

Eclipse File Edit Source Refactor Navigate Search Project Run Window Help

JavaWorkspace - NETS150_final/src/Client.java - Eclipse IDE

Quick Access

HW6
HW7
HW8
HW9
ir
jsoup
multi_threading
NETS_finalProject
NETS150_final
JRE System Library [JavaSE-1.8]
src
 (default package)
 Client.java
 Match.java
 MusicCorpus.java
 MusicDocument.java
 MusicMatcher.java
 MusicVectorSpaceModel.java
 Profile.java
 RestaurantFinder.java
 Server.java
 ShowCorpus.java
 ShowDocument.java
 ShowMatcher.java
 ShowVectorSpaceModel.java
 Test.java
Referenced Libraries

Client.java

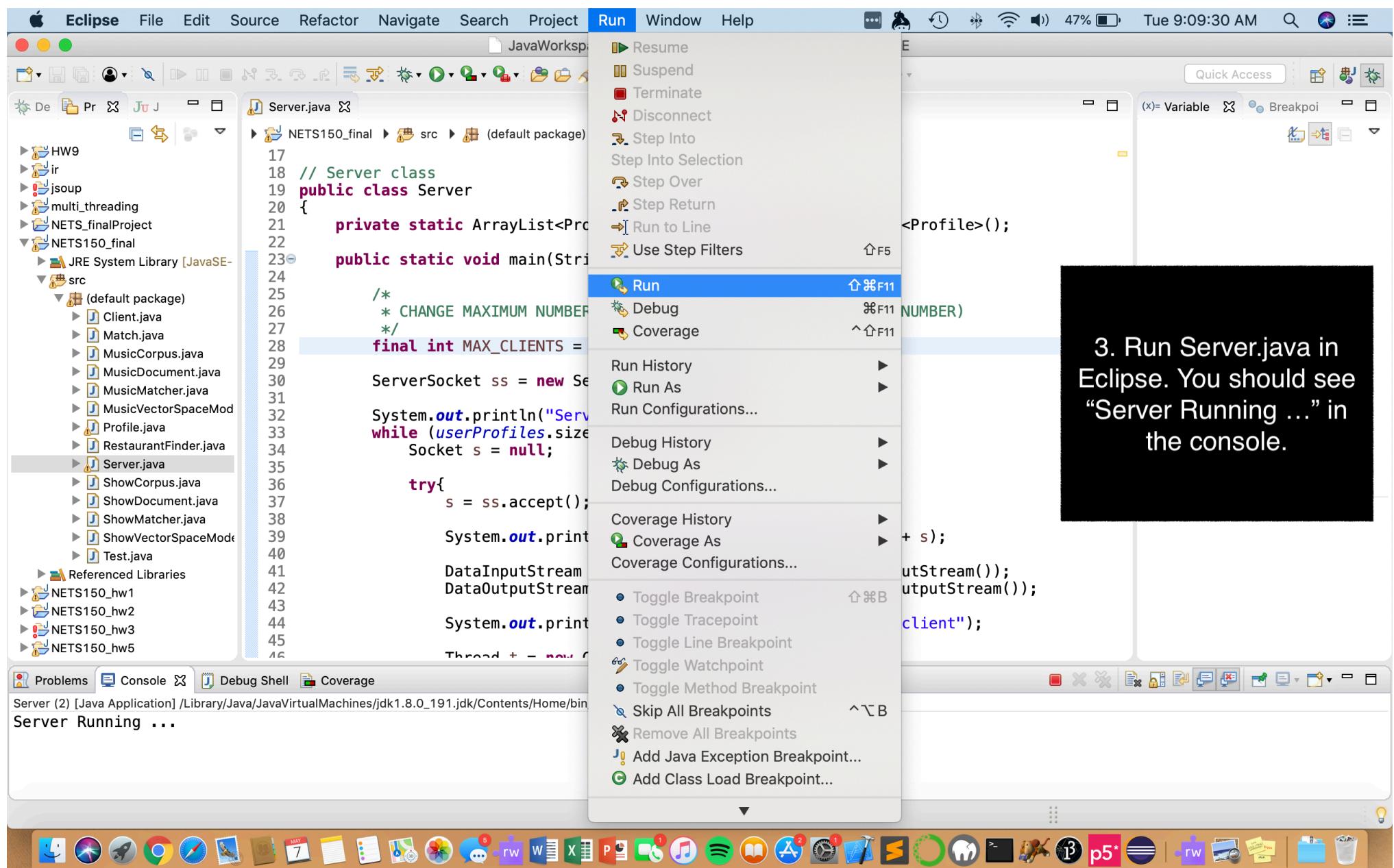
```
2+ * Help from: 
9
10+import java.io.*;
13
14 // Client class
15 public class Client {
16     public static void main(String[] args) throws IOException {
17         try {
18             Scanner scn = new Scanner(System.in);
19
20             /*
21             * ENTER CORRECT IP ADDRESS HERE
22             */
23             Socket s = new Socket("158.130.175.33", 4000);
24
25             // obtaining input and out streams
26             DataInputStream dis = new DataInputStream(s.getInputStream());
27             DataOutputStream dos = new DataOutputStream(s.getOutputStream());
28
29             // the following loop performs the exchange of
30             // information between client and client handler
31             while (true) {
32                 System.out.println(dis.readUTF());
33                 String tosend = scn.nextLine();
34                 dos.writeUTF(tosend);
35
36                 // If client sends exit, close this connection
37                 // and then break from the while loop
38                 if (tosend.equals("exit")) {
39                     break;
40                 }
41             }
42         } catch (IOException e) {
43             e.printStackTrace();
44         }
45     }
46 }
```

