SEGMENT 2

"Fear of failure must never be a reason not to try something." – Frederick Smith

Overview:

In ***Segment 2 (your second coding project)*** you will now *convert most of the html files you created in Segment 1 to JSP pages*. This will enable the pages to interact with the database and extract information to be displayed to the user.

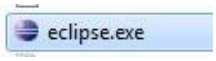
Before Starting:

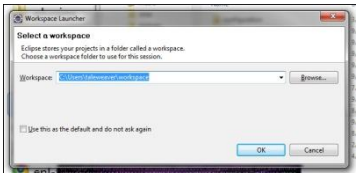
Before starting Segment 2 make sure you have the following installed and ready:

1. Latest Java SE Development Kit (J2SE SDK)
2. Latest Tomcat webserver
3. Eclipse Helios
4. ODBC Connection

Preparation:

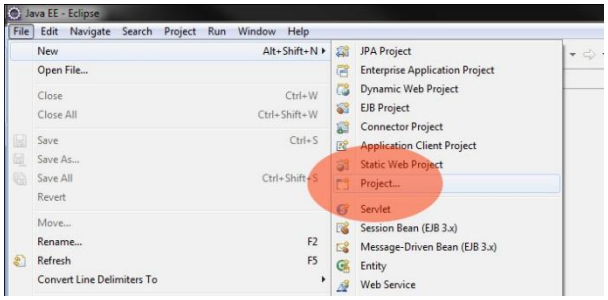
If you have not used Eclipse Helios to develop your HTML pages please do the steps below. If you have already used Helios it is also good to review the steps below to confirm that you're Eclipse Helios is ready for use in Segment 2.

1. **Open Eclipse:**
   Go to your Eclipse folder and double click on the Eclipse icon. 

2. **Select your Workspace:**
   The "Workspace" dialogue box will open. 

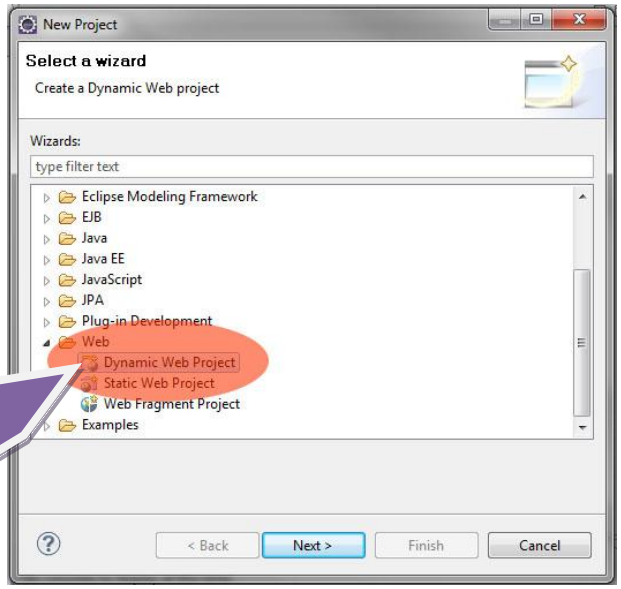
   Remember: Your workspace is where all your project files are kept.

3. **Create a new Dynamic Web Project:**
   The Eclipse workbench should now be open. Click on File → New → Project



4. **Select the project type:**
   Select “Dynamic Web Project” then click on next.

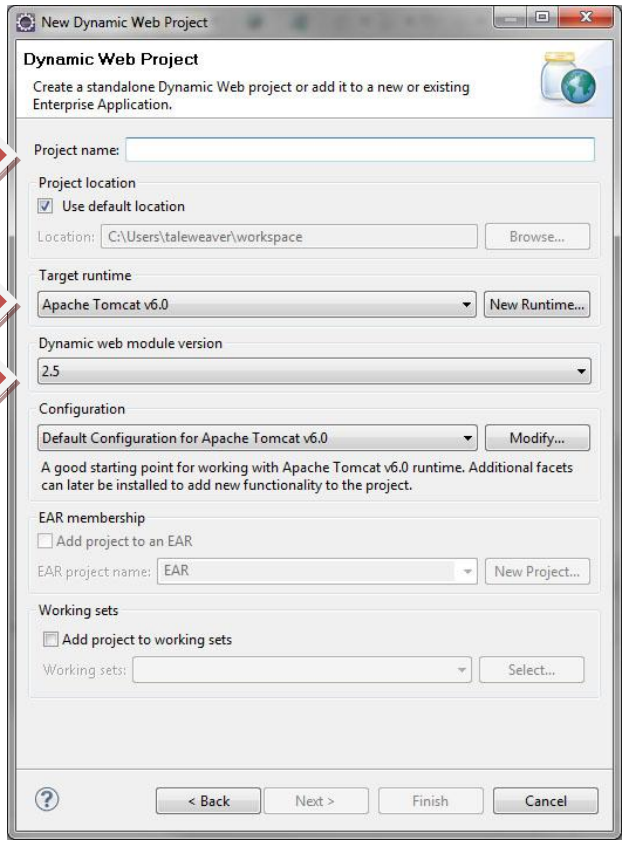
   REMEMBER: Since we are using JSP we need **a dynamic web project.**



