6.005 — Software Construction on MIT OpenCourseWare (https://ocw.mit.edu/ans7870/6/6.005/s16/) | OCW 6.005 Homepage (https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-005-software-construction-spring-2016/) **Spring 2016**

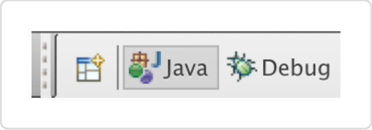
Eclipse FAQ

Resources

Java Debugging with Eclipse (//www.vogella.com/tutorials/EclipseDebugging/article.html) introduces Eclipse's debug mode, a recommended tool for tracking down bugs using breakpoints. This step-by-step tutorial explains the *Debug perspective* and provides screenshots to decode the mysterious debugger icons in Eclipse.

FAQ

I messed up my Eclipse and it doesn't show all the things it used to show

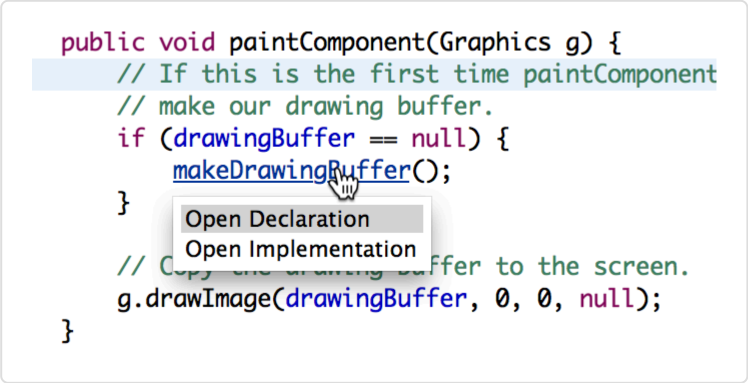


First, make sure you are in the *Java perspective* : in the top right corner of the window, click the **Open Perspective** toolbar button and select **Java** .

If your Java perspective doesn't look right, go to **Window** → **Reset Perspective** to return to the Eclipse defaults.

Tips & Tricks

Navigating



```
public void paintComponent(Graphics g) {
    // If this is the first time paintComponent
    // make our drawing buffer.
    if (drawingBuffer == null) {
        makeDrawingBuffer();
    }

    // Copy the drawing buffer to the screen.
    g.drawImage(drawingBuffer, 0, 0, null);
}
```

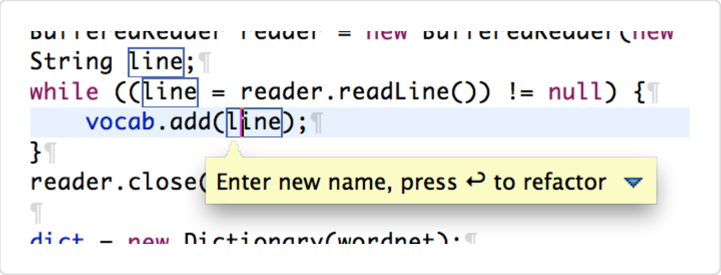
Command-click (on Mac) or control-click to **jump straight to the definition** of a variable, field, method, class, etc.

Return to the previous file you were working on with **Navigate** → **Back** , which might have keyboard shortcut Alt–left-arrow or Command-[. Eclipse's *Navigate Back* and *Forward* are analogous to the back and forward buttons in your web browser.

Command- (or control-) -shift-R and command-shift-T bring up a dialog for quickly searching for and **navigating to files** (a.k.a. resources) **and types** (classes), respectively.

Editing

Select an element and use ***Refactor*** → ***Rename*** (or its keyboard shortcut) to rename that variable, field, method, class, etc. wherever it appears.



```
BufferedReader reader = new BufferedReader(new
String line;¶
while ((line = reader.readLine()) != null) {¶
    vocab.add(line);¶
}¶
reader.close(
¶
dict = new Dictionary(wordnet);¶
```

Enter new name, press ↩ to refactor

Take advantage of **code completion** to help you use a new API: if you have an object `squiggle` , typing the name and then a period will bring up code completion options.

Use **Source** → **Organize Imports** or command- (or control-) -shift-O to automatically add and organize Java import statements. Just start using a new class, *Organize Imports* , and you're good to go.

Don't like formatting and indenting your code manually? Use ***Source*** → ***Format*** to clean up your act.

Use **Source** → **Toggle Comment** or keyboard shortcut Command-/ to quickly toggle on or off some lines of code.

Use **block selection** (//blog.jooq.org/2013/10/12/eclipses-awesome-block-selection-mode/) to edit across multiple lines at once.

Have fun in 6.005!

MIT EECS