HW: Forms

1. Make a registration form that collects the user's first name, last name, phone number and shoe size. Send the data to a servlet that will display the 4 values in an HTML table where the first column will be the property (name, shoe size, etc.) and the second column will be the property's value. The servlet should use a default value for any form value that the user fails to supply. Be sure to use a relative URL of some sort for the form's action so that you can test at home, deploy to JHU and test again, all without making any changes to the servlet or the form. **Use this approach (no changes to code after deploying to JHU) on all assignments for the semester.**

2. Make a variation of the previous servlet that uses default values for missing values, but have this one also filter the input for special HTML characters before displaying it. So, for shoe size, one should be able to put in " < 10". Create a form to test it (this can be identical to the form #1 except for the action).

3. Make a variation of the first servlet and accompanying form, but, in this case, if the user fails to supply a value for any of the fields in the form, you should redisplay the form but with an error message saying that the value is missing. Don't make the user reenter values that they've entered already. Note that this problem should be simple, but is instead a bit tedious. We'll see better solutions later in the semester. (You are not allowed to use more advanced approaches on this homework even if you know how.)

   Hint: you have two main choices given the tools available to us so far: have one big servlet that generates both the HTML form and the results page (the form submits the data to itself), or two separate servlets (one that generates the HTML form and the other that generates the results page). There is an example of the first approach in the book. If you want to try the second approach, you might want to know about response.sendRedirect, which lets you redirect the browser to a specified page. For example, here is a doGet method that sends the user to www.whitehouse.gov:

```
public void doGet(HttpServletRequest request,
                  HttpServletResponse response)
throws ServletException, IOException {
     response.sendRedirect("http://www.whitehouse.gov/");
}
```

The two approaches (one big servlet or two servlets—one for the form and one for the results — are about equal difficulty). Whichever you choose, do *not* use the more advanced approach from the book with beans that are filled in automatically. That approach is overkill for this simple problem.

Be sure to hand in source code for all the servlets and forms (for each problem Java first, HTML next). On the front page, prominently list the URLs of the three problems so that I can test them. On this and all future assignments, you can omit the source code for any classes that are taken directly from me without any modifications.