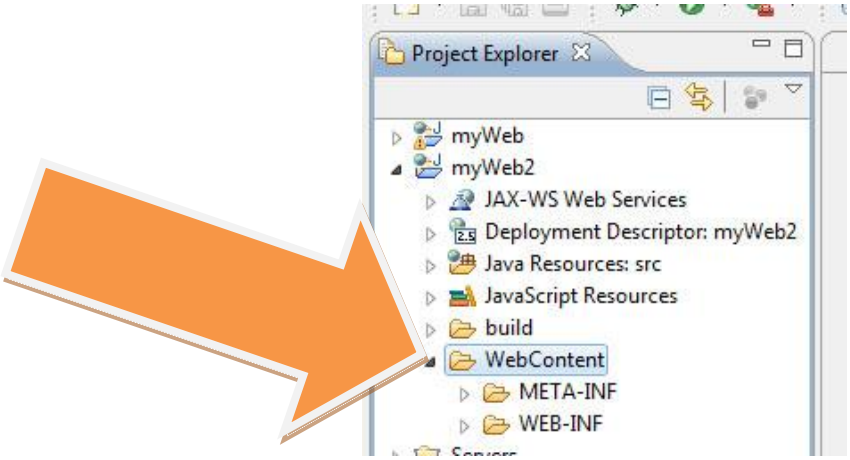
5. **Fill in the project options for:**
   - Project Name: **myWeb**
   - Target Runtime: Apache Tomcat 6.0
   - Dynamic web module version: 2.5

   Click on “Finish”

   You now have a new Dynamic Web project!



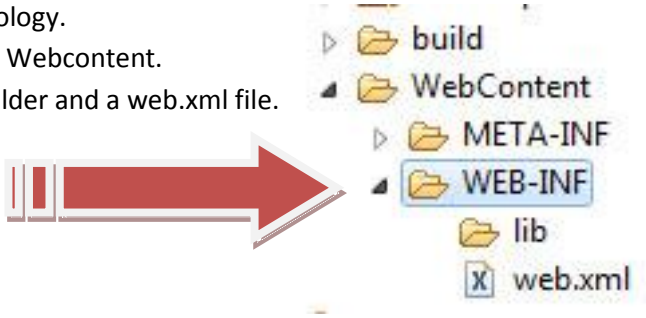
NOTE: Now you have a new dynamic web project. Always remember when you add a new HTML or JSP file to put them in the "webcontent" folder. *They will not work if placed in other folders.*



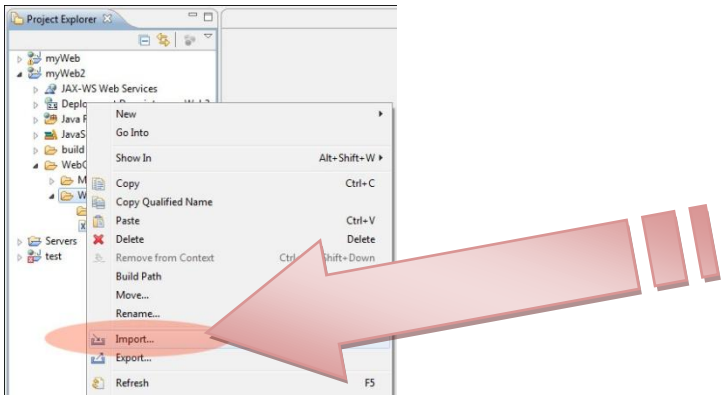
6. **Add pre-coded class files.**

   In a real-world programming project you are not required to code the entire project most of the time. The pre-coded class files contain database access and business logic needed by the application so that you will no longer need to do those parts. Your major role in this project is to code the display logic via JSP technology.

