


Getting Started in Eclipse for CSE332

We want you to work on your projects in [Eclipse](#), a powerful environment for Java and a number of other languages. Eclipse may seem like overkill for CSE332 - probably because it is. But as projects get larger, having an integrated development environment with lots of features will come in handy, and learning new tools is always helpful.

You can use Eclipse in the lab, or download it to your personal machine. The [download site](#) offers a number of different versions; you'll want 'Eclipse IDE for Java Developers.' (We do not recommend trying to run Eclipse remotely, for example over an X connection, because it would react too slowly.)

Running Eclipse

Linux machine in basement lab: Open Eclipse by going to Applications -> System Tools -> Terminal and, at the prompt, typing:
`eclipse &`

Windows machine: Open Eclipse by clicking on its icon, either on the desktop or taskbar: . There may also be a link in the Start Menu.

Workspaces

If this is the first time you've run Eclipse, it will ask you to choose a location for a workspace. The default location is probably fine.

A note on workspaces: Workspaces in Eclipse are useful for organizing your files; you may have a workspace for cse332, another for cse341, one for your personal projects, etc. Within a workspace you can create numerous *projects*; each project can contain various Java files that interact together. In cse332, for instance, it makes sense to create a different project for course project 1, course project 2 and course project 3, all inside your workspace for cse332.

The package explorer & Creating a project

Upon running Eclipse you should see a pane labeled 'Package Explorer', probably on the left. If you do not see one, you can open it up by going to the **Window** menu, selecting **Show View** and **Package Explorer**. If it doesn't show up under **Show View**, select **Other**, then 'Java', then 'Package Explorer'.

The Package Explorer lists each project in the current workspace. To create a new project, right click in the package explorer pane and select **New**, then **Java Project**; you'll be asked to provide a name for your project (something like project1 for the first course programming assignment is reasonable), then click **Finish** and your project will appear in the package explorer.

If an icon in the package explorer looks like a blue folder, the project is closed and can be opened by double clicking on it. To close a project (say, to save memory if you have numerous projects available), click right on it and select **Close Project**. You can traverse the tree of an open project to view the packages and Java files it contains, and you can double click a file to open it up. Eclipse projects can get pretty big, with multiple packages and large numbers of classes, so the organization provided here can be quite helpful.

Creating a new class via the package explorer

Just right click on the package in the package explorer that will contain your new class, and select **New**, then select **Class**; Eclipse will create the file with the proper package header (if any) and the class statement.

Compiling

To compile, go to the **Project** menu and select **Build All** (the hot-key for which is Ctrl-b). Your project may be set to 'Build Automatically' (in the **Project** menu) - if so, **Build All** is disabled, as it happens, well, automatically. If there are errors in your code, the files will be marked in the package explorer with a red icon. When you are viewing a file, lines with errors are marked (to the left) with a red icon; hover the mouse cursor over the icon to see the error.

Also, Eclipse checks your program as you type, marking errors and warnings, so you may see errors appear before you even 'build'. This helps you debug as you go, instead of all at once when you compile.

Running the program

Once you've got your program compiled, you need to create a 'Run Configuration' to run it. A Run Configuration specifies which class's main() method to run, and what arguments to pass in.

Open the **Run** menu and select **Run Configurations**. Double click on 'Java Application' in the list; this creates the configuration and from there you can give your run configuration a particular name and select the 'Main class' to run from. Select the 'Arguments' tab if you want to fill in arguments to be passed to your program as it runs, and enter them in the 'Program arguments' box.

Once you are done creating your run configuration, you can run it by selecting **Run** from the **Run** menu, or hitting Ctrl-f11. To terminate a running program, go to the **Run** menu and select **Terminate**.

Note: If your program executes without outputting to the console, creating a window or the like, you may not get any indication that it's running.

Adding existing files to a project

Eclipse makes it easy to add existing files (such as those provided for project 1) to a project. First locate the files on the file system; drag and drop them onto the package in which you want to add them.

Getting the files from Eclipse for turn-in (or backup or whatever)

To find out where Eclipse is keeping your files, right click on the file in the package explorer and select 'Properties'; the location is shown there. You can also export the entire project to a directory of your choosing by right clicking on the project in the package explorer and selecting **Export**. In the window that appears, select 'General' and then 'Filesystem', and, finally, select the location you want to export to.

Java 1.7

For the projects in this course, you'll need Java 1.7 (technically, 1.6 will work, but will be more of a hassle to use for project 3). To check the version of Java being used by a project in Eclipse, explore its tree in the package explorer until you see something like 'JRE System Library'. It should list '1.7'. If it lists an earlier version of Java, right click and select **Properties** (of 'Configuration' in some versions), and set the 'Execution Environment' to JavaSE-1.6.

It's possible that Java 1.7 simply isn't installed on the machine; to install it, go to [sun's homepage](http://www.oracle.com/technetwork/java/javase7/index.html) and download and install the JDK7. You may still need to change the version of Java used in your project as described above. Finally, you may need to change the compiler settings to use Java 1.7 settings; to do this, right click the project name, select **Properties**, go to 'Java Compiler' and make sure things are set to 1.7.

Anything we missed?

This introduction is intended to give you a smooth start to Eclipse, so if there's anything confusing or missing, let

us know and we'll try to get it covered.