Additional Guidelines

The first week of lab session is completed, and we expect you to have started the assignment. This document answers the frequently asked questions that practical supervisors received in the first week.

- **Assignment Update: The server implementation should support the "400: Bad Request" status code, when the HTTP Client does not include the host header in its request for HTTP version 1.1.**
- The assignment should be built either in Java version 1.6 or 1.7.
- Do not use HTTPURLConnection library to establish the connection with a remote HTTP server.
- URI parsing may be done using the URI library of Java. However, you should not use the URL library to connect to a HTTP server. It is sufficient to support the URI of the format www.example.com instead of http://www.example.com.
- You can choose any parser to scan the html files, but you are not allowed to use any API's from the parser library to retrieve the embedded objects. The parser should be used only to detect the embedded object tags and the associated URIs. You should use GET method to retrieve the embedded objects from the HTTP server.
- For HTTP/1.0, a new connection should be made for each embedded objects.
- For HTTP/1.1, both the html content and the embedded objects should be received using the same connection.
- While retrieving the relative path embedded objects, the retrieved objects should be stored in their own local relative path in the server.
- It is not obligatory to download the absolute path embedded objects.
- In the case of HTTP/1.1, the termination of the connection is done using the **Connection: close** header. The server closes the connection after handling the **Connection: close** header **or after some timeout period.**
- **Chunked Encoding** is **not mandatory** for this assignment. Your client should retrieve the html pages from the servers that do not use the chunked encoding scheme.

- You can use the following URI’s to test your client program:
    - www.example.com
    - www.tcpipguide.com
    - www.jmarshall.com
    - www.tldp.org
    - www.tinyos.net
    - www.linux-ip.net
- Your program should take the following inputs from the user: HTTPCommand, URI, Port and HTTP version. You can either execute it in JAVA IDE (e.g. eclipse) or in a command line terminal.
- For the PUT command, the user input should be stored in a new text file on the server.
- For the POST command, the user input should be appended to an existing file on the server. If the file does not exist, then the file should be created.
- Your HTTP Server should handle multiple clients at the same time. Create multiple tabs in your web browser to test your HTTP server.
- The client should send the If-Modified-Since header using the Last-Modified response information. While demonstrating the client program, the first request to a specific URI should be issued without the If-Modified-Since header, but the subsequent GET requests for the html pages should send the If-Modified-Since header. You should discuss this with your teaching assistant, if it is not clear.