**LOUIE JAY A. CENIZA APRIL 16, 2024**

**EXTRACTING UNRELATED SUBPROBLEMS (CHAPTER 10)**

**Pure Utility Code.** The importance of creating utility functions for common tasks that are not provided by the built-in libraries of a programming language.

**Other General-Purpose Code.** This section discusses the practice of extracting unrelated subproblems into separate functions to improve code organization and maintainability. It provides an example where code for pretty-printing a dictionary is embedded within a larger function responsible for handling AJAX responses.

**Create a Lot of General-Purpose Code.** Creating a lot of general-purpose code is a highly beneficial practice in software development. It involves identifying common functionalities or patterns in your projects and abstracting them into reusable components or libraries.

**Project-Specific Functionality.** Extracting project-specific functionality into separate, reusable components can still greatly improve code readability and maintainability. Even if the extracted subproblems are not entirely project-agnostic, breaking them off into standalone functions can simplify the main codebase and make it easier to understand.

**Simplifying an Existing Interface.** Creating wrapper functions to simplify an existing interface is a powerful technique to abstract away complexity and improve code readability, encapsulating the complexity of the underlying interface, you make your code cleaner, easier to understand, and more maintainable. This approach allows you to achieve a clean and elegant interface even when working with less-than-ideal underlying implementations.

**Reshaping an Interface to Your Needs.** Reshaping an interface to suit your needs can greatly simplify your code and make it more focused on the actual logic of your program.

**Taking Things Too Far.** It's valuable to break down complex tasks into smaller, more manageable parts, there's a risk of overcomplicating things by breaking them down too much. When you create many small functions, each serving a very specific purpose, it can make the code harder to follow because readers have to keep track of the flow of execution between all these tiny functions.