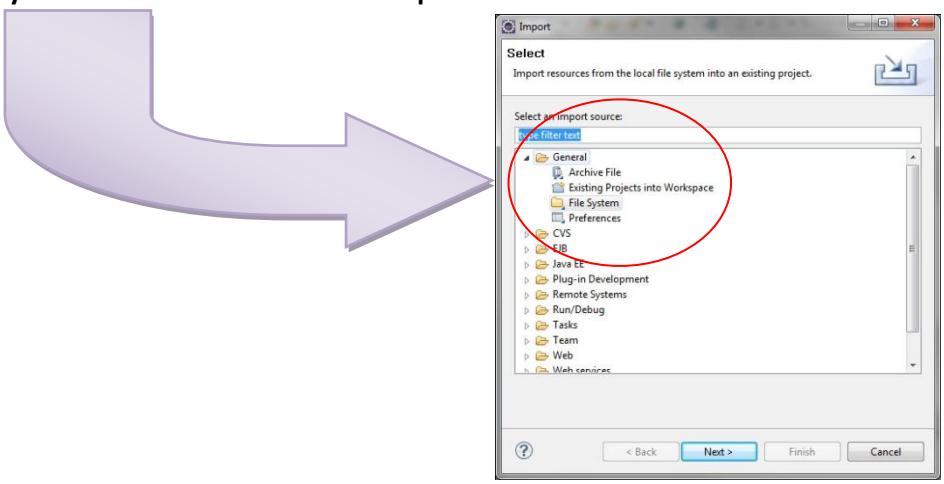
   a. **Expand the WEB-**INF folder in Webcontent.
   You will see it contains a lib folder and a web.xml file.



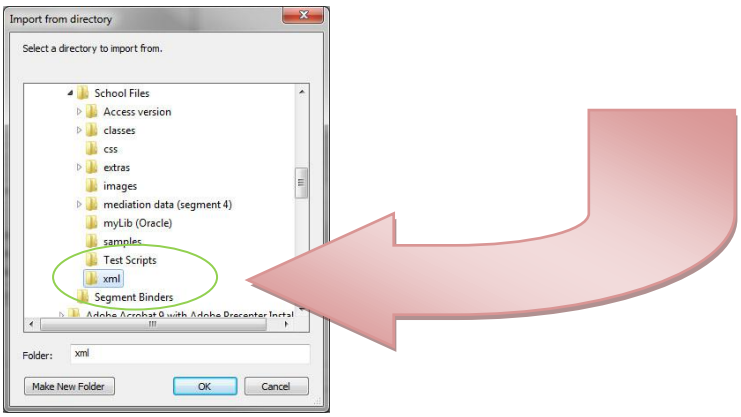
   b. **Right-click on WEB-INF and select IMPORT…**



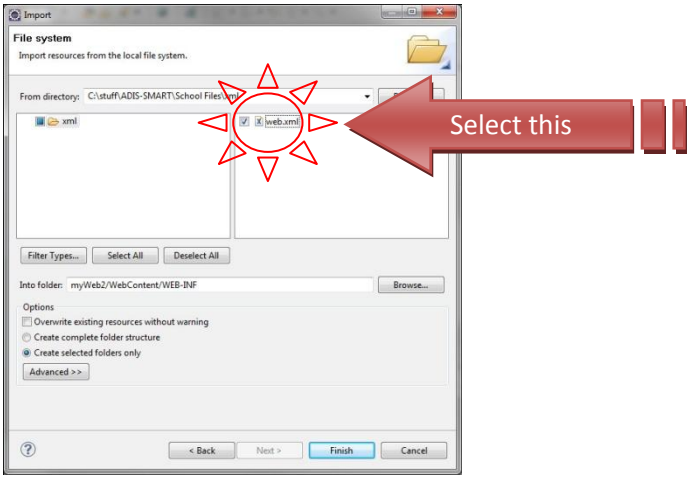
c. Select " File System" under the "General" Group



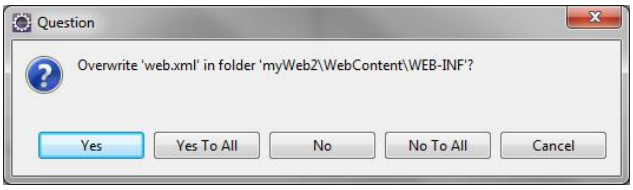
d. Browse for "xml" folder found in "School Files"



e. Select the "web.xml" file. Take note: select the exact file, not the folder.

