

Clean Code: Writing Code for Humans by Cory House

Email: housecor@gmail.com | Blog: bitnative.com | Twitter: [@housecor](https://twitter.com/housecor)

Full 3 hour Pluralsight Course and Slides: pluralsight.com/author/cory-house

Please do me a favor and rate this talk! speakeerrate.com/talks/41881

Handy Jargon (Great for dinner parties):

Signal to Noise Ratio – The ratio of code that conveys intent to cruft.

Side Effects – When a method does more than what its name describes. Avoid by refactoring to separate methods or selecting a more descriptive name.

Magnet Class – Class with ambiguous/high level name. Attracts lazy developers and becomes a dumping ground for unrelated logic. Characterized by low cohesion.

Cohesion – A highly cohesive class has methods that revolve around a common purpose and frequently interact. A class with a clear responsibility should be highly cohesive.

The Outline Rule – Strive to structure your code like an outline with multiple levels of abstractions. This aids the reader by converting implicit concepts in named explicit concepts. It also aids navigation when looking for bugs.

Cyclomatic Complexity – A measurement of the number of discrete paths through the code. Higher complexity is associated with difficulty reading, and more frequent bugs.

Intermediate Variable – A variable that clarifies intent by providing a clear name to express the intent of a conditional. Useful technique to avoid writing a comment.

Zombie Code – Code that is commented out in production. Often code that was previously used in production or was never completed. This noise that hinders reading and refactoring. Move to a branch or delete.

Stringly Typed – Using strings in conditionals instead of strongly typed alternatives like Booleans, enums or classes. Strive to stay strongly typed when working in strongly typed languages so you can lean on the compiler and enjoy Intellisense support.

Primitive Obsession – Passing around a loose list of variables between methods instead of defining a class to encapsulate the concept.

DRY – Don't repeat yourself. Each piece of logic/knowledge should exist in one place.

Return Early – Return when there is nothing more to do in the method.

Fail Fast – Throw an exception/return as soon as something required is missing or something unexpected occurs. This avoids failing slow which often masks the root cause.

Fail slow – Swallowing an exception or continuing processing in a method when you already know that there's no way the logic can succeed. Strive to fail fast so the root cause of the failure is easy to detect and unpredictable states aren't created.

Table Driven Method – A method whose data and/or logic is driven via a data stored elsewhere (typically in a database or configuration file).

Mayfly Variables – Variables that are initialized as late as possible and removed from scope as early as possible. Like a Mayfly, they have a very short lifespan. NOTE: Avoid this technique in JavaScript due to hoisting.

Broken Window Theory – Don't accept ugly hacks in the code, despite the fact that there may already be flaws in the system. Derived from a study that found once a window is broken in an abandoned building it quickly falls into further disrepair due to neglect.

Boy Scout Rule – When adding a feature or fixing a bug, strive to leave the code a little better than you found it.

Is Your Team Writing Clean Code?

I offer on-site training sessions on Clean Code and Software Architecture. Session length varies from 1 hour to a full day.