

Set up your Servlet/JSP project on Eclipse as follows

- Create a new Dynamic Web Application project in Eclipse as follows
  - Choose File > New > Dynamic Web Project
  - As part of the options when you do the above, you will see an option for Target Runtime. If you see an option for Tomcat 7, use it. If not click on the New Runtime button and select Tomcat 7.
- Right click on the project, select Build Path > Configure Build Path
- Go to the Libraries tab if you are not already on it.
- Select Add External Jars, and add the jar file servlet-api.jar, which you can find in /usr/share/tomcat7/lib.
- Repeat the above to add jsp-api.jar, and for el-api.jar, which can also be found in the same directory.
- Now create a servlet by going to the project, right click and choose New > Servlet. Give an appropriate name to the servlet.  
The servlet gets created in Java Resources > src > (default package).
- Copy the following text into the doGet method.

```
PrintWriter out = response.getWriter();
out.println("<html><body>Hello world</body></html>");
```
- Run the servlet by right clicking and choosing Run on Server. Choose Tomcat 7 as the server (this should be the default).  
You should see a new browser window open where the Hello world text is shown
- Now extend your servlet to get parameters from the request and output them as follows
  - Create a html file with a form, as below, to create values for the parameters; note that the action is Test, which should be replaced by the name you used for your servlet.

```
<form action="Test" method="get">
    Enter your name: <input type="text" name = "name">
    <input type="submit" value = "Submit">
</form>
```
  - Add the following lines to your servlet to get the request parameters:

```
String name = request.getParameter("name");
```

and add more lines to print the name to the output to check that you are getting the parameters properly.
  - Right click on the html file and select Run on Server. Fill in the form and see if the Test servlet gets executed.
- To set up tomcat to connect to PostgreSQL, you must add your [postgresql JDBC jar file](#) to the classpath for tomcat; adding it as you did last time to the Java path is not sufficient. To do this, go to Run Configurations, and then select tomcat, and the select the classpath tab of tomcat, and add the jar file there as an external jar file.

```
...
C:\Program Files\Apache Software Foundation\Tomcat 7.0\lib\postgres-jdbc-9.2.1000.jar
...
```
- Now you are all set to actually work on your assignment. For the in-class assignment you have to do the following:
  - Create a password table in your database, which has two attributes, ID, and password, both varchar(20).
  - Fill sample user ID/password in the table.
  - Create a servlet that allows a user to login with a login/password. If the parameters ID and password are not set it displays a form to fill in the parameters. If they are set it checks if the ID/password match. On failed authentication, it displays an error message and the form. On successful authentication of the password, it sets a session variable to store the ID, and shows two links, one to display the courses taken by that ID, and the second to show all courses being taught by that ID.
  - Each of the above options is implemented by a servlet. Each servlet should check for authentication via the session variable. The servlets get the ID from the session, and display all matching records as a HTML table

Submit a zip or tar file containing all the files that you have created. Make sure to name the directory by your roll number, so expanding the files will create separate folders for different students.