DWA_07.4 Knowledge Check_DWA7

---

1. Which were the three best abstractions, and why?

1. Personally, simple functions and methods used to encapsulate code to be used.

2. Object-Oriented Programming (OOP) organizes data and its functions together for clearer complex system design.

3. Data manipulation using array, files and sortings to make large amount of information manageable for the user, coder, and computer

---

2. Which were the three worst abstractions, and why?

1. Global Variables global variables make code confusing. Changes in one place affect others unexpectedly, causing errors.

2. Spaghetti Code:
Using operations in one place then going somewhere else doing it again then needing to go to the top. The code Hard to follow, fix, or make changes, causing confusion and errors.

3. Premature optimization means making code too complex for speed before knowing where it's needed, hurting flexibility and clarity.

---

3. How can The three worst abstractions be improved via SOLID principles.

Global Variables:
Single Responsibility Principle: Arrange code so that each part does just one job, putting global variables into specific modules or classes.
Dependency Injection: Instead of using global variables, provide what's needed directly to each part. This makes things clearer.

Spaghetti Code:
Single Responsibility Principle: Split big tasks into small, clear tasks. Each task should have its own simple function.
Open/Closed Principle: Make code that can grow by adding new things, not changing what's already there.

Premature Optimization:
Single Responsibility Principle: Stick to the main job for each function. Don't mix speed concerns with the main work.
Liskov Substitution Principle: Focus on making code work first. Only after that, think about making it faster, without making things too complicated.