2. Open Client.java, make sure the Port matches the Server's port, and the IP address matches the IP of the computer running the server.

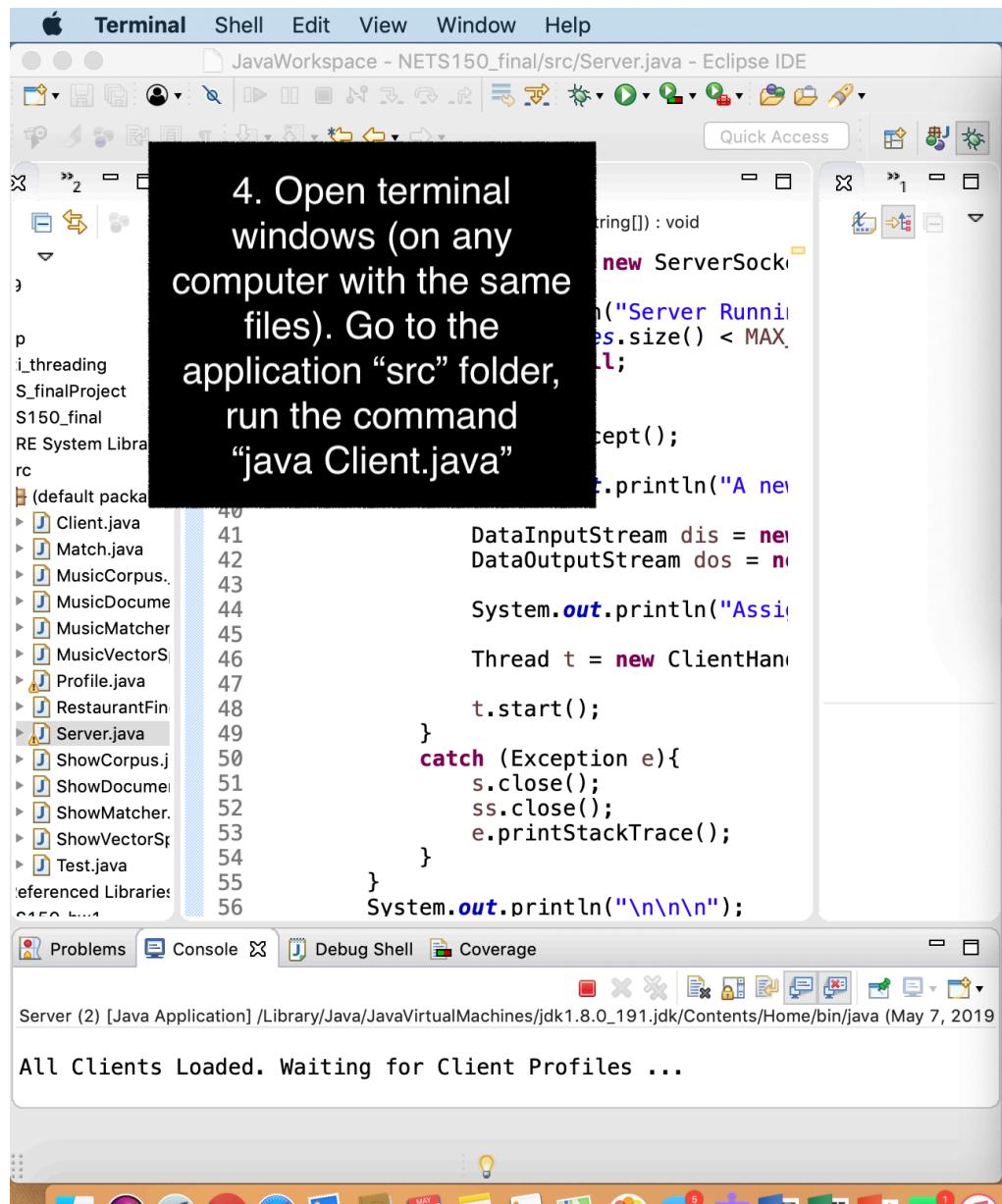
Problems Console Debug Shell Coverage

Use this area to write, run and inspect Java expressions in the context of a debug session.

Mac OS X Dock Icons



4. Open terminal windows (on any computer with the same files). Go to the application “src” folder, run the command “java Client.java”

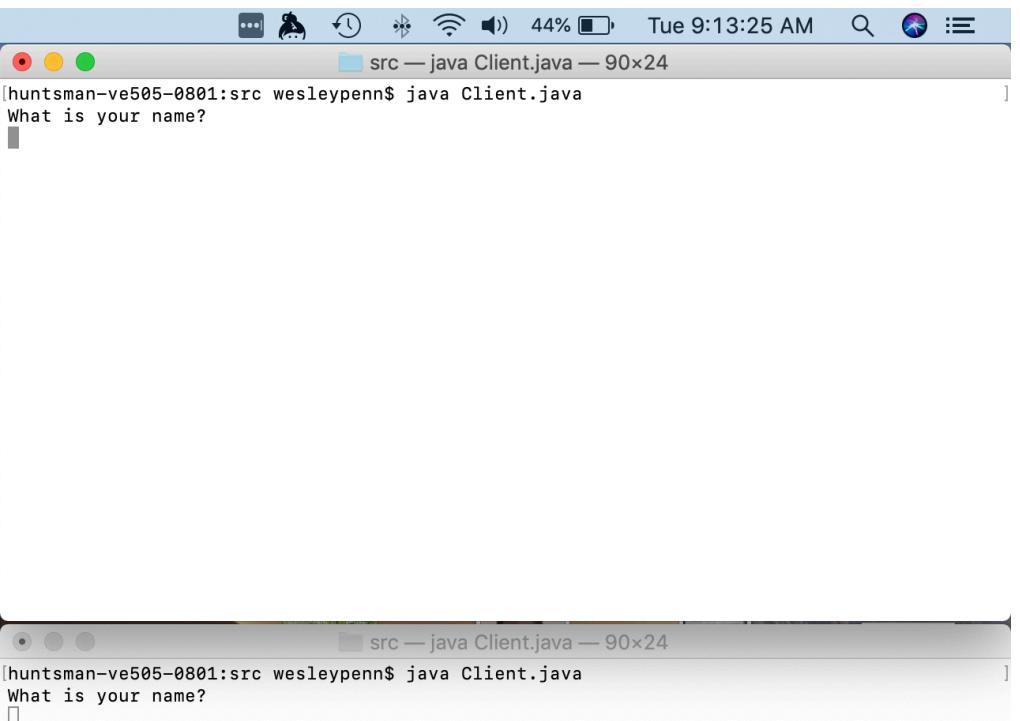


The screenshot shows the Eclipse IDE interface. The title bar reads "JavaWorkspace - NETS150_final/src/Server.java - Eclipse IDE". The main area displays the Java code for the Server class:

```
public class Server {
    public static void main(String[] args) {
        new ServerSocket();
        System.out.println("Server Running");
        while(true) {
            try {
                Socket s = ss.accept();
                ClientHandler t = new ClientHandler(s);
                t.start();
            } catch (Exception e) {
                s.close();
                ss.close();
                e.printStackTrace();
            }
        }
        System.out.println("\n\n\n");
    }
}
```

The code uses a while loop to accept connections from clients. It creates a new thread for each client using a ClientHandler object. The ClientHandler class is defined in another file.

