**LOUIE JAY A. CENIZA APRIL 9, 2024**

**KNOWING WHAT TO COMMENT (CHAPTER 5)**

**What NOT to Comment.** Avoid adding comments to code simply to meet a commenting quota or policy. Comments should serve a meaningful purpose, such as explaining complex logic or providing crucial context that cannot be inferred from the code alone. Similarly, refrain from commenting on poorly chosen variable or function names. Instead, prioritize fixing the names themselves to be clear, descriptive, and self-explanatory.

**Recording Your Thoughts.** These comments offer valuable context to other developers and yourself, helping to understand the rationale behind the code. Furthermore, don't hesitate to comment on the flaws or limitations present in your code. Acknowledging areas for improvement can guide future enhancements and prevent misunderstandings or unintended consequences.

**Put Yourself in the Reader’s Shoes.** When writing comments, strive to anticipate the needs and perspectives of other developers who will read your code.

**Final Thoughts—Getting Over Writer’s Block.** Emphasizes overcoming the reluctance to write comments by encouraging developers to start with whatever comes to mind, even if it feels incomplete or vague at first. By simply getting into the habit of commenting thoughts as they arise, developers can gradually refine and improve their comments over time.