

# CS 251

## Intermediate Programming

### Coding Standards

Brooke Chenoweth

University of New Mexico

Spring 2014

# CS-251 Coding Standards

- All projects and labs must follow the great and hallowed CS-251 coding standards.
- These standards do not necessarily represent the best nor the only good way to write Java code.
- If you have experience programming, then these standards may not be the standards you are used to using.
- However, in this class, these are the standards we will use.

# Primary Reasons for Defined Standard

1. A standard makes it easier for the instructors to read your code.
2. A class standard makes it easier for a grader to recognize when a program does not use 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 persons standard and then check for self-consistency.

3. It is good practice to learn to follow a standard.

# Coding Standard: Naming

- Multi-word names are generally written in mixed-case (also known as camelCase). Internal words start with a capital letter.
- All variables not declared **final** shall begin with a lowercase letter.
- Variables that do not ever change value (that is, constants) shall be declared **final** and shall be all uppercase with words separated by underscores.
- All class and member variables (non-local variables) will be given descriptive names.
- Never use the single letter **I** nor the single letter **O** as a name.

# Coding Standard: Naming

- Method names must be descriptive (usually verbs) and start with a lowercase letter.
- All class and interface names shall be descriptive (usually nouns) and begin with an uppercase letter.

# Coding Standard - Indenting

- Code blocks will be indented to show the block structure with **four spaces** per level.  
(Note: I often use only two spaces in order to fit on the slides. Do as I say, not as I do.)
- Tab characters shall **not** be used for indenting.
- All statements within a block must be indented to the same level.

# Coding Standard: Brackets

- Open bracket is last non-space character on a line.
- Closing bracket begins a line and is indented to the beginning of the block.
- Exception: Empty class/method body.

```
public class ExampleClass {  
    private int anExampleNumber;  
  
    public static void main(String[] args) {  
        // statements go here  
    }  
  
    private void emptyMethod() {}  
}
```

# Coding Standard Blocks and { }

Whenever a structure spans more than one line, brackets must be used. For example:

```
/* OK */  
if (x == 5) y=y+1;
```

```
/* OK */  
if (x == 5) {  
    y=y+1;  
}
```

```
/* Not CS-251 standard */  
if (x == 5)  
    y=y+1;
```



# Comments

- At the top of every .java file, there must be a Javadoc comment block with your name, a description of the what the class/interface is used for and how to use it.
- At the top of every method, there must be a Javadoc comment block describing what the method does, its parameters, and its return value.
- For more information on writing Javadoc comments, see the Javadoc conventions online.

# Coding Standard 80 Character Line Max

- No line shall be more than 80 characters.
- The best way to avoid overly long statements is by not doing too much in a single statement.

# Principle of Least Privilege

In computer science, the *Principle of Least Privilege* requires that in a particular abstraction layer of a computing environment, every module must be able to access only such information and resources that are necessary for its legitimate purpose.

- Variables used in only one method shall be local variables.
- Don't make an instance or class variable public without good reason. (Note: "I couldn't make it work otherwise" is not a good reason!)

# Write Self-Documenting Code

- Self-documenting code uses well-chosen names and has a clear style.
- The program's purpose and workings should be obvious to any programmer who reads the program, even if the program has no comments.
- To the extent that is possible, strive to make your programs self-documenting.

# Java Code Conventions

For examples of how to format specific constructs and guidance on issues not covered in these slides, see [Java code conventions online](#).

## Last Word....

Use clean coding standards *as you code*.

- Of course, only the end product is graded; however, if you take the time to maintain proper indenting and formatting as you write your code, then you will find coding easier.
- When you show clean code to an instructor, the instructor is more likely to be friendly and will require less time to help you.