**Distributed Systems – Assignment 1**

**Name: Brittney Desroches Student Number: 100649514**

**Code:**

**Program Description**

For the purpose of this assignment, I created a Java client/server that uses TCP sockets for communication. The main functionality of this program is as a number-system converter, it is able to convert binary values to decimal and decimal values to binary. After starting the program, giving the port number and host name as the start-up arguments, the program asks for the input values, and makes the user aware of the ‘help’ command if they require assistance in using the program. User input is taken through the client console, and is to be entered in the form of three variables; current type, new type, and the value. If the user was to mess up the input, whether it be a typo in the type names, the incorrect number of arguments, or if the number does not match the ‘current type’ value, messages will be sent to the client to indicate these issues. The client is capable of sending several requests to the server one after the other, and can disconnect from the server using the ‘exit’ command. There are two additional features for this program. The first feature tells the user how many lines of code the program needs to read to complete the conversion, this does not include the error checking the program does, only the lines read in the conversion methods. The second feature is a timer to show the user how long it took the program to work through those lines of code in milliseconds.

**Challenges/Solutions**

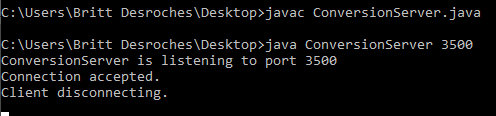
Originally, the program's additional features were going to be a help command and a line counter. However, after a bit of thought, it was clear that the help command should not be a special feature, but rather it was an important part of the program. Having a command to assist users in using the program is an integral part of the usability of the application. As such, I included the help command, then also added the duration timer as the true second feature.

Another challenge I had was the sending and receiving of messages to and from the server. My goal was to ensure the least number of messages being sent, such as sending all of my user input as a single string and breaking it apart on the server side. However, this was not quite as simple when sending the messages back as there were several possible things to return where there was an issue with the conversion, or just the two different conversions in general. Binary to decimal method returned an integer while decimal to binary returned a string. Due to this, separate write statements had to be made which caused issues when I occasionally needed to write once rather than twice. My solution was not the most elegant as during the instances that I only wrote once to the client rather than twice, I snuck a ‘\n’ at the end of the line I was returning such that the second reader.readLine() would read and output just a new line.

**Sample Run Screenshots:**

Server:

The server announces when a client has connected and disconnected from the server.



Client:

The client here shows each of the options the program has to offer. First, it shows the help command which gives context to the possible inputs to the program, then it shows a binary to decimal conversion followed by a decimal to binary conversion. Finally, the user exits the client and the server says goodbye.

