

Comments on Comments

Comments

- Comments can turn into liabilities.

Oh those comments

- Example — `//todo: ...`
 - IDE Vomits → IDE automatically makes these comments.
- Example — `i++; // increment`
- Example — `Car() // constructor`

Problems with comments

- Subconsciously, you are reading the comments — so having redundant comments is not good.
- Sometimes the programmers put their emotions/frustrations in the comments.
- Sometimes the comments don't represent what the code is doing. The code may have been updated since.

Comments vs. Readability

- Don't comment to make the code readable.
- A good code is like a good joke. You lay it down and walk away.
 - A good joke doesn't require any explanation.
 - A good code doesn't require any descriptions — it should be very self-readable.

When to comment?

- **Comment *why* and not *what* the code is doing.**
- For special cases or gnarly situations/conditions.
- To warn the programmer about certain peculiar conditions. — to describe why you had to implement things a certain way.
- Comment when comment can boost the understandability of the code, not to improve the readability.

What to comment?

- Comment the purpose of the code.
 - Why does this method exist?
 - What are the parameters in terms of the preconditions?
 - What are the expectations on the parameters? Is the input supposed to be a positive number?
- Another way to comment is automated tests.