

8 Most Useful Keyboard Shortcuts for Beginner Eclipse IDE Users

1. Organize Imports (Ctrl + Shift + O)

Whether you're working with Java's native libraries and classes or incorporating third-party frameworks into your code, one thing is true: in order to use a class, you have to first import the class before Eclipse will identify it as valid and available for code autocompletion (real-time typing suggestions).

But who has time to memorize every single package path for every single class in every single library? You can let Eclipse handle it for you using the Ctrl + Shift + O shortcut, which automatically imports unrecognized classes in code.

Instead of typing import lines by hand, you can just write code as normal until you see the **red squiggly lines** (indicating unrecognized classes), then hit the Organize Imports shortcut.

```
public class Hello {  
    public static void main(String[] args) {  
        ArrayList list = new ArrayList();  
    }  
}
```

Now use the Organize Imports shortcut, **(Ctrl + Shift + O)** it becomes like this:

```
import java.util.ArrayList;  
  
public class Hello {  
    public static void main(String[] args) {  
        ArrayList list = new ArrayList();  
    }  
}
```

2. Correct Indentation (Ctrl + I)

Code readability is important, not only for you (so you can come back at a later time and understand what you wrote) but for anyone else who may look at your code (partners, professors, open source contributors). Proper indentation is essential.

Maybe you wrote it that way, or maybe you copy-pasted from elsewhere. Either way, the good news is that Eclipse makes it trivially easy to fix: highlight the portion of code that's hard to read, then use the Ctrl + I shortcut to instantly bring it to proper indentation.

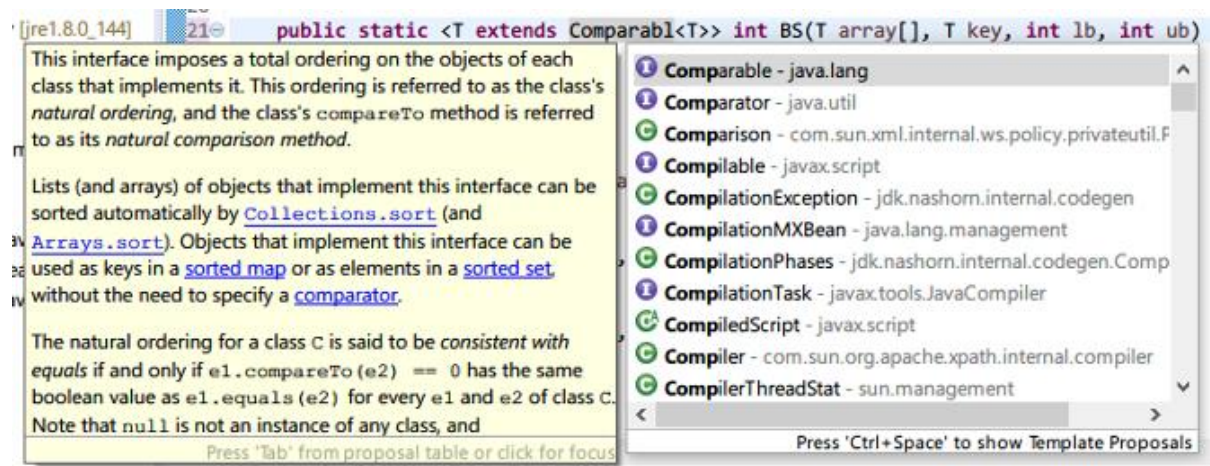
3. Delete Current Line (Ctrl + D)

When coding in Java, it's natural to remove entire lines of code at a time. The worst way to do this? Highlight with the mouse, then hit Backspace. The rookie way to do this? Hit the End key, hold Shift, hit the Home key, then Backspace. But the pro way? Simply hit Ctrl + D.

4. Autocomplete Recommendation (Ctrl + Space)

Java is unfortunately known for being extremely verbose — the names of classes, methods, and variables are some of the longest in the entire programming industry. Typing them all by hand every single time? Not my idea of a fun time.

