Overview

For this assignment you will need to create a simple network echo server that communicates with a provided network client. The client will connect to your echo server and send the line inputted to the echo server. This assignment is intended as a very simple primer to familiarize you with the concepts of network hosts, TCP sockets, and ports. Keep in mind that one third (30%) of your grade is related to your source code, so make sure to create comments and use small, well-defined methods when appropriate. This assignment is expected to take around 2 hours to complete, though it may take more or less time depending on your abilities and background. It is due BEFORE CLASS on Monday, September 21. You must submit via the Sakai Assignments tool. Only on-time assignments submitted through Sakai will be accepted.

Specifics of Operation

Your program MUST be completely contained in a file called “EchoServer.java" without any Java packaging. This means that compiling and running your program must be identical to the output below:

javac -cp .
EchoServer.java java -cp .
EchoServer 1234

If you are unfamiliar with the concepts of Java packages or class paths, you should see your Teaching Assistant during recitation or office hours. Within your class, you should create multiple methods in addition to the static main(String[]) method. An example skeleton class will be provided during recitation to assist you, though you do not need to use this file as long as your class has the same name (EchoServer). To successfully complete this assignment, your program must do the following:

1. Accept the echo server port number as args[0] in the main method. The port should be parsed into an int orInteger.
2. Construct a server socket and listen for incoming connections from the echo client based on the port specified in the command-line arguments.
3. Read the message that the client has sent to you and reply to the client with the inverted string received including any new line characters (CR/LF).
4. Read a single line response from the client and print it toSystem.out.
5. Close the connection between the client and the server when the clientsends # or $ as a line of text.
6. Close the Socket to the client and any IOStreams that you mayhave opened during the program's lifetime.
7. Exit your program. At no point in your application should you callSystem.exit(int).

A class file containing an appropriate client is available for you for testing. This is the same client that will be used during grading on the iLab computers. During recitation, an outline of a client will be provided to assist you if you are unfamiliar with Java programming. The class file is executable using the following command:

java TCPClient 127.0.0.1 1234

In this case, "1234" is the port number for the client to connect to. And “127.0.0.1” is the IP address of your server. You may choose any port you wish, but be aware that ports below 1024 are reserved for the "root" user in most operating systems.

Grading Outline

Your submission will be graded on one of the iLab machines located in the Hill Center. If your code does not compile or run correctly in this environment, you will not receive credit for the portions that fail. You should write or test your program on the iLab computers; future assignments will also be graded there and you can get any problems/issues worked out early.

- Source code organization and comments: 30%
- Reading command-line parameters: 10%
- Establishing socket to the server based on parameters: 20%
- Reading user input and sending it to the server: 15%
- Reading server response and printing it to the user: 15%
- Closing all Sockets/IOStreams before exit: 10%

Hints and Suggestions

If this is your first time working directly with Sockets, IOStreams, and Strings, you may find the following classes/resources helpful:

- Java Tutorials Lesson: Basic I/O (oracle.com) - http://docs.oracle.com/javase/tutorial/essential/io/
- I/O From the Command Line (oracle.com) - http://docs.oracle.com/javase/tutorial/essential/io/cl.html
- Java Tutorials Lesson: All About Sockets (oracle.com) - http://docs.oracle.com/javase/tutorial/networking/sockets/
- Reading from and Writing to a Socket (oracle.com) - http://docs.oracle.com/javase/tutorial/networking/sockets/readingWriting.html