Task #1: TCP Server

1. To accomplish the task, I first read input from the input stream (keyboard) on the client side then stored it. Next, I sent the message to the server via a print stream.

   On the server side, I received the message from the client by reading from the client data input stream, then stored it. Next, I computed the summation from 0 to the number via a "for" loop. The average of the summation was computed afterwards and then sent back to the client via the print stream (client output stream).

   Lastly, all that needed to be done was to receive the server message on the client side via the client input stream, then display the computed average on the client.

2. A challenge I encountered was sending the proper message (number) to the server, and not the number's memory address. I solved this problem by reading the user's input as a string, then parsing the string as an integer.

3. As this was my first sockets programming challenge, I learned how to send and receive messages via sockets between a client and a server.

Task #2: Reversing a String

1. To accomplish the task, I first read input from the input stream (keyboard) on the client side then stored it. Next, I sent the message to the server via a print stream.

   On the server side, I received the message from the client by reading from the client input stream, then stored it. Next, I reversed the string via a "for" loop then sent the modified string back to the client via the print stream (client output stream).

   Lastly, all that needed to be done was to receive the server message on the client side via the client input stream, then display the reversed string.

2. A challenge I faced was receiving the message on the server. I had originally tried to read the input stream inside of a loop; however, I realized this was not necessary and therefore removed the loop.

3. I learned how to send text via sockets between a client and a server.

Task #3: Repeated Inputs from Client

1. To accomplish the task, I first read input from the input stream (keyboard) on the client side then stored it. Next, I sent the message to the server via a print stream, in a "while" loop.

   On the server side, I received the message from the client by reading from the client input stream, then stored it. Next, I reversed the string via a "for" loop then sent the modified string

back to the client via the print stream (client output stream). All of this was done in a “while” loop.

Lastly, all that needed to be done was to receive the server message on the client side via the client input stream, then display the reversed string; this was done in the same “while” loop on the client side.

2. The only challenge I faced was determining which portions of code should be inside the “while” loops.

3. I learned how to continuously send text via sockets between a client and a server.

Task #4: Multithreading

1. The same procedure was followed as task #3 to allow for continuous sending of messages. However, like task #1, the system will handle sending and receiving numbers as messages. Since the system needed to process requests from multiple clients, a thread had to be set up for each client request. Furthermore, since the server also needed to handle a fixed number of clients, there was a limit as to how many threads may be opened.

2. The main challenge was to figure out how to set a limit on the amount of clients allowed to communicate with the server.

3. I learned how to set up communications between a client and a server using sockets and multithreading.