

## Report R2: Coding Standard - James Small

Purpose	Show Coding Standard for C++ Programs
Program Headers	Begin all files with a header
Header Format	<p>Header to contain the following, with each line starting on the left side of the file</p> <pre>// Name: James Small // Program: 2A // Class: CSE455 // Description: Main Program to count lines of code in a file</pre>
Method Headers	<p>Methods must contain a header in the format of a comment. The header will consist of one or more comment lines with a brief description of the method. See comments section on formatting rules</p> <pre>// This function calculates the LOC of a file</pre>
Identifiers	Use Descriptive names for identifiers. No abbreviations or single letter for variables, functions, or any other identifiers.
Identifiers Example	<pre>int a = 1           // this is bad int letterCount = 1 // this is good</pre>
Comments	Document code where needed using comments. Comments on their own line must begin with // and start on the far left side of the file. Comments on same line of code must start 2 spaces after the end of the line of code. Never use /* */ style of commenting
Blank Spaces	Program must be written so different logical sections of code must be separated by a blank line. If lines of code are related, then no blank lines between them.
Indenting	Indent every level of brace from the previous one. Indenting is done using 4 spaces at all times. See examples below for indenting for different types of commands.
Capitalization	Class names start with a capital. Functions, methods, and variables start with a lower case letter and follow standard camel case structure. Any define statements are all capitalized.
Brackets	<p>For class and method declarations, the opening and closing {} must be on their own lines. For all other brackets, the opening { will be on the same line as the command, while the closing bracket will be on its own line. See examples in the sections below. For all brackets used besides those on classes and methods, if the number of items inside the brackets is greater than 1 line, then go ahead and use the opening and closing brackets. If there is only one line of code that would exist inside the brackets, then no opening or closing brackets are used. The one line will be indented as usual on the next line below the command. Example:</p> <pre>while (getline(infile,currentString))     if ((currentString.find_first_not_of(' ') != string::npos))         if (!(currentString[0] == '/'))             count++;</pre> <p>The while and both if statements each contain one line below them, so no brackets are used.</p>

## Report R2: Coding Standard - James Small

Variables	variables of the same type can be declared on the same line  int count, score, points;  An initial value can be specified on the declaration of the variable.  int count = 0;
Class Interface	Begin class interface with header above. Then show #ifndef, #define, and end with #endif. Next will be all #include's. Separate public: private: sections. If no items exist for one of those sections, don't list the section with no items in it. If both sections exist, put a space between them.
Class Interface Example:	// Name: James Small // Program: 2A // Class: CSE455 // Description: Counter class Header File  #ifndef COUNTER_H #define COUNTER_H  #include <string>  using namespace std;  class Counter { public: Counter(string fileName); int calculateLOC();  private : string fileToRead; }; #endif
if, if/else, if/else if/else	examples of if constructs:  if example:  if (count == 5) { count += 2; result += count; }  if/else example:  if (count == 5) { count += 2; result += count;

## Report R2: Coding Standard - James Small

	<pre>    } else {         count += 3;         result += count;     }  if/else if/else example:  if (count == 5) {     count += 2;     result += count; } else if (count == 6) {     count += 3;     result += count; }</pre>
while Statement	<p>Example of While Statement</p> <pre>while (count != 5) {     count++;     result += count; }</pre>
do while statement	<p>Example of do while statement</p> <pre>do {     count++;     result += count; } while (count != 5);</pre>
for statement	<p>Example of for statement</p> <pre>for (int i = 0; i &lt; 5; i++) {     count += i;     result += count; }</pre>
switch statement	<p>example of switch statement</p> <pre>switch (grade) {     case 'A':         cout &lt;&lt; "Excellent!" &lt;&lt; endl;         break;     case 'B':     case 'C':         cout &lt;&lt; "Well done" &lt;&lt; endl;         break;     case 'D':         cout &lt;&lt; "You passed" &lt;&lt; endl;         break;     case 'F':         cout &lt;&lt; "Better try again" &lt;&lt; endl;         break;     default :</pre>

## **Report R2: Coding Standard - James Small**

	cout << "Invalid grade" << endl; }
	Switch statement must contain a default section at the bottom.