Introduction

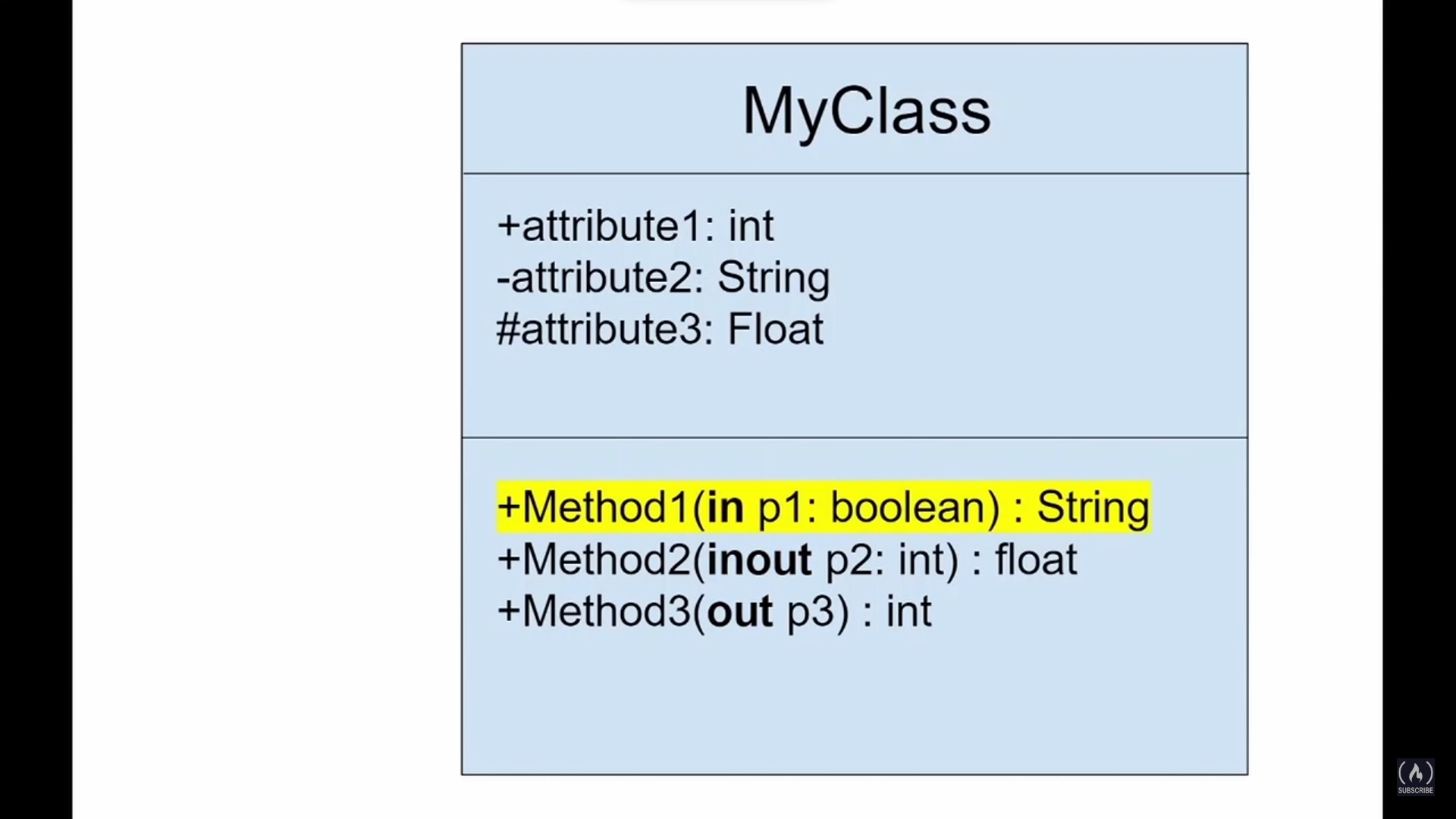
UML – Unified Modeling Language

Types of the diagrams

1. Class Diagram
2. Component Diagram
3. Deployment Diagram
4. Object Diagram
5. Package Diagram
6. Composite Structure Diagram
7. Profile Diagram
8. Use Case Diagram
9. Activity Diagram
10. State Machine
11. Sequence Diagram
12. Communications Diagram
13. Interaction Overview Diagram

