

Clean Code Fundamentals

Names++

Pre-work

- Video: <https://cleancoders.com/episode/clean-code-episode-2>
- Exam: <https://cleancoders.com/episode/clean-code-episode-2/exam>

Chapters

Chapter	Time
Reveal Your Intent	00:00:42
Describe the Problem	00:04:24
Avoid Disinformation	00:08:19
Pronouncable Names	00:13:24
Avoid Encodings	00:16:36
Parts of Speech	00:20:01
The Scope Length Rule	00:25:06
Recap & Conclusion	00:31:39
Distance to the Sun	00:36:07

Timetable

Activity	Time
Warmup	5 min
Exercise 1	10 min
Exercise 2	10 min
Intro to technical writing	45 min
Exercise 3	10 min
Wrap up	5 min

Warmup

- What was the most confusing name you ever encountered?
 - Type in the meeting chat

Exercise 1

- Prompt
 - Discuss a possible meaning of a variable given its name
- Time limit: 10 minutes

Possible answers

Name	Kind	Meaning
<code>is_true</code>	Function	Check if boolean statement evaluates to true
<code>important_details</code>	Variable	Details that are important
<code>dataType</code>	Variable	Contains type of dataset
<code>ProcessInfo</code>	Class	Structure with fields about a process
<code>count_if</code>	Function	Count elements that satisfy a condition
<code>ENXIO</code>	Constant	Error code for “no such device or address”
<code>active?</code>	Function	Check if a process is active
<code>theTable</code>	Variable	List of rows with column elements
<code>set</code>	Class	Collection of unique elements

Exercise 2

- Prompt
 - Fill-in the blanks for using different parts of speech in names
 - Parts of speech **reference**
- Time limit: 10 minutes

Possible answers

Entity	Part of speech	Example
Class	Noun	Solver, SupplyChain
Variable	Noun	transaction, table
Boolean variable	Predicate	is_active, is_valid
Function or method	Verb	open, fill_hierarchy
Function or method returning a boolean	Predicate	is_empty, is_open
Accessor (getter or setter)	Verb	get_name, set_name
Property (method pretending to be a field)	Noun	name, age
Boolean property	Predicate	is_empty, is_open
Enum (state or object descriptor)	Adjective	RED, SWITCHABLE

Putting parts of speech together

- Write in active voice
- Choose strong verbs
- Reduce the use of *there is* and *there are*
- Minimize certain adjectives
- Keep list items parallel

See [Google Technical Writing](#) for more details

Write in active voice

- Use the active voice most of the time
 - Most readers mentally translate passive voice to active voice
 - It is easier to read
 - It is more concise
- Active voice sentence structure
 - Active voice sentence = actor + verb + target
 - Passive voice sentence = target + verb + actor
 - Yoda speak = verb + target + actor
- Examples
 - Active voice
 - The user clicks the button.
 - The system sends the email.
 - Passive voice
 - The button is clicked by the user.
 - The email is sent by the system.
 - Yoda speak
 - Clicked the button is by the user.
 - Sent the email is by the system.

Choose strong verbs

- Reduce imprecise, weak, or generic verbs, such as the following
 - forms of be: is, are, was, were, be, being, been
 - occur, happen, take place, come about, come to pass

Weak verb

The exception occurs when dividing by zero.

This error message happens when. . .

We are very careful to ensure. . .

Strong verb

Dividing by zero raises the exception.

The system generates this error message when. . .

We carefully ensure. . .

Reduce the use of *there is* and *there are*

- Remove unnecessary references
 - If possible, delete *there is* and *there are*
 - Repair the sentence by moving the true subject and true verb
 - Replace *there is* and *there are* with a more specific noun or pronoun

Original

Revised

There is a problem with the system.

The system has a problem.

There are no errors in the system.

The system has no errors.

There is a function `distribute` that assigns keys.

The `distribute` function assigns keys.

Minimize certain adjectives

- Provide facts instead of loosely defined adjectives and adverbs
 - Adjectives and adverbs are often vague and imprecise
 - Can make the reader wonder what the writer really means
- Example
 - Original: Setting this flag makes the application run screamingly fast.
 - Revised: Setting this flag makes the application run 150% faster.

Write effective lists

- Keep list items parallel
 - All items in a list should “belong” together
 - Align all items using similar
 - grammar and parts of speech
 - logical category
 - capitalization
 - punctuation
- Example of an ineffective list
 - Parts of speech – choose well;
 - logical category is essential,
 - capitalization and punctuation

Exercise 3

- Improve the following sentences
 1. The moth was removed by Grace Hopper.
 2. Performance optimization is overridden by the `--noperf` flag
 3. The measurements were taken by the static `Timer()` method until it was discovered by the QA team the bugs were present in the `Timer()` method.
- Time limit: 10 minutes

Possible answers

1. Grace Hopper removed the moth.
2. The `--noperf` flag overrides performance optimization.
3. The QA team discovered that the `Timer()` method contained bugs. The static `Timer()` method provided measurements until the bugs were discovered.

Wrap up

- Choose your names thoughtfully
- Communicate your intent
- Avoid disinformation
- Use pronounceable names
- Avoid encodings
- Choose parts of speech well
- Use The Scope Rule

What is next?

- Expect an e-mail with instructions for upcoming coding dojo

Final words

Always leave the code better than you found it.
– *The Software Craftsmanship Rule*