Contents

[Week 1 Tips 1](#_Toc512702403)

[**Project - Name** 1](#_Toc512702404)

[**OO - Online Tutorials** 1](#_Toc512702405)

[**Java Coding Conventions** 1](#_Toc512702406)

[How to have Eclipse warn of missing Javadoc comments? 2](#_Toc512702407)

# Week 1 Tips

### **Project - Name**

Please ensure that your project name has no spaces in it. For example, it should be ‘FleetRental’ and not ‘Fleet Rental’.

**In addition to the project name, do ensure that none of your folders have a space/blank in its name!**

Reason being in weeks to come, we’ll be referring to a file (application properties) in our project where we have include the entire path to the file and spaces can cause havoc!

### **OO - Online Tutorials**

**Object-Oriented Programming Concepts:** <https://docs.oracle.com/javase/tutorial/java/concepts/index.html>

### **Java Coding Conventions**

Software can be so subjective, unlike in other engineering disciplines, for example Electrical Engineering, the IEEE standards(<http://en.wikipedia.org/wiki/Reference_designator>) are enforced, which forces us to use the letter ‘R’ for Resistor, ‘C’ for Capacitor etc. with no room for variation.

Unfortunately, software engineering is still a relatively new field and standards (such as naming conventions of variables) tend be reflected based on one’s background (assembler programmer, FORTRAN, C, C++ …). Hence you see variation in opinions.

How might we solve this issue?

Well, my take is since are coding in Java, for guidance:

1. Here is Java standard Google uses for Java development (influenced by K-R style)

<https://google.github.io/styleguide/javaguide.html>

1. For specific sample, let’s inspect Java classes in the JDK.
   1. <http://www.docjar.com/html/api/java/lang/String.java.html>
   2. <http://www.docjar.com/html/api/java/io/File.java.html>

I provided links to the source code above as a quick reference and hence they are not meant to be comprehensive. String is perhaps the most used class in Java and the most quizzed in interviews and studying the source code is one way to look under the hood.

### [How to have Eclipse warn of missing Javadoc comments?](https://worldclass.regis.edu/d2l/le/186824/discussions/threads/912223/View)

In the weeks to come you’ll see HW feedback from me requiring that all code be documented using ‘Javadoc tags’. See link for how to document using Javadoc comments -<http://www.oracle.com/technetwork/java/javase/documentation/index-137868.html>

**Note:**What I’m requiring is document using Javadoc tags and not about running the Javadoc Tool (included in the JDK) which generates a HTML documentation of your code.

Eclipse has way to have it warn you of missing Javadoc comments. Here are the steps:

Bring up Eclipse and choose Project -> Properties, scroll down to Compiler -> Javadoc.

Check "Enable project specific settings", and then check "Process Javadoc comments". Change the settings for "Malformed Javadoc comments", "Missing Javadoc tags" and "Missing Javadoc comments" to Warning. Now hit OK.

This causes Eclipse to produce warnings anytime you haven't fully Javadoc'd something, thereby helping you to do better in documenting your homework.