Class Diagrams

1. **Structure:**



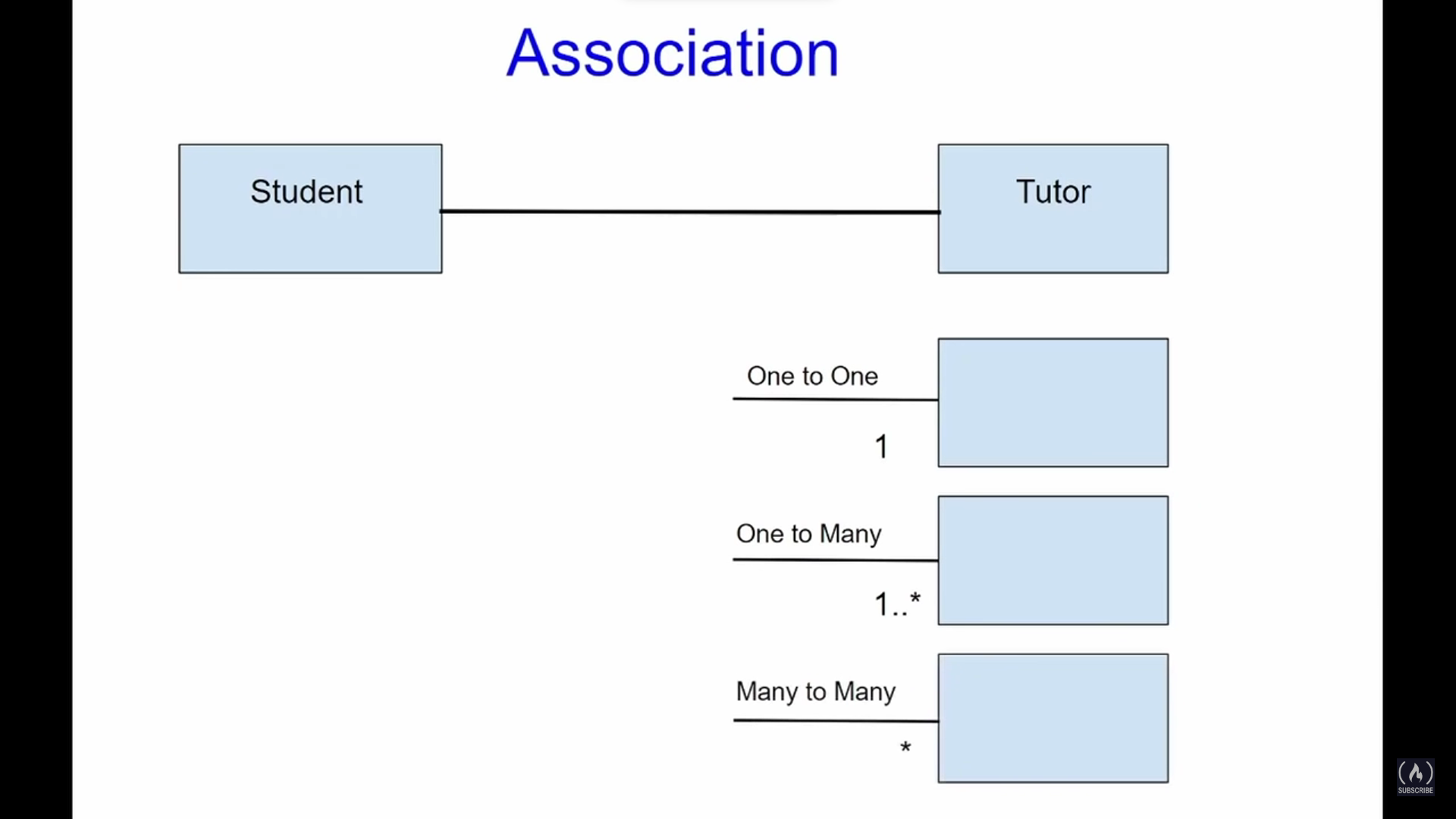
1. **Access modifiers:**

* “+” – public
* “-” – private