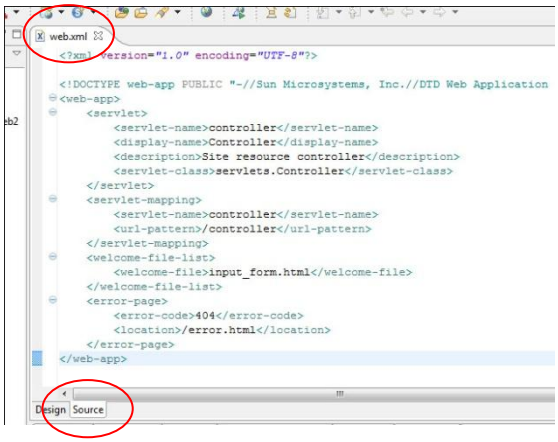


f. **Eclipse will ask you if you want to OVERWRITE the existing XML File. Click on "Yes".**

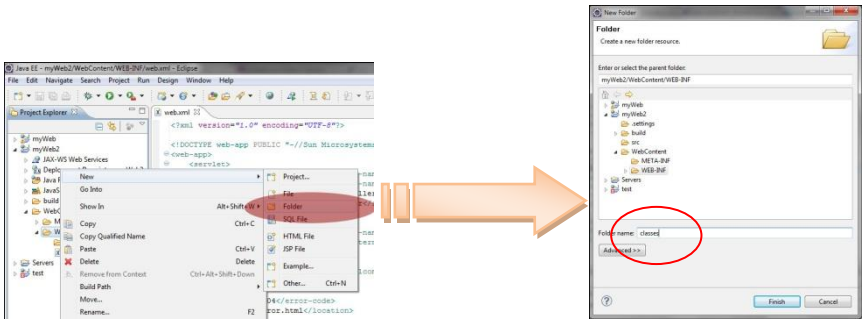


g. **Go to "Project Explorer." Double-click on "web.xml." This will open the file in the coding window. You should now have something similar to the one in the picture.**

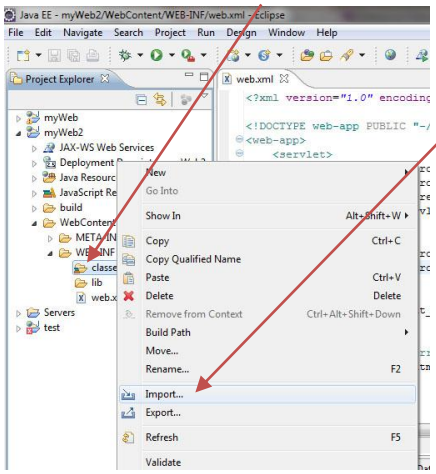
NOTE: The web.xml file is the main configuration file for a website running in Tomcat.



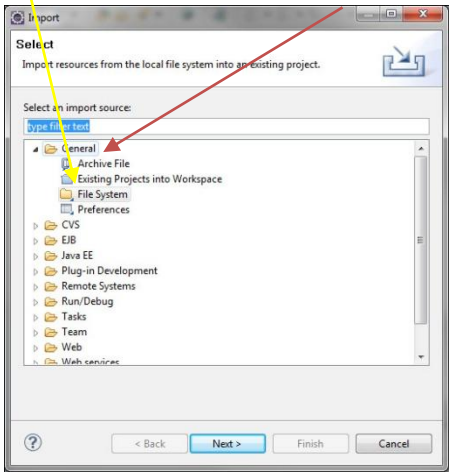
7. **Now that we have added the proper xml file we add the pre-coded classes. In "Project Explorer" right click on "WEB-INF", select "NEW" and then "FOLDER". Name the new folder "classes"**



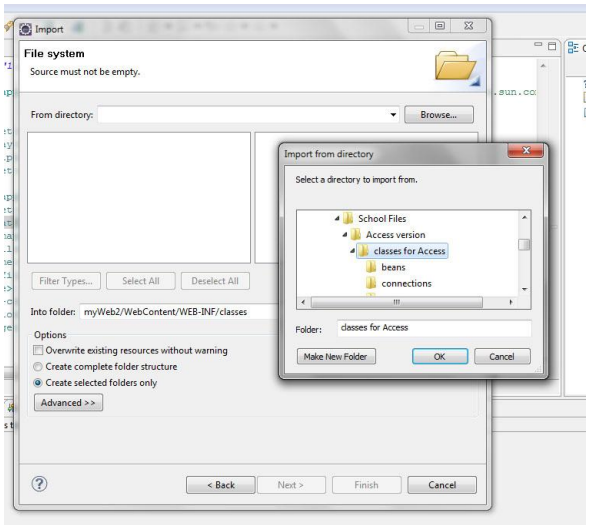
8. **Right click on the “classes” folder and select “import.”**



