Formatting | Coursera

coursera.org/learn/interacting-system-managing-memory/supplement/OT6Xd/formatting

Formatting

Formatting

The formatting of code can also have a large impact on its readability. Programmers largely agree on some basic rules. One common rule is that code should be indented to reflect the block structure—*i.e.*, in C, code inside more curly braces should be indented more than code inside fewer curly braces. Another is that one should generally place a newline at the end of a statement, so that consecutive statements appear on different lines.

However, programmers typically have much dissension about the specific details of code formatting. Probably the most contentious topic is the placement of curly braces. The code below shows four different brace-placement conventions: Java Style, 1TBS, Allman Style, and GNU Style. We note that there are other variations on many of these, typically involving whether the **else** "cuddles" the close curly brace of the corresponding **if** (that is, the **else** is on the same line as the closing brace), and/or whether curly braces are used around single-line blocks. There are of course, other styles not shown here as well.

```
40

}
```

We strongly prefer the so-called "Java" style (so named because Sun’s style guidelines used it, and thus much of the code for Java uses it) for a couple reasons. First, placing the open brace on the same line as the**if/else/for/**... that it belongs to helps to avoid errors where a programmer accidentally adds a line of code to the body, but mistakenly places it before the braces. To see this problem, consider the following code:

```
3

4

5

1

2

{
  someSecureActivity(action);
}
if (hasPrivileges (user))
  logActivity(user,action);  //accidentally added in wrong place
```

Here someone has added code—a call to a function to log some activity—intending to write it in the body of the **if**, but has separated the curly braces from the **if**. In this case, the **logActivity** call is the entirety of the body of the **if**, and the remaining code is legal, but is not conditionally protected by the **if**. That is, the code behaves as if it were:

```
if (hasPrivileges (user)) {
  logActivity(user,action);
}
someSecureActivity(action);
```

Such mistakes can contribute to serious errors in the software. In this hypothetical example, the mistake would almost certainly prove to be a security vulnerability, allowing any user to perform the privileged action (and to make matters worse, not logging the activity if the user does not have privileges to perform that action).

We prefer Java style over 1TBS primarily from a consistency perspective (the curly brace is *always* in the same place relative to the rest of the code). Perhaps most importantly, we recommend a brace style in which you always use curly braces, even when they can be omitted around single line blocks. Omitting them can make introducing errors easier.

Ultimately, what brace style you use is going to either be dictated by the style guidelines wherever you work, or left up to your personal choice.

Completed