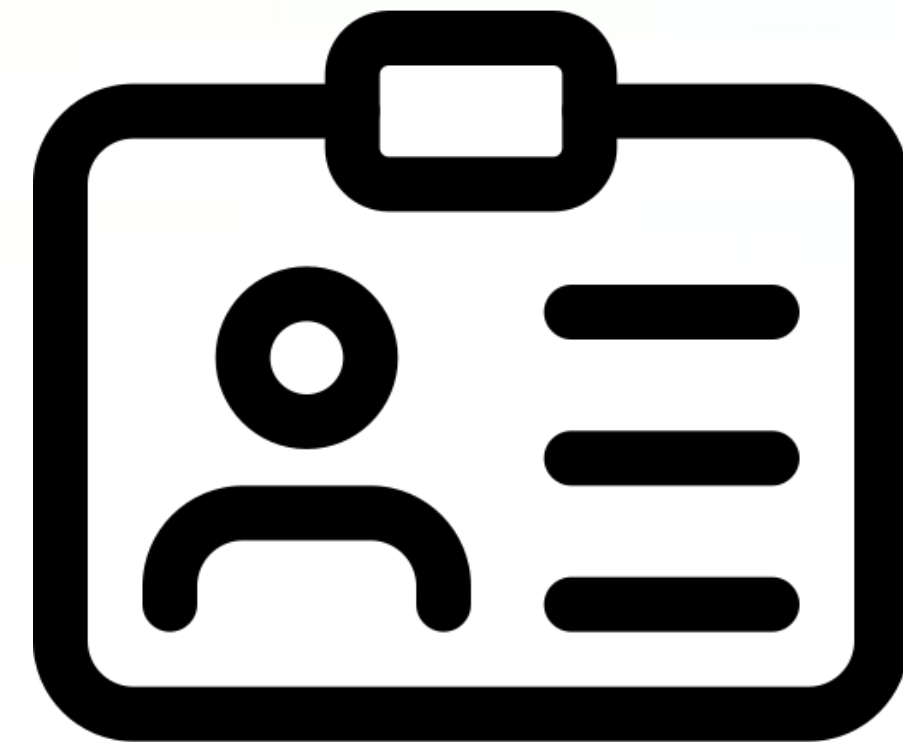


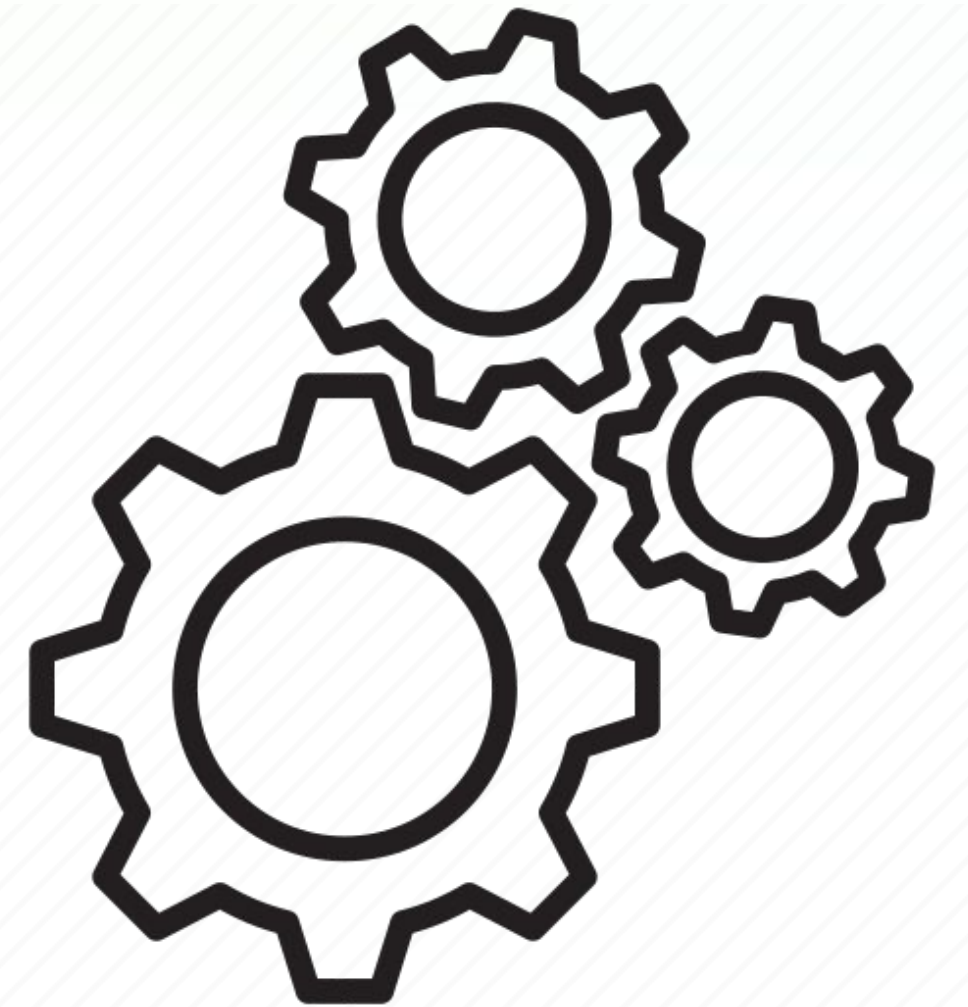
- Use Intention-Revealing Names
- Make Meaningful Distinction
- Use Pronounceable Names
- Use Searchable Names
- Magic Number
- Follow Convention
- Pick One Word Per Concept

