Intro to CompSci coding style (This was written in a *loose* tone to make it easier to read.)
By - TAs Moshe Rosensweig and Aaron Shakibpanah

[] Why should I care about coding style?

Good question. The main reason to care, is because you or someone else will have to read your code without you there to explain what every line means or does. Remember the time before you started intro to compsci, where all code looked like hieroglyphics? You're eyes kind of glazed over any code you saw. Well, if you don't use good coding style, it will be nearly that difficult to read the programs you write (especially as they become bigger and more complex). So you need to get in a good habit now.
Basically, just like it's important to write clearly and in paragraph format, etc… in your language arts classes, so too with compSci.

[1] Naming Convention

Ok, so conventions. CompSci is not naturally occurring in nature. Consequently, we don't get an exemption from NAWO (natural world core). So who determines what the rules are? I don't know, but someone did. And if you look online, there are conventions on how to do various things. As you would expect, because it's all contrived, there a disagreements about the *proper* ways to do stuff, but there are still general rules and ideas that everyone agrees on.

The first area of business is the naming convention. When you have to make a new variable, give it a good name. In most contexts (aside from a *for* loop) don't use a single letter. Don't do "int a = 1;". Later when you read the code, you'll end up with lines like "if(a < y) b++;" If that's 50 lines down, you may/will not easily remember what "a" ,"y" and "b" mean, and you'll have to waste time finding them earlier in your code and trying to figure out what they mean. Instead, if you make a file, call it something like "File hwFile = new File(input);" Later, I'll know that hwFile is a not only a file type, but I'll even be able to tell it apart from other files I've made.

[2] camelCase

The next convention is camelCase. As you can see, "camel" starts a with lower case letter and "Case" is starts with an upper case letter. This represents the naming conventions for variables. Remember above, I named it "hwFile" not "HWFile". Why? To follow the convention.

This convention has a second part. Naming classes. When you name a class (look at your old homework) - when you write "public class" it's followed with a capital lettered word. All this is helpful because it makes it clearer what is what. When you read your code, you can know if something is a class or a variable just by checking if it starts with a capital or doesn't.

[3] Indenting

This is a big one. You NEED to indent. When? When you write a block of code within another block of code. For example, consider the following code block:

```
int example = 1;
for(int i = 0;  i < 10; I++){
        System.out.println(“line number ” + i);
        if(i == example){
                System.out.print(“Yay!”);
        }
        System.out.print(“passed the if”);
}
```

Notice that once you go inside the “for” loop, all lines of code, started indented from the “for”. And inside the loop, any lines of code inside the “if” were also indented from the “if”. Why should I care? Because it makes reading the code SO MUCH EASIER, for you - when you come back to the code later (to figure out a mistake or to add to it) - and for others (like TA’s trying to help you, etc…) to read.

[4] J-Doc (Java Documentation)

When you write code, sometimes you want to give more information. That’s fine. There are 3 ways to write comments in your code that only humans read (but computers won’t).

(1) Write // at the end of a line. It means that anything to the right of it is a comment. Example: int cheese = new Cheese(“Yellow”); // creates a yellow cheese item

(2) You can make a block of comments, by doing this: /* */ or by doing this /** */. Anything written in the block will not be read by the compiler. Example

```
/**
 * Create a new cheese item to use for cheesecakes.
 */
Cheese cheese = new Cheese(“Yellow”); // creates a yellow cheese item
```

Whenever you do things that are somewhat complicated, you should add comments. If you’re not sure if you should or shouldn’t add a comment, add a comment - make your life easier for yourself.

There is more to discuss here, but this is the short of it.