* “#” – protected
* “~” – local package

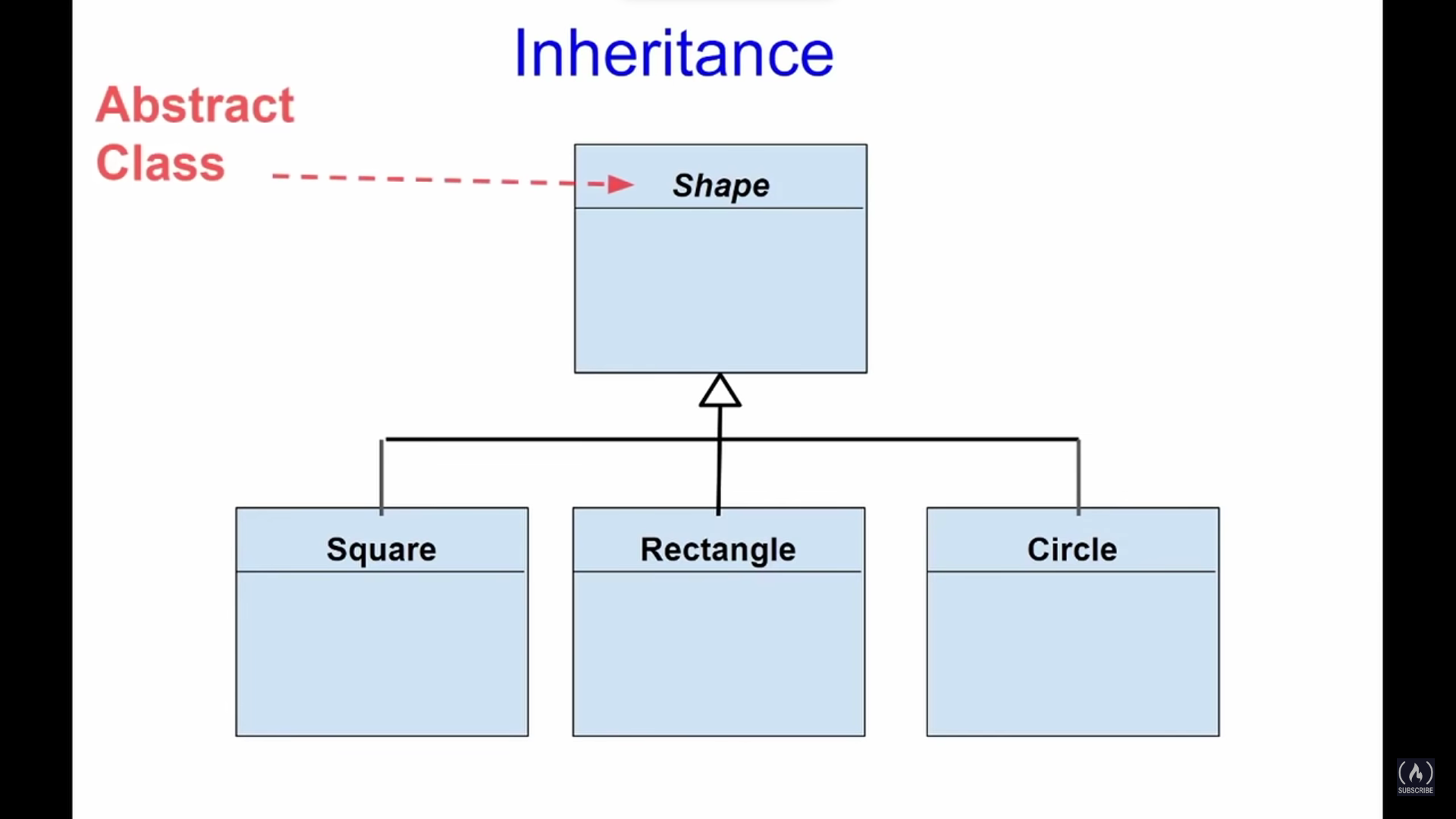
1. **Direction in the respect of the caller:**