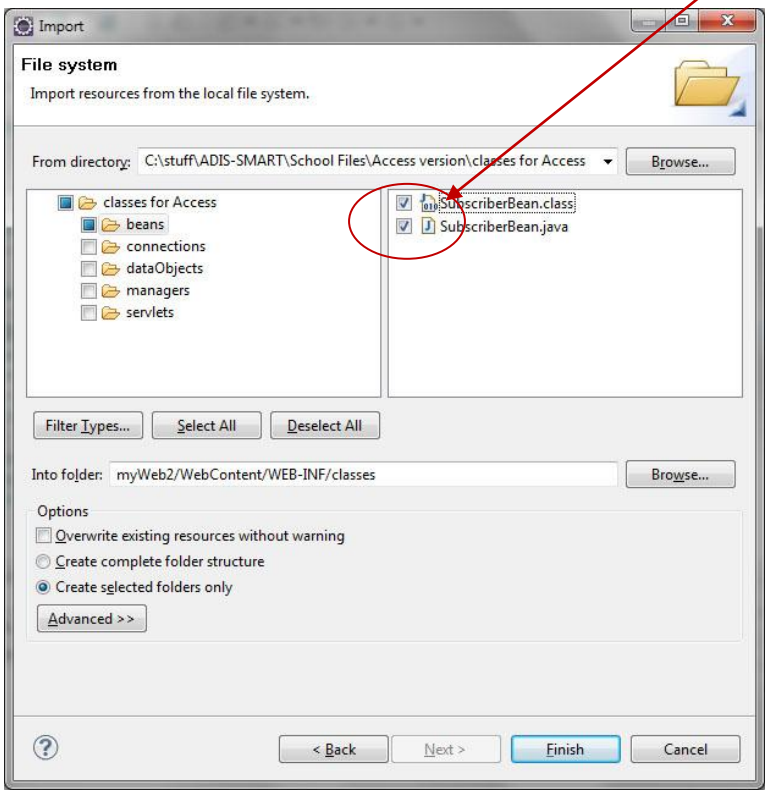
9. **Select “File System” under the “General” Group.**



10. **Browse for “classes for Access” folder found in “Access Version” under “School Files”**

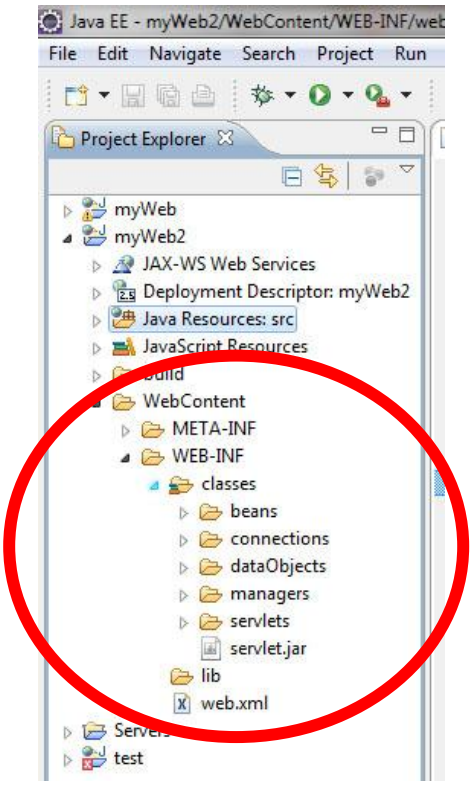


11. **Select all the ".jsp" files. Take note: select the exact file, not the folder.**
Do this for all the folders found under "classes for Access"



Click "Finish" once you have selected all files.

12. **Go to Project Explorer you should now see the directory structure pictured below.**



You are now ready to begin coding Segment 2.

CODING SEGMENT 2 (SOME HINTS TO START YOU OFF)

If you did SEGMENT 1 well then it's just a matter of copying your HTML content into their corresponding JSP files. After you have done this you need to add the JSP logic that will make the page run dynamic content.

NOTE: Read the ***technical specifications*** section of the Segment Binder to answer the question "What do I need to code?"