Naming

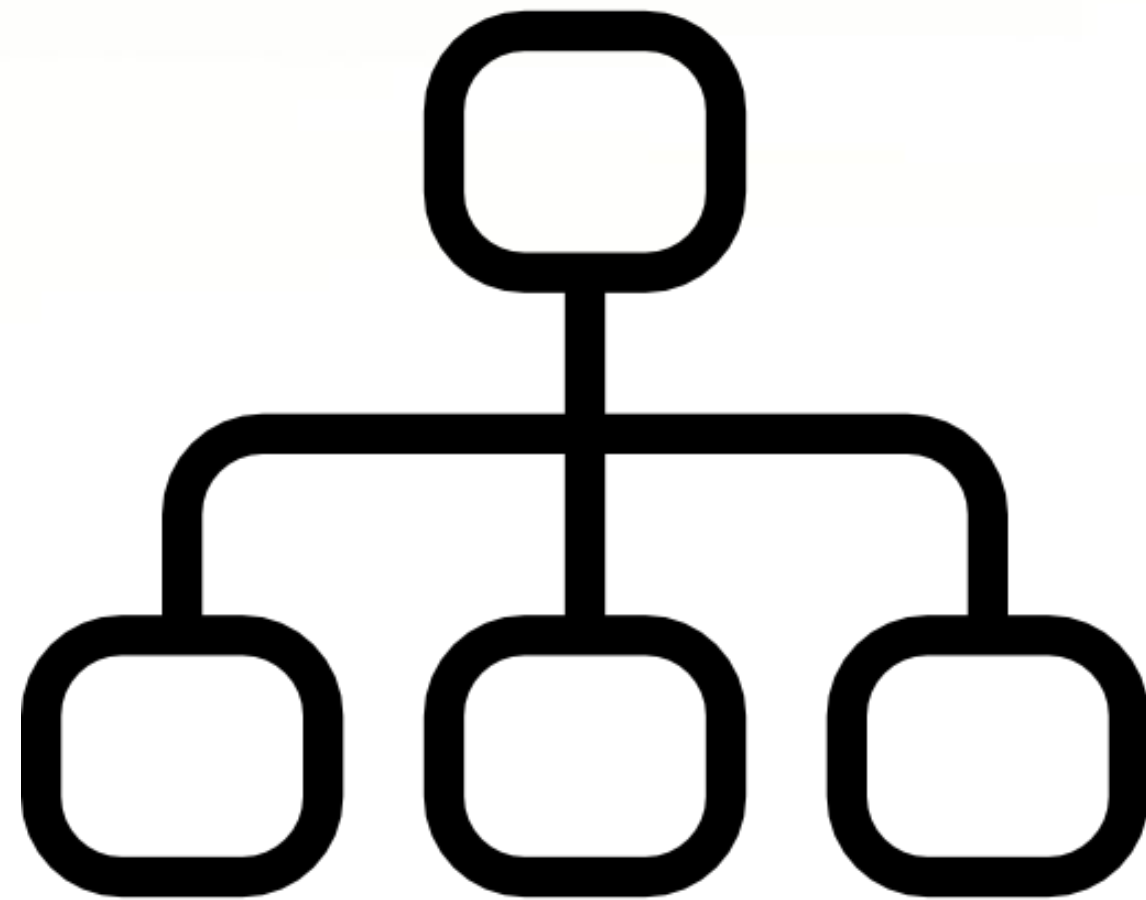


- Small
- Do one thing
- Have no side effects
- A function is independent of another function.
- Do something or answer something
- Don't repeat yourself
- Output argument
- Too many arguments.
- Don't use flag arguments.
- Condition to function
- One level of abstraction per function

Function



Source Code Structure



Formatting

- Choose one and follow it, be consistent
- Convention of framework, language, company
- Delete unused code

Organize

- Separate concepts vertically
- Related code should appear vertically dense
- Declare variables close to their usage
- Use white space to associate related things
and disassociate weakly related things
- Keep line short

Function

- Dependent functions should be close together
- Similar functions should be close together
- The Stepdown Rule: Reading code from top to bottom
Pay attention to the level of abstraction.
- Make sure the function is no longer than one page

Comments



“Don’t comment bad code – rewrite it.”

–Brain W.Kernoghan and P.J. Plaugher



Bad comments

- Don't be redundant
- Don't add obvious noise
- Don't comment out code, just remove it



Good comments

- Use as explanation of intent
- Use as clarification of code
- Use as warning of consequences

As a general rule, if you need to add comments, they should explain "why" you did something rather than "what" is happening.

Let's sum it up

- Read code!
- Read clean code
- Keep it simple \neq easy
- Refactor
- Be your own reviewer

