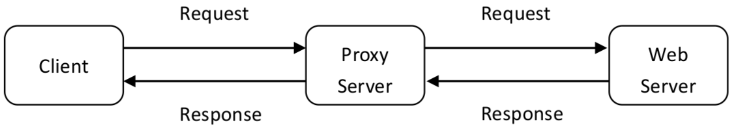
Assignment 1:
Sci-fi Sports Proxy Server

due: **Friday, October 02, 2020 (11:59pm)**

HTTP Web Proxy Server (40 Marks)

Learning Objectives. In this assignment, you will develop a small web proxy server. You will learn the basics of socket programming in Java: how to create a socket, bind it to a specific address and port, as well as how web proxy servers work. You will also learn some basics of the HyperText Transfer Protocol (HTTP) used by the World Wide Web.

Assignment Description. A web proxy is an intermediary that intercepts each and every request and (generally) forwards it on to the web server. The servers direct their responses back to the proxy, which in turn passes them on to the client. In this way, a web proxy acts as a middle-person between a client and a server. Here's the neat part, though; when requests and responses travel through a proxy, the proxy can control what gets passed along and modifies the requests and responses.



We are going to make a Web proxy that alters certain content on simple Web pages before they are rendered by the Web browser, so that the user sees factually incorrect information without knowing it. To keep the assignment simple, we will restrict ourselves only to HTTP (not HTTPS), and consider only basic HTML pages with a few images.

For our purpose, you program should apply these changes:

- 2019 to 2219
- NBA to TBA

- World to Titan
- Drummond to Kobe-B24

These modification applies to all text, hyperlinks, and images.

Skeleton code. The code is divided into three classes as follows:

- Proxy holds the start-up code for the proxy and code for handling the requests.
- HttpRequest contains the routines for parsing and processing the incoming requests from clients.
- HttpResponse takes care of reading the replies from servers and processing them.

Your work will be to complete the proxy so that it is able to receive requests, forward them, read replies, modify, and return those to the clients. You will need to complete the classes Proxy, HttpRequest, and HttpResponse.

Testing. Configure your browser to use your proxy, run the proxy server program using your command prompt and then request a web page from your browser.

Running the proxy is as follows:

. **java Proxy port**

where the port is the port number on which you want the proxy to listen for incoming connections from clients. To use the proxy server with browser and proxy on separate computers, you will need the IP address on which your proxy server is running. In this case, while running the proxy, you will have to replace the "localhost" with the IP address of the computer where the proxy server is running. Also note the port number used.

Your proxy will be tested on five pages in the following URL:

Test Pages (main page and four embedded pages)

Configuring your Browser. You can also directly configure your web browser to use your proxy. This depends on your browser. In the proxy setting of your browser, you need to give the address of the proxy and the port number that you gave when you ran the proxy server. Ask your TA for more details.

What to Upload in D2L You will upload the complete proxy server code in a single zip file (*proxy.zip*) and one-page user manual (PDF) in D2L, Assignment 1.

Grading. You will upload programs in a single zip file. Also you need to provide a README file (PDF) for your TA, describing how to run your code.

- **26 marks** for the design and implementation of a functional Web proxy that modifies page based on the assignment description. Your implementation should include reasonably commented code.
- **4 marks** for a clear and concise user manual (at most 1 page) that describes how to compile, configure, and use your Web proxy. Make sure to clarify where and how the testing was done (e.g., local host, CPSC Machine), what works, and what does not. Be honest!
- **10 marks** for a suitable demonstration of your server to your TA in your tutorial section. A successful demo will include marks for the test cases above, as well as clear answers to questions asked during your code walk-through.

  Up to 2 bonus marks will be awarded for adding a proper CSS file to apply on the modified Web pages.

Tips.

- This is a rather challenging assignment, so please get started early. You will likely need at least a week to get it fully working.
- If you have never done socket programming before, you should make sure to attend the tutorials.
- Java separates the input streams according to whether they are text-based or binary, which presents a small problem in this case. Only DataInputStreams are able to handle both text and binary data simultaneously; all other streams are either pure text (e.g., BufferedReader), or pure binary (e.g., BufferedInputStream), and mixing them on the same socket does not generally work.
- Focus on the basic HTTP proxy functionality first, by simply forwarding everything that you receive from the client directly to the server, and everything you receive from the server directly back to the client. Then add more functionality.
- Your proxy will need one socket for talking to the client, and another socket for talking to the server. Make sure to keep track of which one is which!
- Your proxy will actually need to dynamically create a socket for every new server that it talks to. Make sure to manage these properly.
- Start with very simple text-based HTML files, such as first test file. Once you have these working, then you can try more complicated Web pages with embedded objects.

Good Luck!