CSI 3140 – Laboratory 4 - Servlets

1- Setting up your environment

Before you can write Servlets, you need to get a working install of the environment and familiarize yourself with the basics. Install the necessary software and adapting the code provided in the book, create, display and access a simple Servlet that does something simple like printing the current time according to the server.

Once you have this working, try to use the Session object. Create a simple page that links back to your Servlet, displaying the amount of time that has passed since the page was last accessed by the current user.

Modify your code to use Cookies instead of Session.

Finally, turn off cookies on your browser, and check that your application does not work any more (neither the cookies version, nor even the session version). Modify your code to use encodeURL and verify that the session based version of your application now works again.

2- Do exercise 6.13 and 6.14 of the book, page 360

3- Simple http proxy

In some circumstances, it is handy to access to a page indirectly. That is, instead of accessing the server hosting the page, you access another server, and ask it to fetch the page and give it to you. Such a server is called a proxy. Proxies are common, and in fact your browser can be configured to always use a proxy.

Here, we would like to create a small, simple proxy that does not require any browser configuration. Simply access the proxy, passing the requested URL as parameter, and get the corresponding page back as a result.

Question 1

For the initial version of our proxy, we will use Java's java.net.URL and java.net.URLConnection.

Create an html page with a form containing just one input. The user provides a URL in the input and submits the form to your Servlet. It reads the URL that has been input, and using the Java packages mentioned above, it fetches the URL and simply returns it to the user.

Question 2

Our first cut at the proxy doesn't always work very well. Some images might be missing, clicking on links sometimes doesn't work, etc. It is because any relative URL included in the page is now incorrect: it is understood as being relative to the Serlvet's location (the proxy) and not relative to the fetched URL location. Using java.util.regex.Pattern (a java implementation of regular expression), modify your proxy to take care of the issue.