Exercises: The page Directive

1. Use JSP to make an Excel spreadsheet where the first row says "Year," "Apples," and "Oranges." It should then have two rows of data (2008 and 2009), where each entry is a random number between 0 and 10. I.e. the result should look something like this:

Year	Apples	Oranges
2012	9.23456	3.98765
2013	4.45678	2.223344

If you choose to use the tab-separated data approach (which is probably best here), be careful because Eclipse might change your tabs to spaces automatically. If so, you can edit the file outside of Eclipse using a normal text editor, or you can do Window --> Preferences --> General --> Editors --> Text Editors --> Insert spaces for tabs.

2. Make an Excel spreadsheet with a random number of rows.

3. The java.math package has a class called BigInteger that lets you create whole numbers with an arbitrary number of digits. Create a JSP page that makes a large BigInteger from a String you supply as a request parameter, squares it, and prints out the result. Use the online API at http://docs.oracle.com/javase/6/docs/api/ to see the syntax for the BigInteger constructor and squaring operations.

4. Make a JSP page that sleeps for 20 seconds before returning the page. (Call TimeUnit.SECONDS.sleep(20) inside a try/catch block that catches InterruptedException). Access it "simultaneously" from Firefox and Internet Explorer. Repeat the experiment with the JSP page not allowing concurrent access. Verify the slower result.

5. **[Very hard; for the truly inspired.]** The above example worked because our version of Tomcat implements isThreadSafe="false" by queueing up the requests and passing them one at a time to the servlet instance. But, that is not the only legal implementation. Create a test case that will definitively show which of the following three approaches a server uses for isThreadSafe="false":

• Keeps a single servlet instance and queues up the requests to it

• Makes a pool of instances but lets each instance only handle one request at a time

• Ignores isThreadSafe altogether

Note that *all three* approaches have been represented by Tomcat versions in the past.

6. **[Just for fun.]** Grab the ComputeSpeed and SpeedError pages from the page-directive project. Access the ComputeSpeed page with numeric values for the "furlongs" and "fortnights" form parameters attached to the end of the URL. (See page 366 if you want more detail). Now, supply a non-numeric value for one of the parameters. Next, supply a legal number for furlongs, but 0 for fortnights. Can you explain the unexpected result you get?