

# Coding Style and Documentation Guidelines

Steven Castellucci



# Naming Convention

- ▶ Constants should use all uppercase letters, with an underscore between words
  - E.g., MAX\_INT, UPPER\_LIMIT
- ▶ Variables, methods, and packages should use lowercase letters (use uppercase letters to start second and subsequent words)
  - E.g., length, indexOf, getName, type
- ▶ Classes should capitalize the first letter of each word
  - E.g., String, Integer, StringBuffer, TreeMap

# Statement Layout

- ▶ Write only one statement per line
- ▶ Wrap lines longer than 80 characters
- ▶ Leave exactly one space around operators
  - E.g.,  $x = a + 5;$
- ▶ Indent each block one tab (approx. 4 spaces) relative to its braces

# Braces

- ▶ Always use braces in control structures, even if the block contains only one line
- ▶ Vertically align corresponding opening and closing braces

# Magic Numbers

- ▶ Avoid the use of hard-coded numeric literals (a.k.a., “Magic Numbers”)
- ▶ Exceptions:
  - Declaring a constant in a “final” statement
  - Using the literal zero, +/– one, or +/– two

# Full Style Guide with Examples

[www.eecs.yorku.ca/teaching/docs/type/style.pdf](http://www.eecs.yorku.ca/teaching/docs/type/style.pdf)

# Documentation

- ▶ Using in-code comments as documentation has the following benefits:
  - Easier (more concise) to read than code
  - Moves with the code
    - Easier to reference
    - Easier to maintain (update/correct)
  - Can explain design choices, rejected techniques, reasons for code modifications

# Layouts

- ▶ **Single-Line Comments**

```
// This is a single-line comment.
```

- ▶ **Multi-Line Comments**

```
/* This comment can persist  
   over multiple lines as needed. */
```



# JavaDoc Example

```
/**
 * This function takes as its argument an integer C,
 * and returns the smallest integer that is a factor
 * of C, other than 1.
 *
 * @param C An integer to factor.
 * @pre. C must be greater than 0.
 * @return The smallest factor of C.
 */
public int smallestFactor(int C)
{
    ...
}
```

# JavaDoc Example (2)

## **smallestFactor**

```
public static int smallestFactor(int C)
```

This function takes as its argument an integer C, and it returns the smallest integer that is a factor of C, other than 1.

### **Parameters:**

C - An integer to factor.

### **Returns:**

The smallest factor of C.

### **Precondition:**

C must be greater than 0.

# JavaDoc Template for 1030

```
/**
 * Description about the method
 *
 * @param paramName ParamDescription
 * @pre. PreconditionDescription
 * @return ReturnDescription
 */
public rtnType methodName(paramType paramNameC)
{
    ...
}
```

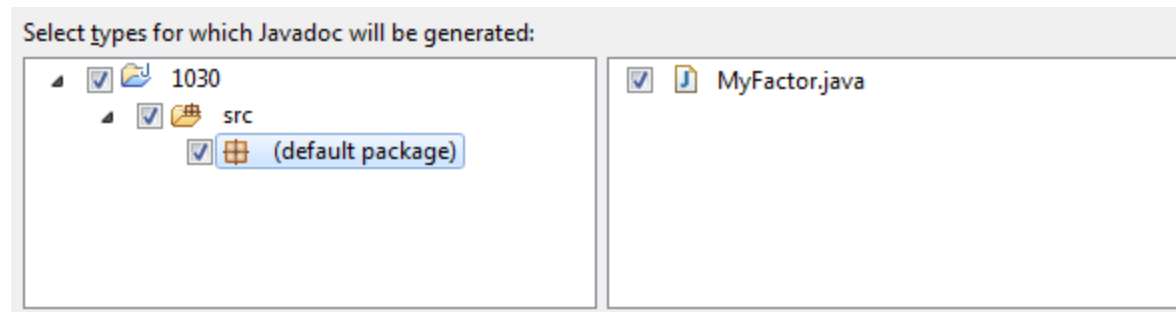
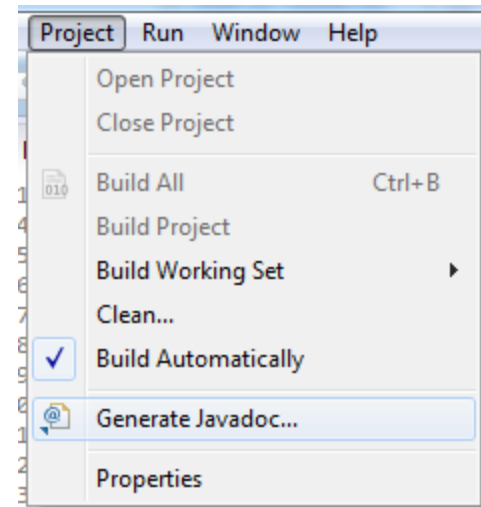
# Generating JavaDoc (Terminal)

- ▶ Navigate to the src directory containing your .java files and enter the following command:

```
javadoc -d ../doc -tag pre.:a:"Precondition: " *.java
```

# Generating Javadoc (Eclipse)

- ▶ Project > Generate Javadoc
- ▶ Select your class (e.g., MyFactor.java)
- ▶ Click “Next” twice



# Generating JavaDoc (Eclipse) (2)

- ▶ Enter the JavaDoc options for the custom “pre” tag:

Extra Javadoc options (path names with white spaces must be enclosed in quotes):

```
-tag pre::a:"Precondition: "
```

- ▶ Click “Finish”
- ▶ Open the html page (MyFactor.html) in the newly created doc directory