Input_form.html

The "input form" does not need to be converted to jsp since the form itself will do no processing. Make sure the target for the submit button correctly follows the specifications. The code will look something like the code below:

Sample code for **input_form.html** **(without layout/page decorations)**

```
<html>
    <head>
        <title>Account Number Input Form</title>
    </head>
    <body>
        <p align="center">
            <form method="post" action="controller">
                Subscriber Account Number:
                <input name="account_number" type="text" />
                <input name="page" value="subinfo"type="hidden" />

                <input name="submit_button" type="submit" value="Submit" />
            </form>
        </p>
    </body>
</html>
```

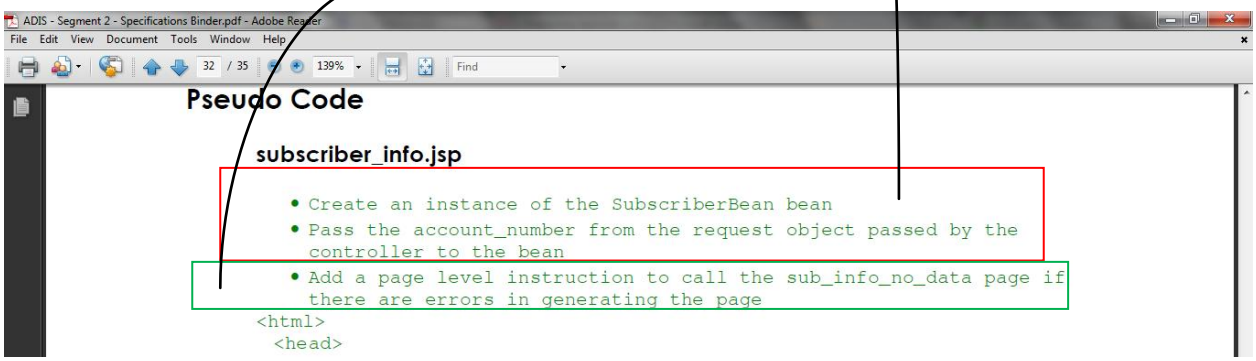
NOTE: Re-check your input_form.html so that it conforms to the form elements here. Take note of the hidden input control.

Subscriber_info.jsp

Make sure you read the pseudocode (Green text in technical specifications) for this very carefully. Sample code snippets below with reference to the pseudocode:

```
<jsp:useBean id="subscriberInfo" scope="request" class="beans.SubscriberBean">
    <jsp:setProperty name="subscriberInfo" property="account_number" />
</jsp:useBean>
<%@ page errorPage="/sub_info_no_data.html" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
```

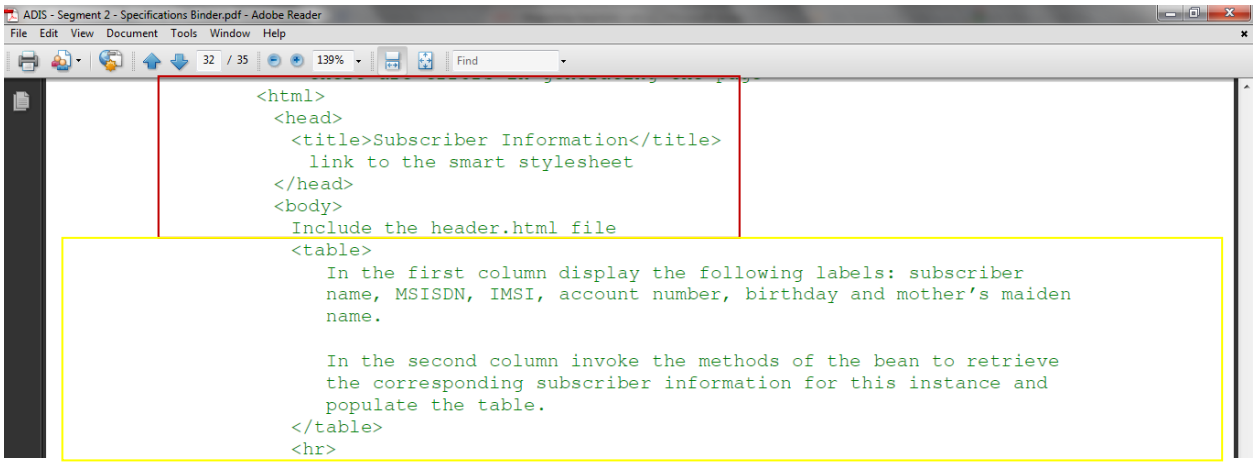
The code above corresponds to the pseudocode below:



