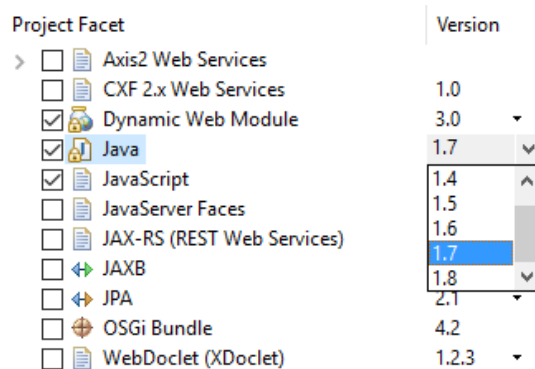# CSCI 441 Eclipse Setup
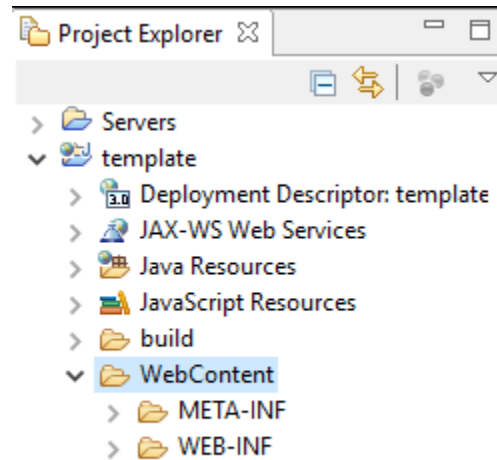
1. Create a folder in your home directory named EclipseWorkspace
   a. Create a folder in the EclipseWorkspace named **server**
2. Launch Eclipse
3. Select the Workspace that you created in step 1
4. When Eclipse loads select Window > Preferences > Server > Runtime Environments.
   a. If a server is not listed > Add > Apache Tomcat v7.0
   b. **Check the Create a new local server** > Click next
   c. Click download install > accept license agreement > navigate to server directory created in step 1a > Click Ok (Wait a bit as it downloads and installs with no visual indication of this process)
   d. Click finish
5. Close the preferences window
6. Start a new web project
   a. File > new > Dynamic Web Project
   b. Project Name > c03a_template
   c. Under Configuration > Click Modify
   d. Change the **Java Version to 1.7**

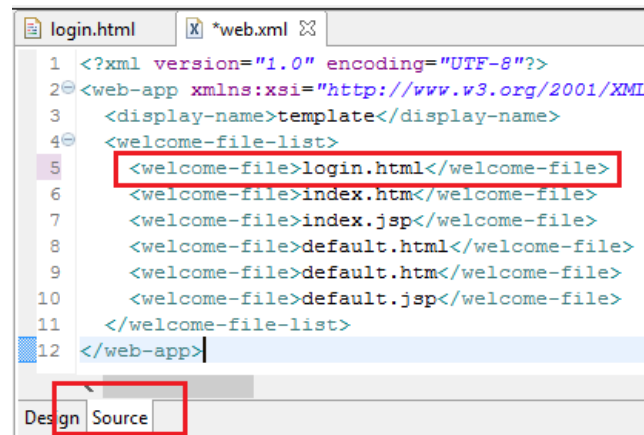| Project Facet	Version	
> ☐ 📄 Axis2 Web Services		
☐ 📄 CXF 2.x Web Services	1.0	
☑ 📄 Dynamic Web Module	3.0	▾
☑ 📄 Java	1.7	⌄
☑ 📄 JavaScript	1.4	⌃
☐ 📄 JavaServer Faces	1.5	
☐ 📄 JAX-RS (REST Web Services)	1.6	
☐ ↔ JAXB	1.7	
☐ ↔ JPA	1.8 ⌄	
	2.1	▾
☐ ⊕ OSGi Bundle	4.2	
☐ 📄 WebDoclet (XDoclet)	1.2.3	▾

   e. Back in the project setup click next > next to get to the Web Module window. Check the Generate web.xml deployment descriptor > Click Finish

7. Develop web project
    a. Right click WebContent > New > HTML file create login.html

    b. Expand WEB-INF > web.xml > change the index.html to login.html (this will be your default page)

    c. C03a_template > Java Resources > src – Right click src > New > Package - name it constants > F5 to refresh
    d. Right click constants package > New > Class – Name it UniversalConstants > Finish
    e. C03a_template > Java Resources > src – Right click src > New > Package - name it servlets > F5 to refresh
    f. Right click servlets package > New > Servlet > Class name – WelcomeServlet > next > next > only check Inherited…, doPost, and doGet > Click Finish

Which method stubs would you like to create?

☐ Constructors from superclass
☑ Inherited abstract methods

☐ init   ☐ destroy   ☐ getServletConfig
☐ getServletInfo ☐ service   ☑ doGet
☑ doPost   ☐ doPut   ☐ doDelete
☐ doHead   ☐ doOptions   ☐ doTrace

Time to code

1. In your WebContent folder open your login.html – add the following then save

```
<form action="WelcomeServlet">
     <input type="submit" value="Login">
</form>
```

