Homework 0: Getting Started!

This homework will serve as a warm-up for web app development. In it, you will practice several skills:

- Setting up Eclipse for Java EE development
- Accessing a Subversion repository
- Running a Java EE web app

The homework will introduce you to several tools that you will be using throughout the semester: Eclipse, Tomcat, and Subversion (SVN). These are professional-quality tools commonly used in practice.

Step 0. Send me your username + encrypted password

Although not needed for this homework, you will need a username and password to commit to the Subversion repository in future assignments.

To set up a username/password, do the following:

1. Go to http://www.cs.memphis.edu/~sdf/util/htpasswd/, and fill out the form. Use your UofM username (i.e., the name before the @ in your university email address). Once you've filled in your username and password, press the button.
2. Copy all of the **encrypted password** string, and paste it into an email to me.

Note that this username/password is independent from all your other university accounts. Don't forget this password! If you need to change it, simply repeat the above steps.

Step ½. Get Java (if you don't already have it)

You must have Java JRE version 6 (aka 1.6) or later installed on your system. If you don't already have it, get it here:

http://www.java.com/

Simply run the executable to install.

Step 1. Get Eclipse IDE for Java EE Developers

First, create a folder somewhere in your user directory (e.g., on your Desktop) called **jee-dev**. (Actually, you can name it whatever you want.)

IMPORTANT! It's important that you pick a good spot for this folder. Moving it later will break things.

Download the latest version (*Juno*) of *Eclipse for Java EE Developers* and save it to your **jee-dev** folder. Eclipse is the most widely used Integrated Development Environment (IDE) for Java. It includes facilities for editing, running, and debugging Java EE web apps. Get it here:

http://www.eclipse.org/downloads/

IMPORTANT! You need the version called Eclipse IDE for Java EE Developers.

Once you've downloaded Eclipse, unpack the zip/tar.gz into your **jee-dev** folder. Thus, you should have an **eclipse** folder inside your **jee-dev** folder. The **eclipse** folder contains the Eclipse executable among other things.

Next, create a folder **workspace** inside your **jee-dev** folder. Eclipse will use this folder to store your project files.

Now, it's time to fire Eclipse up. To do so, run the executable **eclipse** within the **eclipse** folder (i.e., with the same name). When Eclipse prompts you to select a workspace, choose the **workspace** folder you created earlier.

Once Eclipse has launched, click the **Workbench** button (that looks like this:) to open the **Java EE** perspective.

Now set Eclipse aside for a bit, while we move on to the next step…

Step 2. Get Tomcat 7

Download and unpack Apache Tomcat version 7 on your machine. Tomcat is a widely used Java EE Web Container. You will use Tomcat as a server for your Java EE web apps. Get it here:

http://tomcat.apache.org/download-70.cgi

Choose one of the Core Binary Distributions.

IMPORTANT! Windows users, don't choose the service-installer version. This version will interfere with your ability to start/stop Tomcat from within Eclipse.

Download tomcat and unpack the zip/tar.gz into your **jee-dev** folder. Thus, your jee-dev folder should now contain three folders: **eclipse**, **workspace**, and **apache-tomcat-7.0.*XX***.

You should take a look at the **RUNNING.txt** file within the **apache-tomcat-…** folder. Depending on your platform, you may need to set certain environment variables.

Step 3. Create a Tomcat server in Eclipse

To run a web app in Eclipse, you must configure Eclipse so that it knows where a web server is located on your system.

Starting from Java EE perspective, do the following to add a Tomcat server to Eclipse.

1. Click **File → New → Other…** This should open a wizard-selection window.
2. Expand **Server** folder, select the **Server** wizard, and click **Next**. This should start the **Server** wizard.
3. Expand the **Apache** folder, select **Tomcat v7.0 Server**, and click **Next**. This should take you to a dialog box that asks for configuration information about the server.
4. To fill in the **Tomcat installation directory** field, click **Browse**. Find the **apache-tomcat-…** folder, select it, and click **Open**. Then click **Next** to advance the wizard. This should bring up the last dialog box of the wizard.
5. Click **Finish**. In the **Project Explorer**, you should see a **Servers** folder. Expand the folder, and you should see your Tomcat server.