* In – used in Class without changes
* Inout – used in class and class can change it
* Out – used as a container for the return value

1. **Perspectives (or how to use):**
2. **Conceptual –** represents the concepts in the domain. Consider language independent.
3. **Specification** – Describing abstractions of software or components with some specifications and interfaces but without any reference to a specific implementation.
4. **Implementation** – software implementation in a particular technology or a programing language.
5. **Relationships between classes**
6. **Association** – between same-leveled objects



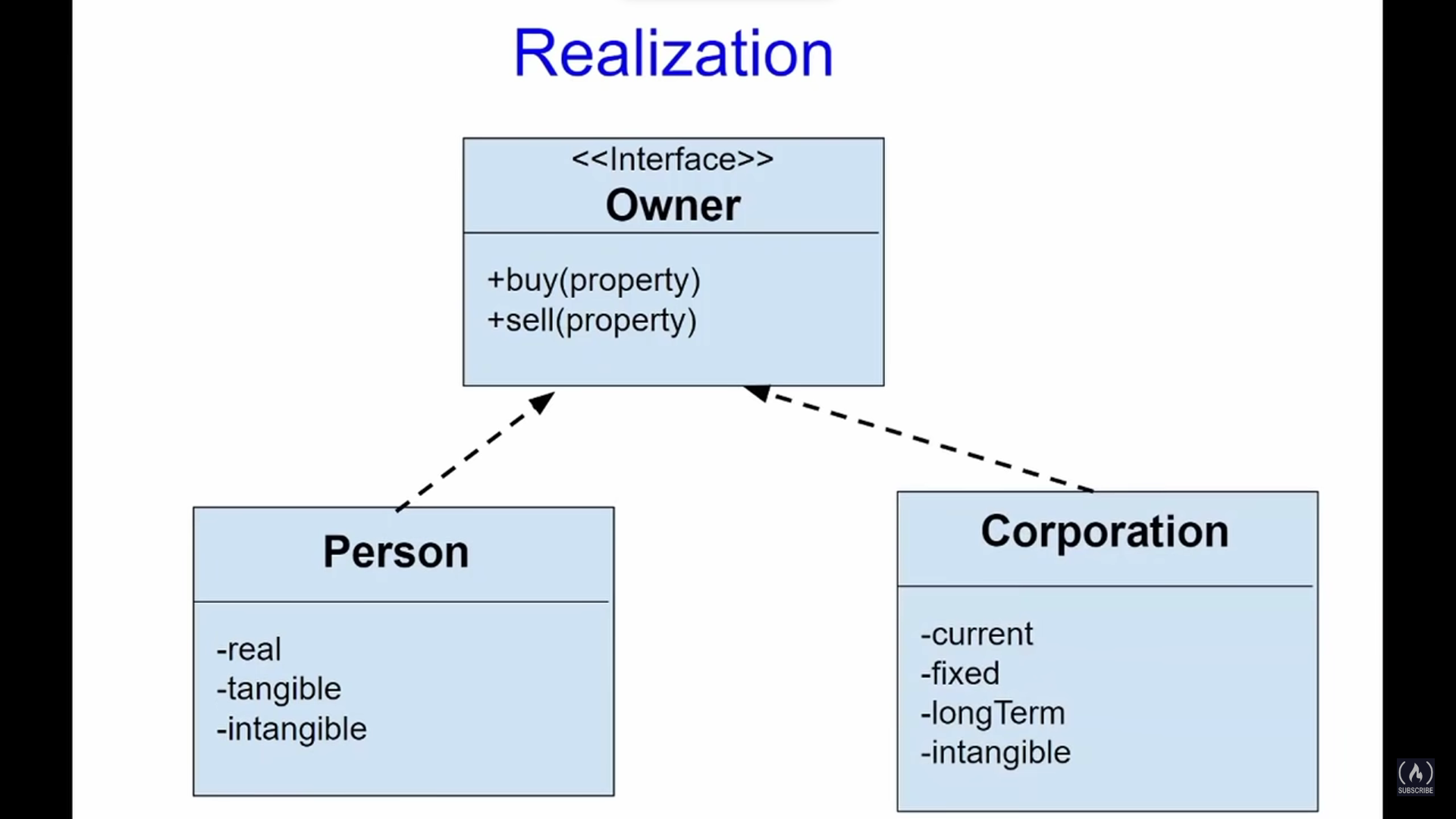
1. **Inheritance** – relations between parents and their descendants.

