**LOUIE JAY A. CENIZA APRIL 11, 2024**

**MAKING CONTROL FLOW EASY TO READ** **(CHAPTER 7)**

**The Order of Arguments in Conditionals.** The general rule for determining the order of arguments in conditionals is to prioritize readability and clarity.

**The Order of if/else Blocks.** in this lesson it talks about the importance of thoughtful organization in code, particularly when it comes to the order of if/else blocks.

**Avoid do/while Loops.** It advises us against using do/while loops due to their potential for confusing code structure and readability issues. Unlike other loop constructs like while and for, where the condition appears before the block of code, in a do/while loop, the condition is evaluated after the block of code has executed at least once.

**Returning Early from a Function.** It advocates for the use of early returns in functions when it makes sense for code clarity and readability. It challenges the idea that functions should always have a single exit point, arguing that using multiple return statements, especially for handling special cases or edge conditions at the beginning of a function, can result in more natural and easier-to-understand code.

**Minimize Nesting.** Deeply nested code is hard to understand. Each level of nesting pushes an extra condition onto the reader’s “mental stack.”

**Can You Follow the Flow of Execution?.** This chapter has been about low-level control flow: how to make loops, conditionals, and other jumps easy to read. But you should also think about the “flow” of your program at a high level.