

DWA_07.4 Knowledge Check_DWA7

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

1. Object-Oriented Programming (OOP): Organizes code into reusable objects, enhancing maintainability and productivity.
2. Virtualization: Abstracts physical resources, enabling multiple operating systems and applications on a single machine, leading to cost savings and flexibility.
3. Database Abstraction Layers: Simplify database interactions, shielding developers from complexities and allowing for portability and easy maintenance.

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

Here are three abstractions that can cause difficulties in simple terms:

1. Global State: Storing shared data in a global scope can make code harder to understand and maintain.
2. Over-Engineering: Adding unnecessary complexity can lead to bigger and slower codebases.
3. Premature Optimization: Optimizing code for speed too early can make it more confusing and harder to work with without noticeable benefits.

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

Single Responsibility Principle (SRP): Encourages breaking down responsibilities into separate classes, reducing the likelihood of global state.

2. Open-Closed Principle (OCP): Designing classes to be easily extended prevents over-engineering and reduces the need for modifications.

3. Liskov Substitution Principle (LSP): Ensuring objects can be substituted seamlessly reduces the impact of premature optimization and maintains code flexibility.
