

Exercises: Network Clients

Note: there is no need to use my NetworkClient class for these exercises; you could easily write everything from scratch. Using NetworkClient will just save you a few lines of code. If you do use it, you would need to copy NetworkClient.java and SocketUtils.java into your Eclipse project, then do something like this:

```
public class MyClient extends NetworkClient {
    public MyClient(String host, int port) {
        super(host, port);
    }

    @Override
    public void handleConnection(Socket s) throws IOException {
        PrintWriter out = SocketUtils.getWriter(uriSocket); // Omit if not used
        BufferedReader in = SocketUtils.getReader(uriSocket);
        // Your code here
    }
}
```

Then, your driver routine (the one that has “main”) would do:

```
MyClient client = new MyClient(someHost, somePort);
client.connect(); // Connect now
```

- 1.** Connect to `time-c.nist.gov` on port 13 and read the two-line result. Ignore the first line and print out the second line to show the current time in GMT. Major hint: you can do this problem by copying one of the examples from the lecture notes and adding *one* line of code to it. If that host is overloaded and refuses connections, try one of the other servers listed at <http://tf.nist.gov/tf-cgi/servers.cgi>
Before writing your Java client, be sure to telnet to the site to verify you can reach it.
- 2.** Make a program that accepts the name of an FTP server, connects to that host on port 21, and prints out the first line of the welcome message. You can try `ftp.microsoft.com`, `ftp.cisco.com`, or `ftp.intel.com`: all of those have welcome messages as of early 2016.
- 3.** Make a program that simply checks if a given URL exists or not. Your program will be similar to the example from the notes, but with two changes:
 - Send HEAD instead of GET, so as not to bog down the server. The HEAD command returns the HTTP response headers, but not the actual document.
 - Print out the first line to see if the result is good or bad. Or, look at the first line of the response, take the second token, and see if it is 200 (good) or something else such as 404 (bad).
- 4.** Make a program that connects to a Web page and determines if the content of that Web page contains the word “Java”. Or, even better, make the term it searches for configurable.