2. Open the servlets.WelcomeServlet
   a. In the doGet method remove:

      // **TODO** Auto-generated method stub
      response.getWriter().append("Served at: ").append(request.getContextPath());
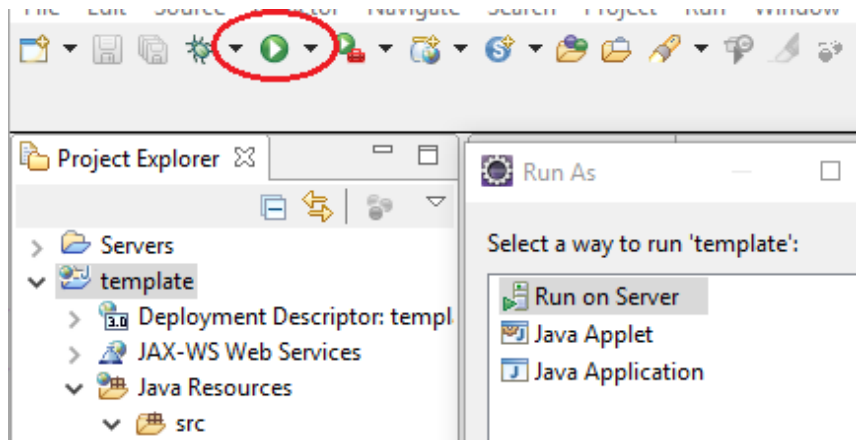
   b. Add the following to the doGet

      ```
      response.setContentType("text/html");
      PrintWriter out = response.getWriter();
      try {
         out.println("Welcome Page");
      } finally {// always executes when try block exists
         out.close();
      }
      ```

   c. PrintWriter needs to be imported. Either Ctrl + Shft + O or hover PrintWriter and choose Import 'PrintWriter' (java io)
   d. Save document

Time to run

1. Highlight the project > Click the Run As Button (green circle with arrow) > Select Run on Server > Click OK > Click Finish

2. Click the submit button to go to the WelcomeServlet

Time to code

1. Open the constants.UniversalConstants.java and add the following

```java
public static String htmlHead (String title) {
    return("<!DOCTYPE html>"
            + "<html>"
            + "<head>"
                    + "<title>" + title + "</title>"
            + "</head>"
            + "<body>");
    }
public static String htmlClose ="</body></html>";
```
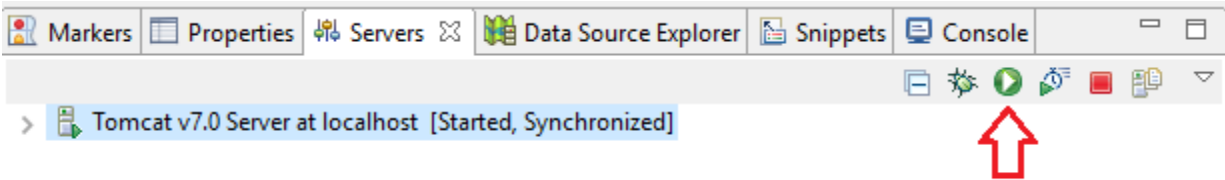
2. Back in the WelcomeServlet add the following above the PrintWriter code

```java
String htmlHead = constants.UniversalConstants.htmlHead("Welcome");
String htmlContent = "";
String htmlClose = constants.UniversalConstants.htmlClose;

htmlContent = "<h1>Welcome</h1>"
                + "<p>hello</p>";
```
3. Change the try to out.println(htmlHead + htmlContent + htmlClose);
4. Restart your server (this is not always needed but sometimes the server is slow when building a new servlet or class).
   a. In the server tab on the bottom > click the run button (green with an arrow)

b. Open the page in chrome (http://localhost:8080/yourProjectName/WelcomeServlet) >
F12 > Elements – take a look at the html
   i. Right click page > view source. Notice a difference?

Create a logout handler

1. In the WelcomeServlet and PageTwo Servlet add the following to the htmlContent String

   + "<form action=\"LogOutHandlerServlet\">"

   + "<input type=\"submit\" value=\"Log Out\">"

   + "</form>";
2. Under src > servlets > Create a LogOutHandlerServlet (see previous instruction for servelts)
3. Add response.sendRedirect("login.html"); to the doGet
4. Save everything > restart server > view in chrome

Challenge

Create a second servlet named PageTwoServlet. Use the constants.UniversalConstants to
create the head and close. Make up your own content.

In constants.UniversalConstants, create a simple nav bar that has the WelcomeServlet and
PageTwoServlet as relative links. Implement this constant in both of your servlets.
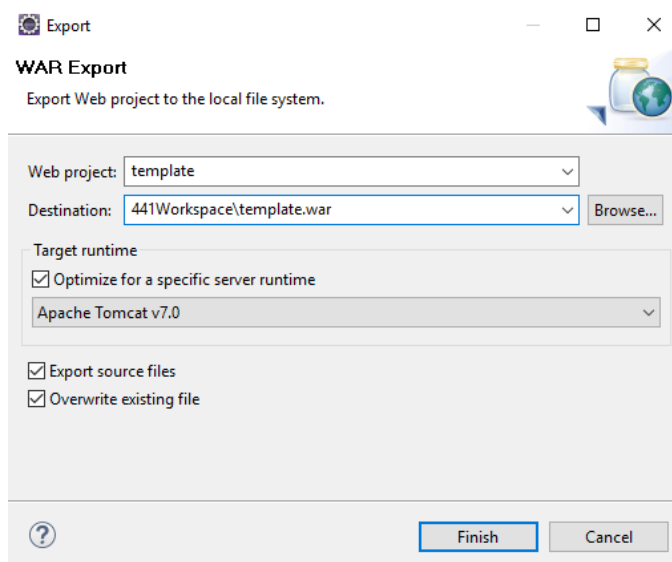
Prepare for and Deploy on Google Server

1. Right Click project name (e.g., template) > Export > WAR file
2. Select Destination, **Apache Tomcat v7.0**, and **Export source files (***important***)**

3. Navigate to your manager on google - http://ipAddress:8080/manager/html

4. Scroll to WAR file to deploy > choose WAR file > Click Deploy

5. Scroll up to the Applications section and click on the like to your page in the Path column (/template in this instance)

6. Success!!!!