Exercises: The MVC Architecture

Number 1 and 2 are warmups: regular Java classes, not servlets. Problem #3 is the goal you are working toward.

1. Define a class called StatePair that stores strings representing a state name (e.g., "California") and a state abbreviation (e.g., "CA", but this bean should have no logic -- it simply stores two strings). Note that Eclipse has a very nice shortcut for creating constructors and getter/setter methods in beans. Start by writing a class like this:

```
package myPackage;

public class StatePair {
  private String stateName, stateAbbreviation;
}
```

Then, R-click in the class, go to the Source sub-menu, and choose "Generate Constructor Using Fields". Eclipse will then make the constructor for you (but you should remove the bogus "super();" on the first line). Then, R-click in the class again, go to the Source sub-menu, and choose "Generate Getters and Setters". You can then have Eclipse automatically create getStateName, setStateName, getStateAbbreviation, and setStateAbbreviation. Depending on your logic, you might not want the two setter methods, in which case you can tell Eclipse to make only the getter methods.

2. Make a class with a method that takes a US state name (e.g., "Maryland") and returns the corresponding post office abbreviation (e.g., "MD"). You might start with a simple mapping where there are only one or two state names are supported.

3. Make an application where the user can enter a US state name and get the post office abbreviation. Start with a form where the user can enter a state name. Send the data to a servlet that looks up the corresponding abbreviation. Don't forget to handle bad data. The servlet would then create a StatePair bean, and if the data is good, send the bean to a results page to display it. If the data is bad, send the user to an error page. (You might have two error pages: one for missing data and one for unknown-but-not-missing state names.) *This is the single most important exercise in the entire servlet and JSP course!*

4. Repeat problem 3, but this time, if the same user returns to the input form in the same browsing session, the previously entered state name should be filled into the textfield automatically.