Downloading and Installing J2SE, and Creating Projects with Eclipse and JCreator

Installing Sun's Java Development Kit

For this course we will use the Standard Edition of the Java Development Kit (J2SE) by Sun Microsystems. I currently have two versions of J2SE installed on my computer, J2SE 8 (1.8.0_192) and J2SE 11. J2SE versions 9 and 10 were only briefly available, and are no longer supported by Oracle. Since J2SE 8, Oracle has embarked on a policy of providing periodic versions that have no long-term support (LTS) interspersed with versions that do have long-term support. Both J2SE 8 & 11 are LTS versions, while J2SE 9 & 10 were non-LTS versions. Therefore, I strongly recommend that you use either J2SE 8 or J2SE 11 in this course, and do not use J2SE 9 or 10.

You can download the latest version of J2SE, as well as installation instructions, at http://www.oracle.com/technetwork/java/javase/downloads/index.html. **Please note that you will need to install two pieces of software:**

1. the Java SE Development Kit (JDK),
2. the J2SE Runtime Environment (JRE).

Note: For J2SE 11 there is no longer a separate JRE to be installed; the only JRE that is installed is the one that comes bundled with the JDK.

Getting Started with the Eclipse IDE

The Eclipse IDE is a very versatile IDE, and contains considerably more features than you are likely to need for this course, but I encourage you to give it a try. Some of the key features I find most useful in Eclipse are:

1. auto-completion of code as you type;
2. auto-detection of syntax errors as you type;
3. auto-compile every time you save a source file;
4. the ability to refactor code: e.g., you can automatically rename a variable, method, or class throughout a project without having to manually locate every instance of the name;
5. auto-generation of getter and setter methods for attributes of a class;
6. the ability to automatically determine which import statements are required, and organize them in a source file; by default, Eclipse filters out "blanket" import statements such as, "`import java.util.*;`", instead importing only the specific classes referenced in the source file.

Below are some bare-bones instructions for creating a new Java project using Eclipse. These instructions assume you already have Eclipse and J2SE installed.

1. Click on the **File** menu and go to **New --> Project**.
2. In the dialog box that appears, choose "Java Project". Click **Next**.
3. A new dialog box will appear asking you for more information about the project.
4. Enter a name for the project in the blank provided.
5. If you already have source code for the project you can choose "Create project from existing source." Otherwise, choose "Create new project in workspace."
6. Make sure you have the appropriate JRE version set for the project. Eclipse allows you to specify different JRE versions for different projects. Click **Next**.

7. For project layout, select "Create separate folders for sources and class files". This allows your `.class` files to be separated from your `.java` files instead of having everything dumped into the same folder.

8. In the final dialog box that appears there are many configuration options that can be set. You can add other projects and libraries to the build path here, and specify the order in which the namespaces should be searched during compilation.

9. Click **Finish**.

To run the project, do one of the following:

1. With the source file containing the main method active in the editor window, choose **Run --> Run As** from the Run menu, and in the flyout menu that appears, choose **Java Application**. (If you try to do this with a file active in the editor window that does not contain a main method, no options will appear when you try to run the file).

2. Choose **Run --> Run...** to set up a runtime configuration. Once this has been set up you can run the project by choosing this runtime configuration and clicking the **Run** button.

Using the API Documentation for J2SE

The API documentation provided for J2SE is an indispensible resource, and it is essential that you become comfortable using it. It will give you information on how to use all of the standard library components that come packaged with J2SE. You can either download the docs to your hard drive if you wish, or alternatively, you can access them online at Sun's website using one of the links below:

API Documentation for J2SE 8 (http://docs.oracle.com/javase/8/docs/api/)

API Documentation for J2SE 11 (https://docs.oracle.com/en/java/javase/11/docs/api/index.html)

The API docs are in HTML format. For the API docs for J2SE 8 the main page contains 3 frames:

1. Package frame (upper left corner). Here you can select a specific package, or you can choose all classes in all packages.
2. Class frame (lower left corner). This frame lists either all the classes, or all the classes in a selected package.
3. Class Details frame (right frame). This frame shows you all attributes, constructors, and methods for the class that you choose in the Class frame.

Example:

1. Go to the J2SE API documentation, either through Sun's website or your own copy if you downloaded the docs to your hard drive (the index.html file in the docs/api directory is the file you want). By default, all classes will be listed in the Class Frame.
2. In the Package frame, click on '`java.lang`' under the package list. The Class frame will update to show only those classes in the package `java.lang`.
3. In the Class frame, click on the class *Object*. The Class Details frame will update to show some basic info about the Object class. It will also show you a summary of the constructors and methods for the class. Below the summary you will find detailed information about each of the constructors and methods: what the return value is, what parameters are required, and what the methods do.

The packages we are most likely to encounter in this course are:

1. `java.lang` (the namespace for this package is automatically included; no import statements needed for these components)
2. `java.io` (components for handling input/output from standard in/out, files, etc.)
3. `java.util` (data structures, date and time components)
4. `java.awt` (Java's basic GUI component package)
5. `javax.swing` (more or less platform-independent GUI components)

The API docs for J2SE 11 are a bit trickier to follow since they are organized by module, and there are no frames. Modules are a new feature that began with J2SE 9 that give you another layer in which to organize classes. The locations of the packages listed above in the J2SE 11 API are as follows:

1. `java.lang`, `java.io`, and `java.util` are all located in the module, **java.base**
2. `java.awt` and `javax.swing` are both located in the module, **java.desktop**

If you have trouble finding a specific class you can always use the API index to find the class you need. Clicking on the class name will tell you which module the class is in.