***Windows->Preferences->General->Keys*. (To see all short cuts)**

"sysout" and **CTRL + Space** 🡪 System.out.println()

**CTRL + H** 🡪 Search Entire Project

**CTRL + SHIFT + O 🡪** Organize imports.

**CTRL + 1 🡪** It activates the quick fix.

**CTRL + SHIFT + T 🡪** Open Type. To look at the "Foo" class in the "Boo" project or in the "foo.bar" package.

**CTRL + F11** 🡪 Run Application

**CTRL + N 🡪** Open a new type wizard.

**CTRL + M 🡪** Maximize or unmaximize your current tab.

**CTRL + I 🡪** Corrects indentation.

**CTRL + SHIFT + F 🡪** Formats code.

**CTRL + L 🡪 E**nter line number to go to line number N in the source file

**CTRL + ALT + H** 🡪 Open Call Hierarchy. Find out where that method is used.

**F4** Open Type Hierarchy. See the subclasses and/or super classes.

**F3** Open Declaration. Jump to the declaration of the variable, method or class.

**ALT + SHIFT + R** Refactor->Rename. Works in the editor and in the package explorer

**CTRL + D** --> Deletes current line.

**CTRL + DELETE -->** Deletes next word after the cursor.

**CTRL + SHIFT + DELETE -->**Deletes from the cursor until end of line.

**CTRL + Backspace -->** Deletes previous word before the cursor.

**CTRL + SHIFT + y -->** Changes a selection to lowercase.

**CTRL + SHIFT + x -->** Changes a selection to uppercase.

**ALT + Left/Right Arrow 🡪** Move to the last location you edited.

**ALT + Up Arrow** --> Moves up current line (or a selected code block) by one line -->

**ALT + Down Arrow** --> Moves down current line (or a selected code block) by one line -->

**CTRL + ALT + Up Arrow -->**Copies and moves up current line (or a selected code block) by one line -->

**CTRL + ALT + Down Arrow -->**Copies and moves down current line (or a selected code block) by one line

**SHIFT + ENTER -->** Inserts a blank line after current line, regardless of where the cursor is at the current line (very different from press **ENTER** key alone) -->

**CTRL + SHIFT + ENTER -->** works similar to the **SHIFT + ENTER** but inserts a blank line just before the current line.

**CTRL + SHIFT + M -->** Adds a single import statement for the current error due to missing import.

**CTRL + /**or **CTRL + 7** --> Toggle single line comment. This shortcut adds single-line comment to current line or a block of code. Press again to remove comment.

**CTRL + SHIFT + / -->** Adds block comment to a selection.

**CTRL + SHIFT + \ -->** Removes block comment.

**CTRL + o** 🡪 for quick outline going quickly to a method

**Alt + SHIFT + W 🡪** to show a class in package explorer

**CTRL + SHIFT + Up**and down 🡪 for navigating from member to member (variables and methods)

**CTRL + k**and **CTRL + SHIFT +K 🡪**for find next/previous

### Eclipse Shortcut for Editing Code

**CTRL + / 🡪**for commenting uncommenting lines and blocks

**CTRL + SHIFT + / 🡪** for commenting, uncommenting lines with block comment, see here for example.

Selecting class and pressing **F4** to see its Type hierarchy

**CTRL + l 🡪 G**o to line

**CTRL + F 🡪** for find, find/replace

**CTRL + Q 🡪** for going to last edited place

### Miscellaneous Eclipse Shortcuts

**CTRL + T** 🡪 for toggling between supertype and subtype

Move to one problem (i.e. --> error, warning) to the next (or previous) in a file -->**CTRL +.**For next, and CTRL +, for the previous problem