Step 4. Install the Eclipse Subversion plugin

To download the example web app, you will need a Subversion plugin for Eclipse. Subversion is a tool that enables you to store your project files in a central repository and to keep track of the different revisions of your files. Later, you will use Subversion to turn in homework/project assignments and to collaborate on code with your team.

Follow these steps to install Eclipse's Subversive SVN Team Provider plugin:

1. Click **Help → Install New Software…** This should pop up an **Install** window.
2. In the **Install** window's **Work with:** dropdown, select **--All Available Sites--**. In the filter field (which initially contains the words "type filter text"), enter **Subversive**. This should cause the **Name**/**Version** columns to update. Note: Eclipse may be *very slow* in updating the table. Be patient and wait for it!
3. Check the box next to **Subversive SVN Team Provider**, and click the **Next** button. Click through the installation wizard by clicking **Next**/**Finish**/etc., and restart Eclipse when prompted. When Eclipse starts, it should pop up an **Install Connectors** window.
4. From the **Install Connectors** window, select the newest version of **SVN Kit** (*do not use* **JavaHL**), and click **Finish**. This should cause an installation wizard to pop up.
5. Click through the installation wizard by clicking **Next**/**Finish**/etc., and restart Eclipse when prompted. When Eclipse restarts, you should be back at the **Welcome** screen.

Note: If you mess up the **Install Connectors** bit and can't get that window to pop up again, try removing your workspace folder (I assume you have no valuable work in there at this point), and then restarting.

So what did this installation procedure do? It added a **SVN Repository Exploring** perspective to Eclipse. To open the perspective starting from the Welcome screen, do the following:

1. Click the **Workbench** button ( ). This should open the **Java EE** perspective.
2. Click **Window → Open Perspective → Other…** This should pop up a list of perspectives.
3. Select **SVN Repository Exploring** from the list to open the perspective. The left column of the main Eclipse window should have a **SVN Repositories** view instead of the usual **Project Explorer** view.

Step 5. Checkout the example web app

To download the example web app, you'll need to use Subversion to "checkout" a copy of the example source code.

Go to the **SVN Repository Exploring** perspective (see Step 3), and perform the following steps to do the checkout:

1. At the top of the **SVN Repositories** view is a **New Repository Location** button—click it. This should open a **New Repository Location** window with fields for you to fill in.
2. Fill in the fields as follows, and click **Finish**.
   - **URL:** https://utopia.cs.memphis.edu/course/comp7012-2013spring/examples/
   - **User:** anonymous
   - Leave the **Password:** field blank.

   This should add the repository to the **SVN Repositories** view. You may get a message about "master password" and/or "password recovery". This is your OS trying to help you keep track of your passwords. Whatever option you choose is probably fine.
3. Expand the contents of the repository by clicking the triangle. You should see a **KnockKnockJoke** folder.
4. Expand the **KnockKnockJoke** folder, and you should see three folders: **trunk**, **branches**, and **tags**.
5. Right click on **trunk**, and select **Find/Check Out As…** This should pop up a **Check Out As** dialog.
6. Select option **Check out as a project configured using the New Project Wizard**, and click **Finish**. This should bring up a wizard-selection dialog.
7. Expand the **Web** folder, select **Dynamic Web Project**, and click **Next**. This should start the **New Dynamic Web Project** wizard.
8. In the **Project Name:** field, enter **KnockKnockJoke**. Make sure that the **Target runtime** and **Configuration** fields say **Apache Tomcat v7.0**. Click **Next**. This brings up a configuration dialog for source and output folders.
9. There's nothing to do here, click **Next**. This brings up web-module configuration dialog.
10. Uncheck the **Generate web.xml deployment descriptor** checkbox, and click **Finish**. This brings up a prompt to switch to the **Java EE** perspective. Click Yes. If you see the **KnockKnockJoke** project in the **Project Explorer**, the checkout was successful.

Step 6. Run the web app

To run the web app, perform the following steps from the **Java EE** perspective.

1. Right-click on the **KnockKnockJoke** project in the **Project Explorer**, and then click **Run As → Run on Server**. This should bring up the **Run on Server** wizard.
2. Click **Finish** to launch the web app.

Submitting

To receive credit for this homework, you must

- demonstrate that you can run the web app at the start of class on the day the homework is due, and
- email me your encrypted password before class on the day the homework is due.