# **97 Things Every Programmer Should Know**

## **Chapter 16: A Comment on Comments**

1. **Before -** We are always reminded by our professor to always put a comment in every program code that we have.

**After -** It is true that comments are not evil. They serve as a documentation of the code's logic and intent, providing valuable information for both the original programmer and other developers who may be working on the same code.

1. **Before** - Do put comments as much as possible.

**After** - Do put comments when needed. Having too many comments in the code, or comments that provide little to no information, can actually harm the code's readability and be difficult to maintain. With this case, comments can become a source of confusion and make it harder for programmers to understand the code's logic and intent.

1. **Before** - Other programmers do not typically review or check the comments.

**After** - While some programmers may not review or check comments in the code, as they might focus on the actual code implementation instead. However, there are still some programmers who do consider comments to be a valuable tool in understanding the logic and intent of the code. Reviewing comments can help programmers better understand the code, especially when it is a complex code written by different programmer. With comment, the programmer can have a clear picture about the code's intent, which makes it easier to enhance or improve.

## **Chapter 17: Comment Only What the Code Cannot Say**

1. **Before -** Do put comments as much as possible.

**After -** Yes, it is true, put comments as much as possible. But remember that there are good comments, and bad comments. Some comments can be well-formed and helpful, while others can actually detract from the code's readability and maintainability. If the comments are ill-formed, then it is not working on its purpose, and can become a source of confusion that leads to errors and mistakes.

1. **Before** - Comments are important as they clarify the code’s logic and intent.

**After** - But there are some comments that do not have a value which leads to a noise comments. They are present without serving any purpose and are unrelated to the code. These comments can create distractions and misinformation about the code. With noise or bad comments, this encourage programmers to ignore or skip all comments.

1. **Before** - Comments should provide information that cannot be conveyed by the code alone. They should clarify the reasoning behind the code and supplement the code's functionality.

**After** - When writing comments, treat them as if they were code. They should have a clear intent, be concise, and readable, making it easier to understand and maintain the code. The comments should explain the logic or intent behind the code, explaining what the code cannot say, not simply repeating what it does say.