Continuing the code:

```
<html>
  <head><title>Subscriber Info</title></head>
  <body>
    <p align="center">
    <table>
        <tr>
            <td>Subscriber Name:</td>
            <td><jsp:getProperty name="subscriberInfo" property="subscribername"/></td>
        </tr>
        <tr>
            <td>MSISDN:</td>
            <td><jsp:getProperty name="subscriberInfo" property="MSISDN"/></td>
        </tr>
        <tr>
            <td>IMSI:</td>
            <td><jsp:getProperty name="subscriberInfo" property="IMSI"/></td>
        </tr>
        <tr>
            <td>Account Number:</td>
            <td><jsp:getProperty name="subscriberInfo" property="account_number"/></td>
        </tr>
    </table>
```

Pseudocode:

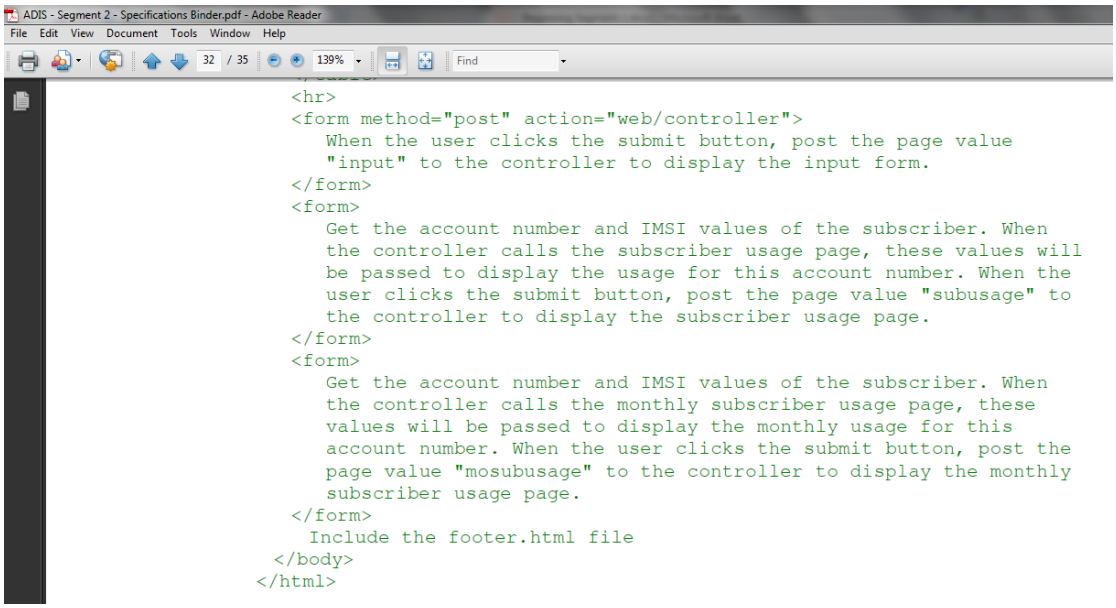


Final Part of the Code:

```
    <hr>
    <form method="post" action="controller">
        <input name="page" value="input" type="hidden"/>
        <input value="Input Form"
               type="submit"/>
    </form>
    <form method="post" action="controller">
        <input name="account_number"
               value="<jsp:getProperty name="subscriberInfo" property="account_number"/>"
               type="hidden"/>
        <input name="imsi"
               value="<jsp:getProperty name="subscriberInfo" property="IMSI"/>"
               type="hidden"/>
        <input name="page" value="subusage" type="hidden"/>
        <input value="Get Subscriber Usage"
               type="submit"/>
    </form>
    <form method="post" action="controller">
        <input name="account_number"
               value="<jsp:getProperty name="subscriberInfo" property="account_number"/>"
               type="hidden"/>
        <input name="imsi"
               value="<jsp:getProperty name="subscriberInfo" property="IMSI"/>"
               type="hidden"/>
        <input name="page" value="mosubusage" type="hidden"/>
        <input value="Get Subscriber Monthly Usage"
               type="submit"/>
    </form>

    <form method="post" action="/web/controller">
        <input name="account_number"
               value="<jsp:getProperty name="subscriberInfo" property="account_number"/>"
               type="hidden"/>
        <input name="imsi"
               value="<jsp:getProperty name="subscriberInfo" property="IMSI"/>"
               type="hidden"/>
        <input name="page" value="ir" type="hidden"/>
        <input value="Enable International Roaming"
               type="submit"/>
    </form>
    </p>
  </body>
</html>
```

Final Pseudocode for subscriber_info.jsp:



NOTE: You should now be able to finish Segment 2.

