abstraction not concreteness?

It means we should write our code to depend on an interface or abstract concept, not on the concrete class itself.

For example, if we have IShape interface, we use it in the code, not Circle or Square directly.

This way, if tomorrow we add a new shape, we don’t need to change the main code.

In short: depend on what it does, not how it does it.

Q3: What is abstraction as a guideline and how we can implement this through what we have studied?

Abstraction is about focusing on the essential behavior and hiding the details.

As a guideline, it tells us: “don’t make your code tied to details, make it talk with high-level contracts.”

We implement this by:

Using abstract classes or interfaces.

Making the code work with the abstraction (like IShape) instead of the actual implementation (like Circle).

This gives flexibility, reusability, and easier maintenance.