

CSCI 233: Programming Notes 3

The Programming Process

Loosely, the way in which development of a program looks something like this:

1. Figure out *what* your program (module, function) should do.
2. Work out how to tell if your program works.
3. Work out *how* to do it, including both the *representation* of data and the *algorithm*.
4. Write the program.
5. Validate that it works correctly.
6. Polish the program:
 - Write more clearly.
 - Format more neatly.
 - Test better/more compactly.
7. Reflect: What have I learned?

This isn't a strictly sequential process. At any point, you may learn something that lets you do an earlier step better, in which case you spiral back through some of the earlier steps before moving on. Steps 1–3, in particular, often proceed together; the important thing to remember is to complete all three.

Working out the design and the program itself will often proceed by breaking the problem into smaller pieces. This process then applies (recursively!) to each piece. That is especially true for validation (testing): it is important to know that the pieces of a larger program work correctly.

Some of the steps are unavoidable: for example, you can't skip actually writing code. But the steps that seem skippable are especially important, even if they aren't the most exciting. The final step—reflection—is especially important for getting better at programming.

At this point in the class, the functions you are writing are pretty simple, which makes some of the steps trivial and other “optional” steps (such as testing) are bigger than the “real” work. But you will do well to stay close to this pattern, developing the habit of paying attention to all of these steps, because they will prove valuable when the problems you are solving are larger and more complicated.