**Chapter 3 Names That Can’t Be Misconstrued**

* analyzed your names by asking the question, “What other meanings could someone interpret from this name?
* that question will help you spot the misinterpretations of each name.
* example Filter();
* the problem with Filter is it is unclear whether it means “to pick out” or to get rid of. a better name is select to pick out and exclude to get rid of.
* the clearest way to name a limit is to put max\_ or min\_ in front of the thing being limited.
* CART\_TOO\_BIG\_LIMIT → MAX\_ITEMS\_IN\_CART
* for inclusive ranges, a good choice is first/last.
* 
* unlike stop, the word last is clearly inclusive.
* in addition to first/last, names min/max may also work for inclusive ranges assuming they “sound right” in that context.
* it’s more convenient to use inclusive/exclusive ranges: PrintEventsInRange(“date”, “date”);
* naming convention for inclusive.exclusive is begin/end
* adding words like is, has, can, or should can make booleans more clear.
* it’s best to avoid negated terms in a name.
* bool disable\_ssl = false; → bool use\_ssl = true;
* most programmers are used to the convention that methods starting with get are “lightweight accessors” that simply return an internal member.
* best names are ones that can’t be misconstrued-the person reading your code will understand it the way you meant it, and no other way.