**JSP Custom Tags Quiz**

Question1:  
Rewrite the JSP page in the first demo (forEach) to use scripting instead of the JSTL forEach.

Answer:  
<% for(int i=0;i$lt;temperatures.length ;i++) {%>  
  
<%= temperatures[i][0]%>   
<%= temperatures[i][1]%>   
<%= temperatures[i][2]%>   
  
<%}%>

Question 2:  
How do custom tags relate to JSTL?

Answer:  
JSTL is a standard library of JSP actions, but JSP allows developers to create their own actions. A custom functionality  
that can be packaged and reused by content developers.

Question 3:  
What is the role of the URI in the TLD and the taglib directive?

Answer:  
The URI attribute value resolves to a location the container understands.

Question 4:  
What is a tag handler class?

Answer:  
A tag handler class is a class that extends the tag support framework class.

Question 5:  
What is the role of attribute setters in a tag handler class?

Answer:  
Setters in a tag handler class is to store the value set in jsp page and conforms to the JavaBeans architecture conventions.  
The Container calls this method to set the value from the tag attribute. It uses JavaBean property naming conventions to  
figure out that a attribute should be sent to the setter method.

Question 6:  
What is the role of the doTag() method in a tag handler class?

Answer:  
The heart of a simple tag handler is a single method, doTag, which is invoked when the end element of the tag is encountered.  
The Container calls doTag() when the JSP invokes the tag using the name declared in the TLD.

Question 7:  
What does the operation getJspContext().getOut().write("Hi Bob") do when called in a doTag() method?

Answer:  
It will render "Hi Bob" text where the custom tag invoked.

Question 8:  
What does the operation getJspBody().invoke(null) do when called in a doTag() method?

Answer:  
Process the body of the tag and print it to the response. The null argument means the output goes to the response rather than some OTHER writer you pass in.