

CSE 331 Software Design and Implementation

Handout T4: Eclipse Reference for CSE 331

Contents:

- [Introduction](#)
- [Eclipse Quick Reference](#)

Introduction

Eclipse is a multi-platform integrated development environment with strong support for Java. In addition to various features that facilitate the writing of Java code (such as autocompletion and code refactoring), it has built-in support for JUnit, and Ant - tools that we use in CSE 331.

Eclipse (version 4.2) is already installed in UW CSE instructional lab. See [here](#) for how to use Eclipse at home.

You can read more about and get help on the features of Eclipse in [its online documentation](#).


Eclipse Quick Reference

Related pages:

- [Version Control](#)
- [Editing, Compiling, Running, and Testing Java Programs](#)

Here are some tips for things that can make your life easier in Eclipse:

| | |
|----------------------------|--|
| Autocomplete | <p><code>Ctrl-Space</code> asks Eclipse to help you complete some code you've started. Eclipse can complete lots of things:</p> <ul style="list-style-type: none"> • variables, method names, class names <ul style="list-style-type: none"> ◦ <code>ArrayL</code> <code>Ctrl-Space</code> --> <code>ArrayList</code> ◦ <code>random.next</code> <code>Ctrl-Space</code> --> <code>random.nextInt</code> • constructor and method parameters <ul style="list-style-type: none"> ◦ <code>new ArrayList(</code> <code>Ctrl-Space</code> --> popup menu of <code>ArrayList</code>'s constructors ◦ <code>random.nextInt(</code> <code>Ctrl-Space</code> --> tooltip showing <code>nextInt</code>'s parameters • methods to override <ul style="list-style-type: none"> ◦ <code>class Foo extends Bar {</code> <code>Ctrl-Space</code> --> menu of <code>Bar</code>'s methods that can be overridden |
| Organize import statements | <p><code>Ctrl-Shift-O</code> (that's O as in Organize) automatically updates the <code>import</code> statements at the top of your class, adding classes that need to be imported from other packages and removing classes that you're no longer using. If a class name is ambiguous - e.g.,</p> |

| | |
|---|--|
| | List might be either <code>java.util.List</code> and <code>java.awt.List</code> - then Eclipse pops up a dialog asking you which one you want. |
| Look up Java API documentation | <code>Shift-F2</code> when your cursor is on a class or method name (You have to configure this feature with the location of the API documentation). |
| Comment/uncomment a block of code | <code>Ctrl-/</code> comments the highlighted region. <code>Ctrl-\</code> uncomments the highlighted region. |
| Renaming or moving packages, classes, methods, and variables (updates all references) | <code>Alt-Shift-R</code> (Rename) or <code>Alt-Shift-V</code> (Move) when your cursor is over the package/class/method/variable. Equivalently, right-click on any package, class, method or variable in the Package Explorer and select Refactor » Rename to rename it or Refactor » Move to put it in another package or class. |
| Delete current row | <code>Ctrl-D</code> when cursor is in row to be deleted. |
| Mark TODO items for yourself | Start a comment with TODO to leave yourself a note about a piece of code that you need to fix. Eclipse will automatically put the comment in the Tasks pane, the pane where it shows your compile errors. (If you don't see the Tasks pane, use Window » Show View » Tasks .) You can jump to TODO items or compile errors in your code quickly by double-clicking on them in the Tasks pane. |
| Generate get() and set() methods | Make sure the fields for which you would like to create get() and set() methods are declared in the class, then right-click and use Source » Generate Getter and Setter . |
| Run classes or unit tests that you've run recently | The little "play" icon  on the Eclipse toolbar runs the last class or unit test you just ran. Pull down its menu for your recent history of runs. |
| Turn off console autoraus | When a program run under Eclipse writes to standard out (say, via <code>println</code>), Eclipse raises the "console" perspective. This may be annoying when you run tests, or when you use Continuous Testing. To turn this off, go to Window » Preferences » Run/Debug » Console and uncheck "Show when program writes to standard out". |
| Emacs key bindings | If you prefer to use Emacs key bindings while editing code, do: Window » Preferences... » General » Keys » Modify and set Scheme to Emacs |

For problems or questions regarding this page, contact: cse331-staff@cs.washington.edu.