smoothly.

10. Once the username is accepted, then the other commands can be tested as well. The commands are “<users>”, “<private | user> message”, and “<bye>”. Type in anything else to send a normal message.



```
Rejected, username must not be null, have spaces, have colons, or have greater than/less than symbols. Please enter another username:
Ryan
Connection accepted.

Welcome to the chatroom.

23:39:20 Ryan2 has connected to the chat room.

<users>

23:39:41 List of the users connected
1) Ryan
2) Ryan2
End of user list

<private | Ryan2> hello!
<bye>
Server has successfully closed the connection.

C:\Users\ryano\eclipse-workspace\ChatApplication\src>
```

Command Prompt - java ChatClient

```
Microsoft Windows [Version 10.0.19043.1348]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ryano>cd eclipse-workspace\ChatApplication\src

C:\Users\ryano\eclipse-workspace\ChatApplication\src>javac ChatClient.java

C:\Users\ryano\eclipse-workspace\ChatApplication\src>java ChatClient
Enter the username:
Ryan2
Connection accepted.

Welcome to the chatroom.

23:40:42 hello!

23:41:11 Ryan has disconnected from the chat room.
```

Note: The commands “<users>” and “<bye>” will only work if they are the only string in the message and have no other characters with them (“<users> hi” and “lol <bye>” will just be sent as normal messages). Also, the private message command must have the exact syntax shown to work, or it will send as a normal message as well (must have a space before and after the “|”, and another space after “>”. “<private|User1> hello” will not work and “<private | User1>hello” will not work, but “<private | User1> hello” will work). The private message command will only count the characters after “<private | user> ” as the message to be sent privately and will ignore any characters that come before it.

11. When the testing of the other commands have been completed and you wish to close the client(s), you may exit by either typing “<bye>” in the command prompt and pressing Enter or clicking the X at the top right of the command prompt window to exit.

    Final Note: Using Eclipse to run ChatServer, then run multiple instances of ChatClient will also work.