Networking with Sockets

```java
//get the next line of input
curLine = fRemoteInputStream.readLine();
}
catch (IOException readEx) {
    //we expect this throw when we reach EOF
    //on the stream
    curLine = null;
}
}
while (curLine != null);
System.out.println("<EOF>"); //dump a finished note to stdout
*/

class FingerClient {

To test the FingerClient class, we've provided a simple stand-alone class that takes parameters from stdin and gives them to FingerClient. When we tested this class, we invoked

FingerClientTest carolee@lungfish.com

...and verified that we received results in System.out similar to our previous simulated Finger session. You might want to test this class with the target string poke@l.gp.cs.cmu.edu" to see an interesting example of one of the vending machines still attached to the Internet via a Finger server.

Meet the ServerSocket Class

The ServerSocket is class complementary to the Socket class. It provides listening socket services for a server that complement the connecting socket services for a client. The main difference between the ServerSocket interface and the Socket interface is that the ServerSocket provides a way to wait for an incoming connection. Thus, as its name suggests, the ServerSocket allows the creation of servers.