

The Mañana Principle

Mañana is the Spanish word for *tomorrow*. In a paper titled *A Novice's Process of Object-Oriented Programming* presented at OOPSLA '06 in Portland, Oregon, USA, 22–26 October 2006, Michael E. Caspersen and Michael Kolling introduce the *Mañana Principle*.

Mañana Principle: When —during implementation of a method— you wish you had a certain support method, write your code as if you had it and implement the method later.

Thus, the Mañana Principle is to put something off “until tomorrow”.

If you can see the advantage of having a helper method, stub it in (write a specification and header for it, enough so it compiles), put off completing it until later, and continue your coding, writing calls on the helper method.

This principle is well known! All professional programmers use it. But many, if not most, students in their first or second semester of programming don't take the Principle seriously. They rarely write extra helper methods, and if they do, they rarely specify them.

Perhaps giving a name to the process —the *Mañana Principle*— will make people more aware of the advantage of writing helper methods with good specifications.

Caspersen and Kolling go on to describe special situations in which the Mañana Principle should be applied. We copy them below. Read through them, and become more aware as you program of the advantage of applying the Mañana Principle.

Nested Loop rule: If you have a nested loop, move the inner loop into a separate method.

Code Duplication rule: If you write the same code segment twice, move the segment into a separate method.

Hard Problem rule: If you need the answer to a problem that you cannot immediately solve, make it a separate method.

Heavy Functionality rule: If a sequence of statements or an expression becomes long or complicated, move some of it into a separate method.

Special Case rule: If you write code to treat a special case in your algorithm, treat the special case in a separate method.