DWA_07.4 Knowledge Check_DWA7

---

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

Data separation and encapsulation
This abstraction involves separating data from the method that operates on it .This approach improves maintainability and readability .

Modularization and Reusability
This reduces redundancy ,easier maintenance and quicker development cycles .

Dependency injection
This allows you to abstract away the concrete implementation details of components.

---

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

Bloated Utility class
This mixture of responsibilities makes the class hard to understand .

Monolithic user influence component this results in a class that violates the single responsibility principle and becomes difficult to manage.

Overly Abstracted Database Access
This overcomplication can be avoided by using a more straightforward approach for database access without excessive obstruction layers.

---

3. How can The three worst abstractions be improved via SOLID principles.
By splitting responsibilities into separate classes or functions adhering to single responsibilities principle.

Split the responsibilities into separate classes or components following the single responsibility principle.

Simplify the database access without excessive abstraction and allow the user class to directly interact with the database .