Below the code editor, the Eclipse interface includes tabs for "Problems", "Console", "Debug Shell", and "Coverage". The status bar at the bottom indicates "Server (2) [Java Application] /Library/Java/JavaVirtualMachines/jdk1.8.0_191.jdk/Contents/Home/bin/java (May 7, 2019)".



The screenshot shows two terminal windows side-by-side. Both windows have the title "src — java Client.java — 90x24" and the command "java Client.java" running. The top window shows the prompt "What is your name?". The bottom window also shows the same prompt. This indicates that both terminals are waiting for user input.

5. After all the Client's join the Server, it will print "All Clients Loaded". The Server will prompt the client windows with questions. Type your answer to the Client window, then Press Enter. This will add to the Client's profile.

Note: You must put in exact answers into the console. Otherwise, the program will fail.

i.e. Office
→ The Office US

Kanye → Kanye West

Only worry about Music and TV shows. If the program fails, then it is most definitely an issue with the input (TV or Song)

```
45
46
47
48
49
50
51
52
53
54
55
56
```

```
    Thread.sleep(1000);
    t.s.close();
}
catch (IOException e)
{
    System.out.println("An error occurred while reading from the socket.");
    e.printStackTrace();
}
System.out.println("\n\n\n");
```

Terminal Shell Edit View Window Help

JavaWorkspace - NETS150_final/src/Server.java - Eclipse IDE

src — java Client.java — 90x24

[huntsman-ve505-0801:src wesleypenn\$ java Client.java

What is your name?
Wesley
Response: Wesley
What is your favorite food (or food genre)?
Pizza]

src — java Client.java — 90x24

[huntsman-ve505-0801:src wesleypenn\$ java Client.java

What is your name?

Mac OS X Dock icons

6. When all the clients input their information, then the Server will return the Best Matches of each Client. This will be output to the console.

The screenshot shows the Eclipse IDE interface with the following components:

- Top Bar:** Terminal, Shell, Edit, View, Window, Help.
- Toolbar:** Standard Eclipse tools like Open, Save, Run, Stop, etc.
- Left Sidebar:** Shows the project structure with files like Client.java, Match.java, MusicCorpus.java, etc., under the S150_final project.
- Code Editor:** Displays the `Server.java` file. The code implements a TCP server that listens on port 8080. It uses `ObjectInputStream` and `ObjectOutputStream` to handle client connections. It prints "A new client connected" to the console and creates a new thread for each client using a `ClientHandler`.
- Console Tab:** Shows the output of the terminated server application.
- Bottom Bar:** Shows various Mac OS X system icons for battery, signal, volume, etc.
- Two Terminal Windows:** Both titled "src — java Client.java — 90x24".
 - Top Terminal:** Shows a conversation between the server and a client named "Jon". The server asks for the user's name, favorite food, favorite music artist, favorite song, favorite TV show, favorite season, and favorite episode.
 - Bottom Terminal:** Shows a conversation between the server and a client named "Wesley". The server asks for the user's name, favorite food, favorite music artist, favorite song, favorite TV show, favorite season, and favorite episode.

Server.java Code Snippet:

```
System.out.println("A new client connected");
        Thread t = new ClientHandler(dis, dos);
        t.start();
    } catch (Exception e) {
        s.close();
        ss.close();
        e.printStackTrace();
    }
}
```

Console Output (Top Terminal):

```
[huntsman-ve505-0801:src wesleypenn$ java Client.java
What is your name?
Jon
Response: Jon
What is your favorite food (or food genre)?
Italian
Response: Italian
What is the name of your favorite music artist?
Kanye West
Response: Kanye West
What is your favorite song by them?
All of the lights
Response: All of the lights
What is your favorite TV Show?
The Office US
Response: The Office US
What is your favorite Season from the Show?
3
Response: 3
What is your favorite Episode from that Season?
4
Response: 4
Profile created, exiting program ...]
```

Console Output (Bottom Terminal):

```
[huntsman-ve505-0801:src wesleypenn$ java Client.java
What is your name?
Wesley
Response: Wesley
What is your favorite food (or food genre)?
Yogurt
Response: Yogurt
What is the name of your favorite music artist?
Drake
Response: Drake
What is your favorite song by them?
One Dance
Response: One Dance
What is your favorite TV Show?
Suits
Response: Suits
What is your favorite Season from the Show?
2
Response: 2
What is your favorite Episode from that Season?
5
Response: 5
Profile created, exiting program ...]
```