**CTRL + SHIFT + G** 🡪 which searches the workspace for references to the selected method or variable

**ALT + SHIFT + j 🡪** to add Javadoc at any place in Java source file

**CTRL + SHIFT + P 🡪** to find closing brace. Place the cursor at the opening brace and use this.

**CTRL + E 🡪** Shows you a list of all open editors.

**CTRL + Q** 🡪 *Go to the last edit location*

**CTRL + F4 or CTRL + w 🡪** for closing current file

**CTRL + SHIFT + W 🡪 for** closing all files.

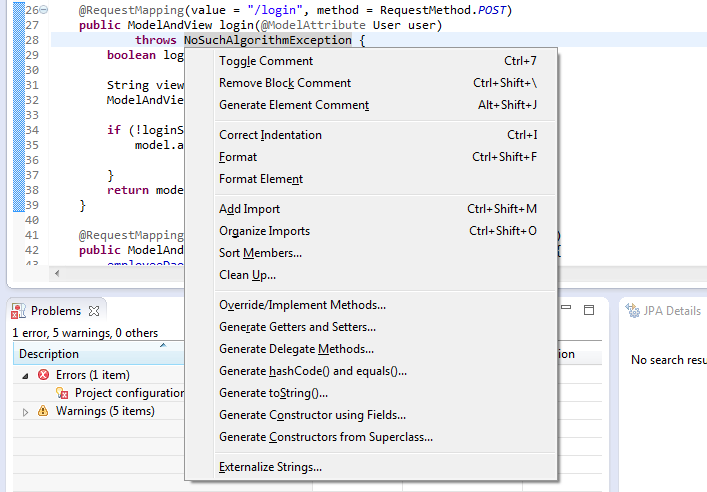
**CTRL + F7 🡪** Move between views.

**CTRL + F8 🡪** Move between perspectives.

**CTRL + SHIFT + L 🡪** Shows you a list of your currently defined shortcut keys.

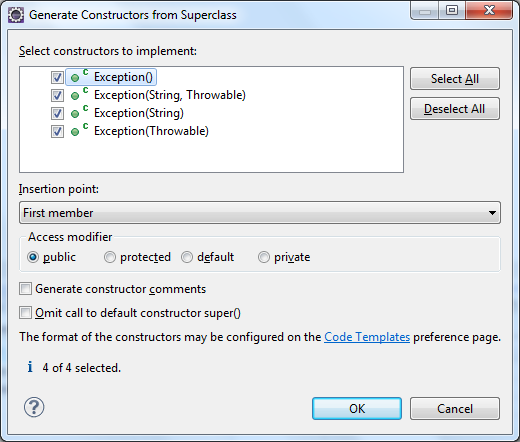
**ALT + SHIFT + V** 🡪 Refactor->Move. Again, works in both the editor and package explorer. Move the method or field to another class.

**ALT + SHIFT + M** 🡪 Extract to method. Break up that monolithic code, takes the selection and try to make it into a method. Will optionally find duplicate code blocks and use the new method there as well. It might take a few try to get this to work the way you want. Just try it, rearrange the code a little and try again.

**Alt + SHIFT + S -->** Shows context menu that lists possible actions for editing code   
From this context menu, you can press another letter (according to the underscore letters in the names) to access the desired functions.

**ALT + SHIFT + S + R -->** Generates getters and setters for fields of a class.

**ALT + SHIFT + S + O -->**Generates constructor using fields. This shortcut is very useful when you want to generate code for a constructor that takes class’ fields as its parameters.

**ALT + SHIFT + S + C -->** Generates Constructors from Superclass. A common example for using this shortcut is when creating a custom exception class. In this case, we need to write some constructors similar to the **Exception**superclass. This shortcut brings the *Generate Constructors from Superclass* dialog which allows us to choose the constructors to be implemented in the subclass -->

**ALT + SHIFT + S + H -->** Generates [hashCode() and equals() methods](https://www.codejava.net/java-core/collections/understanding-equals-and-hashcode-in-java" \t "_blank), typically for a JavaBean/POJO class. The class must have non-static fields. Select the fields to be used in **hashCode()** and **equals()** method, and then click OK.

**ALT + SHIFT + S + S -->** Generates **toString()** method.