↑ 2.20 Debugging

Students:
Section 2.21 is a part of 1 assignment: **CSC108 CH02.11-2.24 P2B**

This assignment's due date has passed. Activity will still be recorded, but will not count towards this assignment (unless the due date is changed). See this article for more info.

Includes: Reading only
Due: 02/06/2025, 11:59 PM EST

2.21 Style guidelines

Each programming team, whether a company, open source project, or a classroom, may have ***style guidelines*** for writing code. Below are the style guidelines followed by most code in this material. That style is not necessarily better than any other style. The key is to be consistent in style so that code within a team is easily understandable and maintainable.

You may not have learned all of the constructs discussed below; you may wish to revisit this section after covering new constructs.

Table 2.21.1: Sample style guide.

Sample guidelines used in this material	Yes	No (for our sample style)
Whitespace		

Each statement usually appears on its own line.	`x = 25;`
	`y = x + 1;`
		`x = 25;  y = x + 1;  // No`
		`if (x == 5) { y = 14; }  // No`

A blank line can separate conceptually distinct groups of statements, but related statements usually have no blank lines between them.	`x = 25;`
	`y = x + 1;`
		`x = 25;`
		`// No`
		`y = x + 1;`

Most items are separated by one space (and not less or more). No space precedes an ending semicolon.	`C = 25;`
	`F = ((9 * C) / 5) + 32;`
	`F = F / 2;`
		`C=25;  // No`
		`F = ((9*C)/5) + 32;  // No`
		`F = F / 2 ;  // No`

Sub-statements are indented 3 spaces from parent statement. Tabs are not used as tabs may behave inconsistently if code is copied to different editors. (Auto-tabbing may need to be disabled in some source code editors).	`if (a < b) {`
	`  x = 25;`
	`  y = x + 1;`
	`}`
		`if (a < b) {`
		`  x = 25;  // No`
		`  y = x + 1;  // No`
		`}`
		`if (a < b) {`
		`  x = 25;  // No`
		`}`

Braces		

For branches, loops, functions, or classes, opening brace appears at end of the item's line. Closing brace appears under item's start.	`if (a < b) {`
	`  // Called K&R style`
	`}`
	`while (x < y) {`
	`  // K&R style`
	`}`
		`if (a < b)`
		`{`
		`  // Also popular, but we use K&R`
		`}`

For if-else, the else appears on its own line	`if (a < b) {`
	`  ...`
	`}`
	`else {`
	`  // Called Stroustrup style`
	`  // (modified K&R)`
	`}`
		`if (a < b) {`
		`  ...`
		`} else {`
		`  // Original K&R style`
		`}`

Braces always used even if only one sub-statement	`if (a < b) {`
	`  x = 25;`
	`}`
		`if (a < b)`
		`  x = 25;  // No, can lead to error later`

Naming		

Variable/parameter names are camelCase, starting with lowercase	`int numItems;`	`int NumItems;  // No`
		`int num_items;  // Common, but we don't use`

Variable/parameter names are descriptive, use at least two words (if possible, to reduce conflicts), and avoid abbreviations unless widely-known like "num". Single-letter variables are rare; exceptions for loop indices (i, j), or math items like point coordinates (x, y).	`int numBoxes;`
	`char userKey;`
		`int boxes;  // No`
		`int b;  // No`
		`char k;  // No`
		`char usrKey;  // No`

Constants use upper case and underscores (and at least two words)	`const int MAXIMUM_WEIGHT = 300;`	`const int MAXIMUMWEIGHT = 300;  // No`
		`const int maximumWeight = 300;  // No`
		`const int MAXIMUM = 300;  // No`

Variables usually declared early (not within code), and initialized where appropriate and practical.	`int i;`
	`char userKey = '-';`
		`int i;`
		`char userKey;`
		
		`userKey = 'c';`
		`int j;  // No`

Function names are CamelCase with uppercase first.	`PrintHello()`	`printHello()  // No` `print_hello()  // No`
Miscellaneous		
Lines of code are typically less than 100 characters wide.	Code is more easily readable when lines are kept short. One long line can usually be broken up into several smaller ones.	

Feedback?

K&R style for braces and indents is named after C language creators Kernighan and Ritchie. ***Stroustrup style*** for braces and indents is named after C++ language creator Bjarne Stroustrup. The above are merely example guidelines.

Exploring further:

- Google's C++ Style Guide

How was this section? | Provide section feedback

Activity summary for assignment: CSC108 CH02.11-2.24 P2B

147 / 147 points

Due: 02/06/2025, 11:59 PM EST

147 / 147 points submitted to LMS

This assignment's due date has passed. Activity will still be recorded, but will not count towards this assignment (unless the due date is changed). See this article for more info.

Completion details

↓ 2.22 Output formatting