# Style Sheet for Java Programs

Most companies have requirements for layout and formatting of code. Code that adheres to a uniform style is easier for others to read.

You are required to use the guidelines specified here in all programs that you write for this class.

**Basic Formatting Rules**

* Class names start with an uppercase letter . Example: **Dog** class.
* Variable method names are lowercase, except that embedded words may have the first letter capitalized. Example: **studentNumber**, or **getSize**.
* Constant names are all uppercase. Underscores may be used to separate words. Example: **TAX\_RATE**.
* Variable names should be fairly long and descriptive. For example, use **studentNumber** (or perhaps **stuNum**), but not **sn**.
* Binary operators and “equals” signs must have a space before and after. Example: **sum = num1 + num2**.
* A beginning brace must be on a new line. The closing brace must be aligned vertically with the beginning brace.
* Code must be properly indented, with tabs set to 3 or 4 spaces. You may need to change the tab stop setting in your editor.

**Rules for Class Files**

Generally, instance variables are private and methods are public. However, some “helper” methods – methods that are called by class methods but are not used outside the class – are private. If you do not explicitly list an access specifier, the default specifier of package is used. The protected specifier allows access for subclasses only.

Class components should be in the following order:

* class constants
* instance variables
* constructors
* methods.

**Rules for Variables and Constants in Methods**

Most variables should be declared at the beginning of the method. For variables that are used in only a few lines of code, it is permissible to declare it just before using it – but this practice should be minimized.

A variable used only for counting loop iterations should be declared close to where it is used, or, in the case of a **for** loop, it should be declared within the loop definition:

Example: **for(int counter = 0; counter < 5; counter++)**

Declare each variable on a separate line.

Example:

**int num1, num2; //bad**

Instead, use

**int num1;**

**int num2;**

Use named constants, rather than embedding numeric values in the code.

Example:

**if(rate < 0.25)… //bad**

Instead, use

**final double TAX\_RATE = 0.25**

**if(rate < TAX\_RATE)**

|  |
| --- |
| When declaring arrays, place the brackets with the type rather than the variable name.  Example: **int[] scores;** |

**Rules for Control Statements**

Indent carefully. This is especially important with nested control statements.

Always use braces to enclose the actions within a control statement, even if there is only one statement.

Example:

Do this:

**if( num < 1)**

**{**

**num = num + 1;**

**}**

But DON’T do this:

**if(num < 1)**

**num = num + 1;**

**Rules for Indentation and White Space**

Use only one statement per line.

|  |
| --- |
| **while (num < MAX\_NUM)**  **{**  **num \*= 2; num++; // Wrong**  **}**  **while (num < MAX\_NUM)**  **{**  **num \*= 2; // Correct**  **num++;**  **}** |

Use blank lines to separate code segments.

Lines should not wrap. Do not put more than 70 characters on a line. If a statement exceeds this length, break the statement and add an indentation level to the second line.

Example:

**tax\_amount = ... + ... \* ... - ... + ...**

**+ ... \* (...) + ...**