CS100J QuickStart Guide to Eclipse

Cornell University Summer 2004

First Launch:

The first time you launch Eclipse you are immediately prompted to specify a "Workspace." The "Workspace" is the location where all your projects and their corresponding files will be stored on your computer's hard drive. The default directory will suffice for most users, but if you have a reason to change this, then go ahead and do so now. If you want to avoid this prompt on future launches of Eclipse, then you should also check the box next to "Use this as the default and do not ask again." Continue and you are greeted by a welcome-screen. You should select the "Workbench" icon on this screen.

Workbench:

The window and icon layouts in Eclipse are called perspectives. You will want to work in the "Java" perspective. You can change perspectives by selecting ***Window --> Open Perspective -> Java***.

Now you are ready to create a new project. Select ***File --> New --> Project***. Make sure ***Java Project*** is selected and then select ***Next***. Name your project something relevant and now select *Finish*. You will see a folder with the name of your project added to the ***Package Explorer*** window.

To add files to the project, select ***File --> New --> File.*** Then select the project in which you wish to add the file and give the file a name. (Note: The name of this file should be the same as that of the class it will contain, also be sure to append the .java extension to the filename. This ensures that Eclipse recognizes the file as housing java code.) If you've already created some 'projects', then you'll be asked at this stage to highlight the project you want this file to be part of. Opening your project's folder in the ***Package Explorer*** window, you will find a *(default package)* subfolder, in which is your new file. The empty file will also have opened in a blank tab inside the center of the Eclipse window. Now you're ready to start coding.

Errors:

As you code, you may notice that Eclipse puts red squiggly lines underneath portions of your code. These squiggles will be complimented by some sort of icon next to the beginning of that line of code. Depending on the icon, this could indicate several things. A red circle with an **X** in the middle indicates a syntax error, a light bulb with an **X**-ed out red square indicates some other error, and a light bulb with a **!** inside a triangle indicates a suggestion that Eclipse wants to make about your code. If you are in the middle of a line or a block of code, then you should finish that section before investigating errors, as they may go away upon completion. However, it will most often be case that Eclipse has spotted an error in your code. To find out more information about the problem, you can hover your mouse over the icon next to the line or look inside the ***Problems*** tab at the bottom of the window. Doing so gives you an error description and, by clicking on the icon, Eclipse will list recommended fixes for the problem. If the suggested fix is the one you want to make, then you can double-click on it and the changes will be made.

Running:

Once you're ready to run your program, select the file containing your main class in the ***Package Explorer*** and then ***Run --> Run As --> 2 Java Application***. This will launch your program and all interactions between you and the program will take place in the *Console* tab at the bottom of the screen. If there are errors in your program and it cannot be compiled or run, Eclipse will alert you via the *Console* window. Notice that there are several tabs near the bottom of your window (in the default arrangement), these being "Problems", "Javadoc", "Declaration" and "Console". In order to see the 'standard output' from your program, you will need to click on the "Console" tab. You'll also find that you can use the green button to run your program (there are various short-cut icons and short-cut keystrokes which you'll discover).

Extras:

If you wish to supply command line arguments to a program you're going to run, then follow as above, except you should now select ***Run --> Run***. In the new window, click the ***Arguments*** tab and click the ***Variables…*** button. Select ***string_prompt*** from the list and select ***OK*** and then ***Run***. From now on, Eclipse will prompt for command line arguments before the program begins to execute.