



## Coding Standards

Joel Castellanos

All projects must follow these coding standards.

Quizzes and exams do not need to follow these standards.

These standards do not represent the best nor the only good way to write Java code. These standards may not be the standards you are used to using. However, in this class, these are the standards we will use. There are three primary reasons for a standard: First, a standard makes it easier for the instructors to read your code. Second, a class standard makes it easier for a grader to recognize when you have not used a consistent standard. Often when each student is allowed to define his or her own standard, students switch standards multiple times in a single project. It is tedious for a grader to deduce each person's standard and then check for self-consistency. Third, it is good practice for you to learn to follow a standard.

As we progress through the semester, none of these standards will change, however additional standards will be appended.

1. All variable names will begin with a lowercase letter.
2. All class names will begin with an Uppercase letter.
3. All identifiers declared with the `const` qualifier will be all UPPERCASE.
4. Variables will be declared in a scope no larger than they are used.
5. No line will be more than 80 characters long.
6. Code blocks will be indented to show the block structure with *two spaces* (no tabs) per level.
7. Open brackets will be indented at the beginning of a line (not at the end).
8. Closing brackets will be indented on a line with no other commands. The only exception is that comments may be placed on the line with a closing bracket.
9. Whenever a structure spans more than one line, brackets must be used. For example:

ok	<code>if (x == 5) y=y+1;</code>
ok	<code>if (x == 5) { y=y+1; }</code>
not the standard	<code>if (x == 5) y=y+1;</code>

10. When submitting an assignment in WebCT, submit a single .JAR file. This file must include all source files and resource files used to build the executable. Do not include class files unless they are third party libraries. Include your name in the class name of whichever of your source files contains the main function that you want run during grading. For example: JoelCastellanos-Snowflake.java.

### Example Code without comments that follows the standards:

```
public static void bubbleSort(int[] array)
{ int length = array.length-1;
  boolean swap = true;

  while(swap)
  { swap = false;
    for (int i = 0; i < length; i++)
    {
      if (array[i+1] > array[i])
      { int tmp = array[i];
        array[i] = array[i+1];
        array[i+1] = tmp;
        swap = true;
      }
    }
  }
}
```