1. This lab consists of two programs that run simultaneously – a client and a server. These two programs communicate by connecting and then binding a socket. The client portion of the code can only run if the server-side program has binded the socket to a memory address. After that, the client sends information across the socket, and the server will run if there is something to receive. The server then manipulates this information and sends it back via the socket through a *write()* call. The program runs in this way until there is no more information to send or receive.

2. The handshake socket is necessary to ensure that a a proper connection is established. In this way, more than one client can connect to the server at a time. Otherwise, one of the programs would be attempting to send or receive information before there is another program on the other side of the socket connection to receive or send information.

3. Sockets are preferred to pipes in this lab for a number of reasons. For one, socket communication does not have to be on the same host machine, but can be across a network. Secondly, pipe communication is unidirectional. Both the client and server program read and write to each other via the socket. We would need two pipes to accomplish the same task. Furthermore, sockets allow for multiple connections.