1-Coding against interface not class

- **Means depend on a contract (interface/abstract type) instead of a specific class → increases flexibility.**
- **Coding against abstraction not concreteness is the same principle → rely on abstract definitions, not fixed implementations.**

2-Abstraction as a guideline

- **Focus on *what* an object does, not *how* it does it.**
- **Implemented through interfaces, abstract classes, and encapsulation (hide details, show only essentials).**