NOTE: The code above only completes the logic portion of the pseudocode. The layout parts are not shown.

NOTE: Complete Code (logic portion) for subscriber_info.jsp:

```
<jsp:useBean id="subscriberInfo" scope="request" class="beans.SubscriberBean">
    <jsp:setProperty name="subscriberInfo" property="account_number" />
</jsp:useBean>
<%@ page errorPage="/sub_info_no_data.html" %>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
  <head><title>Subscriber Info</title></head>
  <body>
    <p align="center">
    <table>
        <tr>
            <td>Subscriber Name:</td>
            <td><jsp:getProperty name="subscriberInfo" property="subscribername"/></td>
        </tr>
        <tr>
            <td>MSISDN:</td>
            <td><jsp:getProperty name="subscriberInfo" property="MSISDN"/></td>
        </tr>
        <tr>
            <td>IMSI:</td>
            <td><jsp:getProperty name="subscriberInfo" property="IMSI"/></td>
        </tr>
        <tr>
            <td>Account Number:</td>
            <td><jsp:getProperty name="subscriberInfo" property="account_number"/></td>
        </tr>
    </table>
    <hr>
    <form method="post" action="controller">
        <input name="page" value="input" type="hidden"/>
        <input value="Input Form"
               type="submit"/>
    </form>
    <form method="post" action="controller">
        <input name="account_number"
               value="<jsp:getProperty name="subscriberInfo" property="account_number"/>"
               type="hidden"/>
        <input name="imsi"
               value="<jsp:getProperty name="subscriberInfo" property="IMSI"/>"
               type="hidden"/>
        <input name="page" value="subusage" type="hidden"/>
        <input value="Get Subscriber Usage"
               type="submit"/>
    </form>
    <form method="post" action="controller">
        <input name="account_number"
               value="<jsp:getProperty name="subscriberInfo" property="account_number"/>"
               type="hidden"/>
        <input name="imsi"
               value="<jsp:getProperty name="subscriberInfo" property="IMSI"/>"
               type="hidden"/>
        <input name="page" value="mosubusage" type="hidden"/>
        <input value="Get Subscriber Monthly Usage"
               type="submit"/>
    </form>
    <form method="post" action="controller">
        <input name="account_number"
               value="<jsp:getProperty name="subscriberInfo" property="account_number"/>"
               type="hidden"/>
        <input name="imsi"
               value="<jsp:getProperty name="subscriberInfo" property="IMSI"/>"
               type="hidden"/>
        <input name="page" value="ir" type="hidden"/>
        <input value="Enable International Roaming"
               type="submit"/>    </form>
    </p>
  </body>
</html>
```

RUNNING THE SITE

Once you have coded *input_form.html* and *subscriber_info.jsp* you can test your work by clicking on the project folder and then clicking on the go button (green arrow) on the toolbar.

The *web.xml* file tells Tomcat to display *input_form.html*.

Enter the value 119777259. This will display the subscriber info for Verlin Samonte. If you get his information then you have coded the two files correctly.

Other test data are found at the end of the Segment 2 Test Script.

TESTING SEGMENT 2

Segment 2 is the first segment that comes with a formal testing document. To complete the segment and to check if you have coded things correctly follow the Segment 2 test script and fill it out completely.

You should fill out the columns for actual result, test result and action taken. Actual results is what you observe (usually a restatement of expected result if it is the same) , test result is passed if it actual results match expected results. Action taken is filled out if Test Result is failed. Usually Action taken contains the entry "Recoded module" or "Fixed coding error."

"What is genius, anyway, if it isn't the ability to give an adequate response to a great challenge?" -- Bette Greene

REMINDERS:

If you think some of the steps are missing please refer to segment 1 and segment 2 binders.

Things to submit for Segment 2:

1 html file: corrected input_form.html

3 JSP files:

- **subscriber_info.jsp**
- **subscriber_usage.jsp**
- **subscriber_monthly_usage.jsp**

1 filled-out Test Script

Deadlines:

End of class (March 19):

- **input_form.html and subscriber_info.jsp**
- **email to: rlramos@gmail.com**
  subject: Appdev Segment 2 A 3rd term 2010-11 <your name>

End of day (March 27, 11:59pm):

- **subscriber_usage.jsp and subscriber_monthly_usage.jsp**
- **filled out test script**
- **email to: rlramos@gmail.com**
  subject: Appdev Segment 2 B 3rd term 2010-11 <your name>
