

# Style Guide for CS 149

This Style Guide should be used as a reference for all work turned in for credit, both labs and programming assignments. It is based on standard java coding guidelines. See

<http://developers.sun.com/sunstudio/products/archive/whitepapers/java-style.pdf> for a complete java guide.

**Names** - All names should be descriptive and readable. subTotal rather than s, grade rather than grd. Multiple word names should use either a Capital letter or underscore (\_) to separate words (not both). subTotal or sub\_total.

- Variable and method names should begin with a small letter. subTotal
  - Variable names should be noun phrases (studentName or subTotal)
  - Method names should be verbs or verb phrases (print\_line or addColumn)
- Class names should begin with a capital letter and use title case. HelloWorld.
- Constant names should be all caps with an underscore separator. PI, INTEREST\_RATE.

**Declarations** - Where and when to declare variables

- Constants should be declared and initialized at the top of the method or class
- All variables should be declared directly after the constants (white space sep ok)
- There must be a line of white space directly after the variable declarations.
- It is NOT permissible to initialize a variable as you are declaring it.
- Each declaration should be on its own line. Comment to the right if not clear.

**Indentation** - Provides a visual structure for your program.

- Sub sections of code must be indented a consistent 3 spaces for readability.
- Blocks of code surrounded by curly braces should have the curly braces line up vertically and the open brace should line up directly beneath the first letter of the header for that block.
- An alternate format for the block is to have the open brace on the same line as the header and the closing brace to line up with the first letter of the header.
- Statements which are too long for the line should be indented 3 space for the 2nd and subsequent lines. Lines may be no longer than 80 chars long.

**Literals and Constants**

- Numeric literals must be of the correct type for the context in which they are used.
- Constants should be used in place of meaningful literals.

**Operators**

- Binary operators should be separated from their operands by a single space on either side.
- Unary operators should not have a space.
- An exception is the dot (.) operator which does not have space surrounding it. Ex: System.out.println();
- In for loop headers, you may compress the operators.

**Structure**

- One line of white space should follow the declaration of variables.
- Use white space to separate segments of code.
- Lines should be kept to a short length (<= 80). You should be able to see the full line in your editor and in the submit system.
- Methods should be no longer than 25 lines long. (after the first PA)
- If a method returns a value, it should have a single return statement.
- Break statements should not be used except in the case of a switch.

## Class structure

- In a class declaration, constructors should precede all other methods, then the methods should be in order by visibility modifier (public, then private) then alphabetically within the same modifier..
- You may not use the break statement except for within the switch statement.
- All class variables should be private unless there is a documented reason for not doing so.
- All class constants should be public unless there is a documented reason for not doing so.
- All visibility modifiers should be explicitly stated.

**Comments** - Comments provide a guide to what the program is doing.

- Must use normal English spelling and grammar. Phrases instead of full sentences are okay.
- Must come before the code that they are describing or if brief on the same line.
- Inline comments (//) should only describe major steps within a method.

**Class comment** - every source file must contain the following at the top of the class. The \* on each line is optional, but you must clearly delineate the class comment from code.

```

/*****
* Overall description of the class goes here
*
* @author Your name goes here
* @version Vn - date
* Description of the assignment (ex: Lab1, PA1 - Mile conversion)
*****/

```

**Programming Assignments** - Every programming assignment must contain the following statement or must cite any sources used (such as a TA). This code must come directly beneath the class comment.

```

/*****
* References and Acknowledgements: I received no outside help with this
* programming assignment
*
*****
OR
/*****
* References and Acknowledgements: TA Glenn helped me with the foo method.
*
*****

```

**Methods comment** - All methods must contain a header which contains the following if applicable. Make sure that you do not give

an @param or @return tag for methods that take no parameters or which do not return a value. The \* on each line is optional, but you must clearly delineate the method documentation from code.

```

/*****
* Overall description of the method goes here
*
* @param paramName Describe each input parameter, must have one @param
line for each
* @return Describe the value that this method returns.
*****/

```