

Keep Comments Compact

The provided example emphasizes the importance of concise comments in code. Instead of using three lines to explain a C++ type definition, the comment condenses the information into one line, improving readability and efficiency. By succinctly describing the mapping of `CategoryType` to a pair of score and weight, the comment demonstrates the power of brevity in conveying essential information within code documentation.

Avoid Ambiguous Pronouns

The main idea is that using unclear pronouns in writing, especially in technical stuff like code comments, can make things confusing. Words like "it" or "this" might not make clear what they're talking about, so readers have to guess. To avoid this, it's best to either say exactly what the pronoun means or rewrite the sentence to be clearer. This shows how important it is to be really clear when explaining technical things.

Polish Sloppy Sentences

The main idea is that when you want comments to be easier to understand, you usually need to make them shorter. By looking at two versions of a comment from a web crawler, we can see that using simpler language not only makes it clearer but also helps to get the important information across faster. In this example, the updated comment is not only shorter but also makes it clear that uncrawled URLs are a top priority, which wasn't clear in the original version.

Describe Function Behavior Precisely

The code and comments show us how to count lines in a file, even in tricky situations like empty files or using different types of new lines. It's a reminder of why it's important to explain functions well in our code. The updated comment makes it easier to understand how the code works and what to expect when using it. This shows how crucial clear comments are for making code easier to read and maintain, which helps with future updates and fixes.

Use Input/Output Examples That Illustrate Corner Cases

This paragraph stresses how important it is to use clear examples in comments when writing code. Instead of just describing things in words, giving specific examples helps people understand how functions work and clears up any confusion. Showing different situations and tricky cases helps developers explain exactly what their code is supposed to do, making it easier for others to understand and work with in the future.

State the Intent of Your Code

This piece shows why comments in code are crucial. It says they should explain what the programmer wants to achieve, not just what the code does. It gives an example of a good comment that talks about why the code exists. Also, it points out a mistake in the code, proving

how clear comments can help catch errors. Even though tests are great for double-checking, good comments still help to understand what the program is supposed to do.

“Named Function Parameter” Comments

This piece talks about how to make function calls clearer with comments right inside the code. It's useful in languages like C++ and Java where you can't name parameters. By adding comments like `/* timeout_ms = */` and `/* use_encryption = */`, it makes things clearer, especially for true/false choices. They say it's best to put the comment before the value to avoid confusion. This helps explain tricky function stuff in a simple way.

Use Information-Dense Words

This piece emphasizes how it's helpful for programmers to know common coding patterns and use short, clear terms to talk about them. Instead of explaining a lot about what a part of the code does, they can use simple phrases that everyone understands. Like calling something a "caching layer to the database" instead of going into detail. Or saying "canonicalizing the street address" to mean cleaning up and standardizing an address format without needing to explain it all. This way of talking makes it easier to understand and read the code.