Naming | Coursera

Naming

Naming

The names that you use for identifiers can contribute significantly to (or detract significantly from) the readability of your code. Name your variables, functions, and types to indicate what they mean and/or do. If you can tell what something does from reading its name, you do not have to work (as hard) to figure it out. Of course, at the same time, you should not name your variables in overly long ways that become cumbersome to type.

A good rule of thumb here is that the length of a variable's name should be proportional to the size of its scope and the complexity of its use. It is reasonable to name the counter variable of a for loop **i** because it has a relatively small scope (one loop) and a simple use (it just counts: you can tell that from reading the for loop where it is declared). Functions and types should therefore generally have relatively descriptive names. They (usually) exist at quite large scopes (all of the functions we have seen so far have had a scope of the entire program), and perform complex tasks.

Some people like *naming conventions*. We have seen a few naming conventions so far. One is placing a **_t** suffix on the type name (*e.g.*, **color_t**). Another is writing the names of constants in all capitals. Some programmers like the Hungarian notation scheme, where variable names are prefixed with a sequence of letters indicating their types (*e.g.*, **chInput** starts with a "ch" indicating its type is char, and while **iLength** starts with an "i" indicating its type is int).

Another set of conventions arise in how you "glue together" multiple words, as spaces are not allowed in variable names. The two major ways are to use underscores (_) wherever you would want spaces (*e.g.*, **num_letters_skipped**). The other is to capitalize the first letter of words other than the first (*e.g.*, **numLettersSkipped**)—this approach is called "inner caps" (also called "camel case"). Either of these methods is fine, though many programmers have a *strong* preference for one over the other. We note that "inner caps" can also be applied to names that start with a capital letter on the first word by convention —such as class names, which you will see if you take a subsequent class in C++.

$\checkmark$

Completed