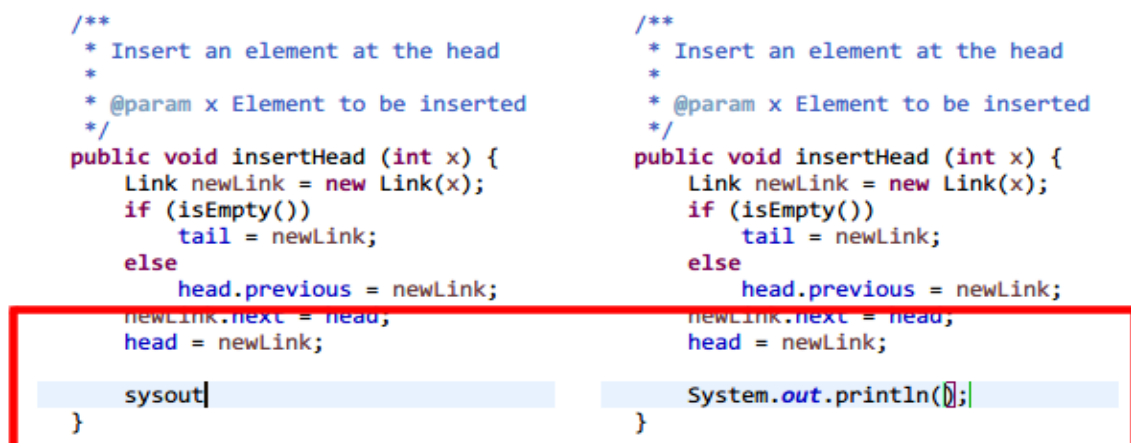
Here's what you do instead: type the first few letters of the class, method, or variable you want, then hit Ctrl + Space. This brings up a list of autocomplete recommendations along with method signatures, variable types, and more. Select the right one, hit Enter, and keep coding.



5. System.out.println ("sysout" and Ctrl + Space) :

When working with console applications, you'll need to use `System.out.println()` for printing messages. But because this is so cumbersome, Eclipse has a quick shortcut for you: type "sysout" (without the quotes), then hit Ctrl + Space.

The best part? The cursor is immediately placed within the method call's parentheses, so you start typing the message right away:



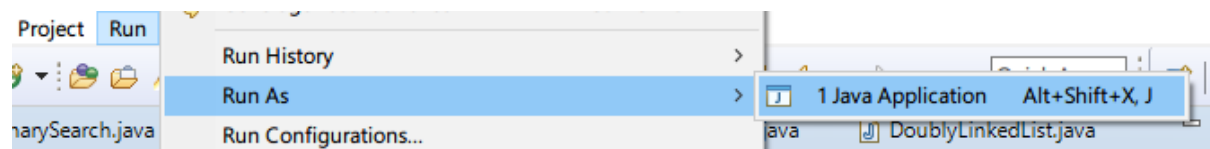
6. Search Entire Project (Ctrl + H):

When working on large codebases, it's easy to forget where you declared certain classes, methods, or variables. Instead of wasting time combing through directories by hand, use the Search Entire Project prompt with the Ctrl + H shortcut.

By default, it comes with four search types: File Search, Task Search, Git Search, and Java Search. You'll mostly use Java Search, which only searches through source files, but the other three can be useful in their own ways.

7. Run Application (Ctrl + F11):

The first time you run a new project, you should do it through Run > Run As... > Java Application. But after that, you can speed things up with the Ctrl + F11 shortcut, which runs the current project using the same configuration as the last time the project ran.



8. Rename (Alt + Shift + R) :

Here's the thing about class, method, and variable names: once declared, they can get referenced dozens, hundreds, or even thousands of times throughout a project. Now imagine if you ever need to change the name of a class, method, or variable. It could take hours (or days!) renaming every single reference.

Or you can right-click on the name, select Refactor > Rename, type in the new name, and have Eclipse change every single reference in the entire project in a second. Even faster, you can click on the name, hit Alt + Shift + R, type in the new name, and hit Enter. Bam, done!

Other Tips for Beginner Java Programmers:

As a Java programmer, you should absolutely understand how the Java virtual machine works and why it allows for cross-platform development. You should also be familiar with these core Java concepts, such as how to use exceptions in Java.

#####