Coding Style in EECE 1080/CS1

Spring 2016

Coding style is hotly debated in industry, but there is consensus that everyone should follow *some* type of coding style. While the location of curly braces, what a tab is (really a tab or spaces), if curly braces are always required, etc.. differ by company, we are going to follow the following guidelines.

Coding style is important for a number of reasons. Once in industry you will be reading other developers code and writing code other developers will have to read. To save you and the company time, it is important that the code is easy to follow. Lastly, the best reason for getting into the habit of following style is it will make you a better and more efficient programmer. Bugs will inevitably come up and tracking them down in your code as fast as possible will be easier if you (or your TA/instructor) can easily see what your code does.

* Each block should be indented with **2 spaces** from the previous block. A block is any sequence of lines between curly braces. Do not use tab characters.
* An if, else, else if, for loop, while loop, do-while loop, all should use curly braces, unless the statement is on the same line.
* All curly braces should be on the same line as a statement: if(a){ ... }else{ … this reduces the number of lines in your source file.
* The ending curly brace should line up with the starting character of the c++ statement that started it, such as if, for, while or }else{.
* **Lines should be 100 characters or fewer.**
* Keep a space between boolean operators. (a && b) is correct but (a&&b) is not. (a < 5) is correct but (a<5) is not.
* Variables names should describe what they are and have \_ between\_words.
* Function names should be camelCase.
* Class names should start with a Capital Letter.

**Example of a Well Styled Program**

// Author: Joe Student

// Filename: joes\_hw.cpp

// Description: Joe’s Homework Assignment

#include <iostream>

using namespace std;

int main(){

int x = 10;

while(x > 0){

cout << x << endl;

if(x < 5){

cout << x\*x\*x << endl;

} else {

cout << x\*x << endl;

}

}

return 0;

}