Prof Bill’s Java Coding Guidelines

Last update: Mar 2014

Your code must be organized, readable and easy to understand. Guidelines, not rules.

1. Most important - **Think, then code**. Sketch out your plan on the back of a napkin, write some pseudo-code, draw UML, or whatever, *before* you sit down in front of your tube.

2. **One** public class or interface per file - This is a Java convention.

3. Write **Javadoc** comments for each class and method - Javadoc is a standard. In Netbeans, start your comment with `/**` and hit enter… NetBeans will start a template for you.

4. Make **class variables** private or protected - This is a common object-oriented paradigm. Access to class variables is often provided by accessor (set) and mutator (get) methods.

5. Use **camel notation** for class, method, and variable names. This is a Java convention. Camel notation starts new words with an upper case letter. For example: `professorPayRaise`.

6. **Capitalize** class, interface and package names. This is a Java convention. For example: `MonsterTruck`.

7. Use all **UPPER CASE** for constants, separating words with an underscore. This is a Java convention. For example: `DEATH_RAY_VOLTAGE`.

8. Use a consistent style for **spacing**, **indentation**, and **curly braces**. This makes your code more readable. The default on your IDE (like NetBeans) is probably fine.

9. Use **inline comments** to explain difficult sections of your code. This makes your work more readable and easier to understand.

You can be more thorough and follow **Oracle’s Code Conventions** if you like: www.oracle.com/technetwork/java/codeconv-138413.html