

developer.*

A Web Magazine for Software Developers

Principled Programming

by Daniel Read

The Principle of Personal Character

Write your code so that it reflects, or rises above, the best parts of your personal character.

The Principle of Aesthetics

Strive for beauty and elegance in every aspect of your work.

The Principle of Clarity

Value clarity equally with correctness. Utilize the proven techniques that will produce clarity in your code. Correctness will likely follow suit.

The Principle of Layout

Use the visual layout of your code to communicate the structure of your code to human readers.

The Principle of Explicitness

Always favor the explicit over the implicit.

The Principle of Self-Documenting Code

The most reliable document of software is the code itself. In many cases, the code is the only documentation. Therefore, strive to make your code self-documenting, and where you can't, add comments.

The Principle of Comments

Comment in full sentences in order to summarize and communicate intent.

The Principle of Assumptions

Take reasonable steps to test, document, and otherwise draw attention to the assumptions made in every module and routine.

The Principle of User Interaction

Never make the user feel stupid.

The Principle of Going Back

The time to write good code is at the time you are writing it.

The Principle of Other People's Time and Money

A true professional does not waste the time and money of other people by delivering poor quality work.

Note: this version of Principled Programming is the “short form.” A longer version is also available which includes commentary and explanation with each of the principles.

###

Daniel Read is editor and publisher of the `developer.*` web magazine. He lives in Atlanta, GA, where he works as a software architect. He is currently at work on a book about software development crafted for a business audience.