How to get a full mark in Programming Assignment 2

Grading Scheme

PA2 will be counted towards 10% of your final course grade. The maximum scores for different tasks are as follows:

- Implementation: 70%
- Document (Project Report): 30%

In your implementation (70%), if you finish all the requirements proposed, and works well after we check, you can get a full mark in the implementation part. The details are as following:

1. You should implement chatting server (10%) and chatting client (10%) by using GUI programming.
2. For server-side:
    a) Listen on a specific port, and wait for connections required from clients. (5%)
    b) Users can configure the listening port of server-side, while the default port is 8888. (5%)
    c) Send a system level message to online users who have connected to the server-side. (5%)
    d) Count the number of the current online users. (5%)
    e) When stopping the chatting service, disconnect all users' connections. (5%)
3. For client-side:
    a) Connect to the server which has provided the chatting service successfully. (5%)
    b) User can configure the IP address and port number of server-side to be connected. (5%)
    c) Users can configure the username for displaying after connection. (5%)
    d) When the server-side is enabled, users can log on and log off at any time. (5%)
    e) User can send messages to all users or a single person in a whisper. (5%)

In your document, you should write clearly how you implement the assignment. Also, you need to show some results: for one on one dialog, or multiple users online conversation. The score for the document also includes: code quality, e.g., good design of the program, good documentation, good coding style, proper exception handling, etc.

The above breakdown is only a guideline. We reserve the right to grade your submission differently in exceptional cases – we will explain to you individually why your case is exceptional if this happens to your submission.

Some common issues:

1. What kind of content should be contained in Help Class: Any tips or comments of your software are fine, such as:
   help.setText("1. Set Server IP Address and Port No"+ "(Default Config\n 127.0.0.1:8888).\n"+ "2. Enter your user name (Default is hkuster).\n"+ "3. Click Logon to Connect to Server; \n"+ " Click Logout to disconnect to Server.\n"+ "4.

Select user to receive messages and enter message in the message box\n"+ ". Meanwhile, select emotion and send messages.\n");

2. Some students encounter that when using ObjectInputStream and ObjectOutputStream to do the data transfer between server and client, the program stucks at somewhere like new ObjectInputStream(client.getInputStream()) and cannot process further. In this case, You need to create the ObjectOutputStream before the ObjectInputStream at both sides of the connection. When the ObjectInputStream is created, it tries to read the object stream header from the InputStream. So if the ObjectOutputStream on the other side hasn' t been created yet, there is no object stream header to read, and it will block indefinitely. Or phrased differently: If both sides first construct the ObjectInputStream, both will block trying to read the object stream header, which won' t be written until the ObjectOutputStream has been created, which will never happen since both sides are blocked in the constructor of ObjectInputStream.
3. Some students' program will throw out the error that: java.io.NotSerializableException. This may due to that you use a class to represent the information being communicated between the server and the object. To avoid this problem, you do not need to use a new class to construct the information. You can just use the readObject() several times to read from client or sever since only objects that support the java.io.Serializable or java.io.Externalizable interface can be read from the streams. For more details can be found here: https://docs.oracle.com/javase/7/docs/api/java/io/ObjectInputStream.html