****

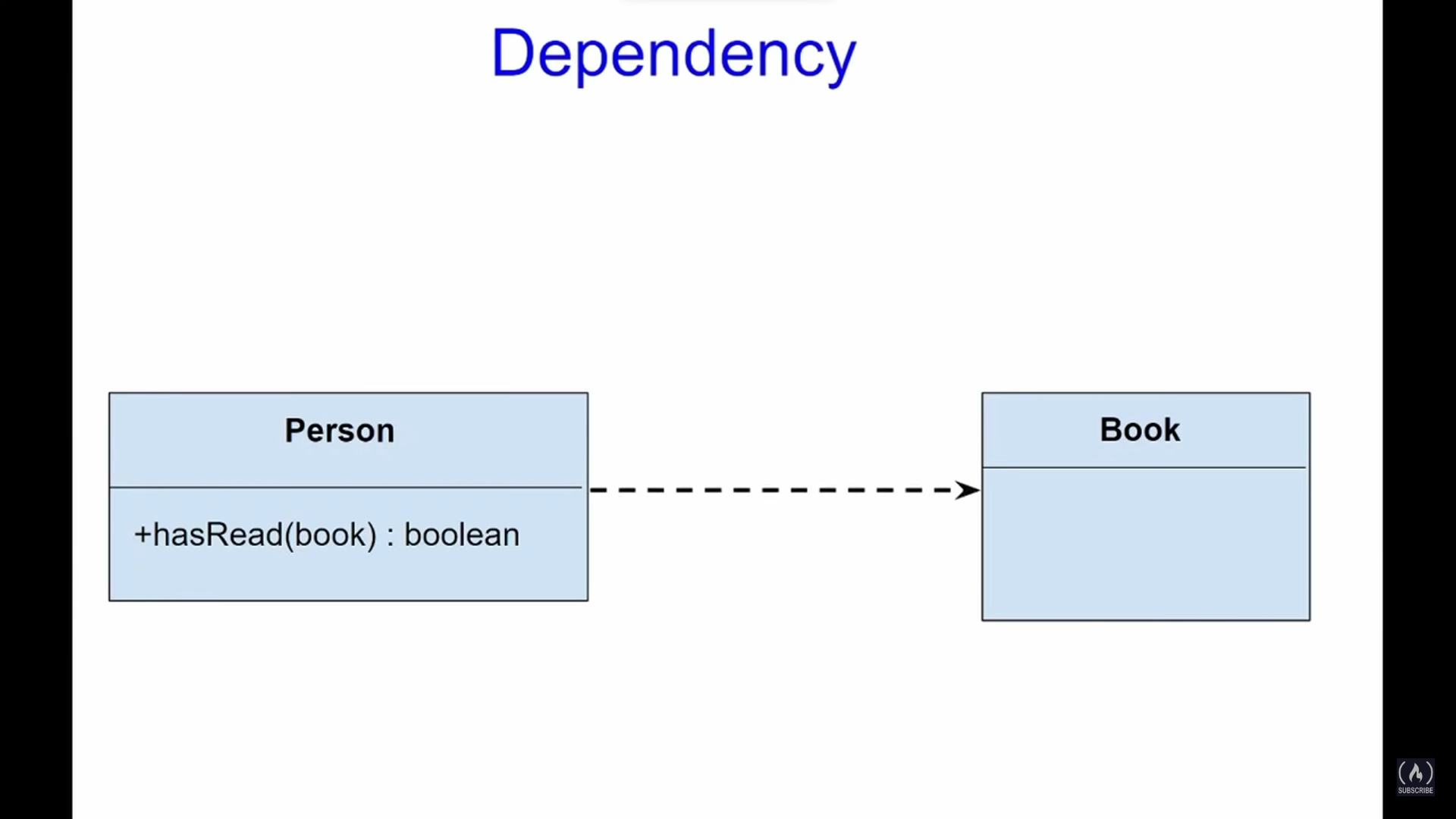
if

Italic font

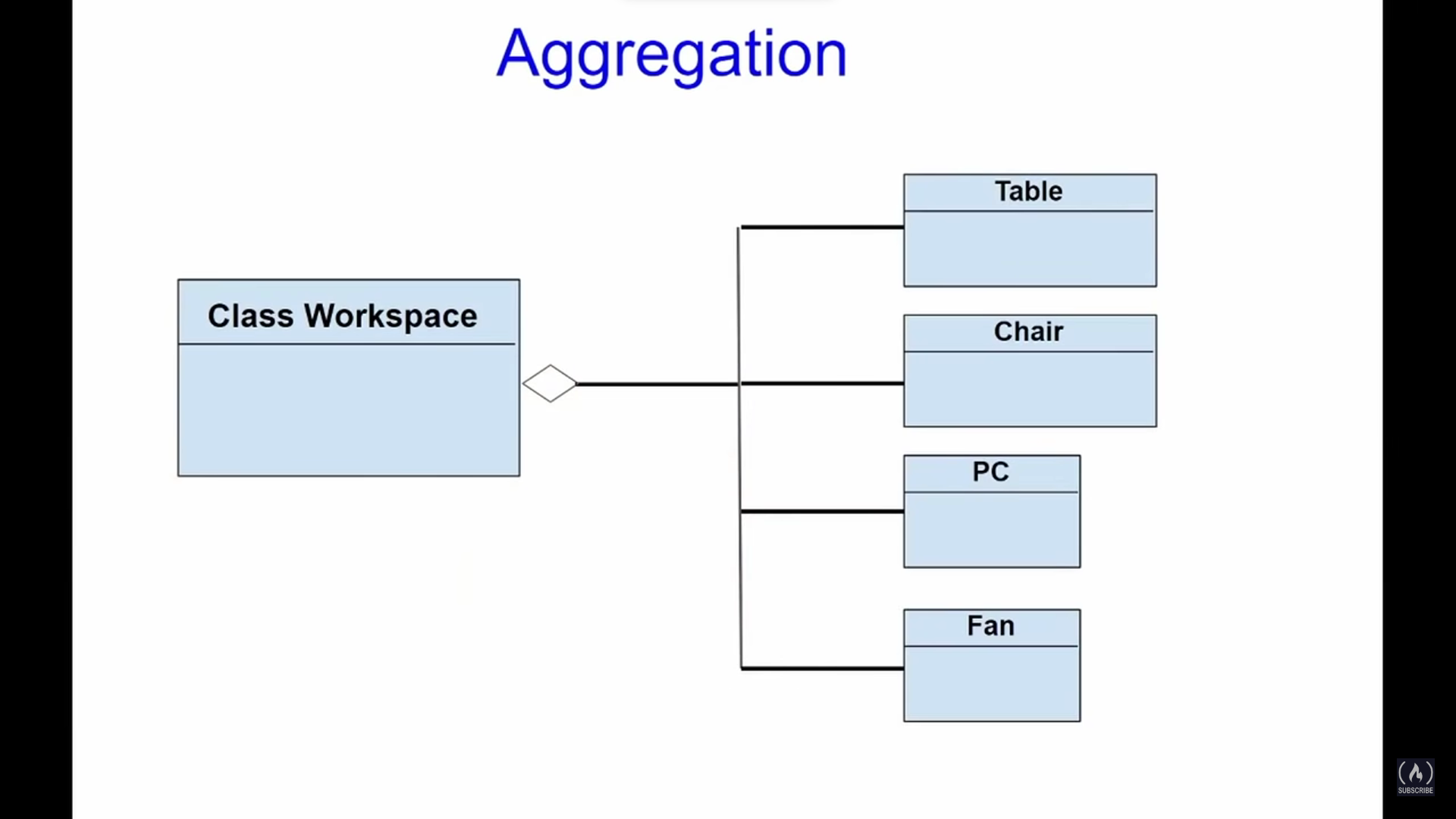
1. **Realization** – relationships between interface and objects which implements this interface.



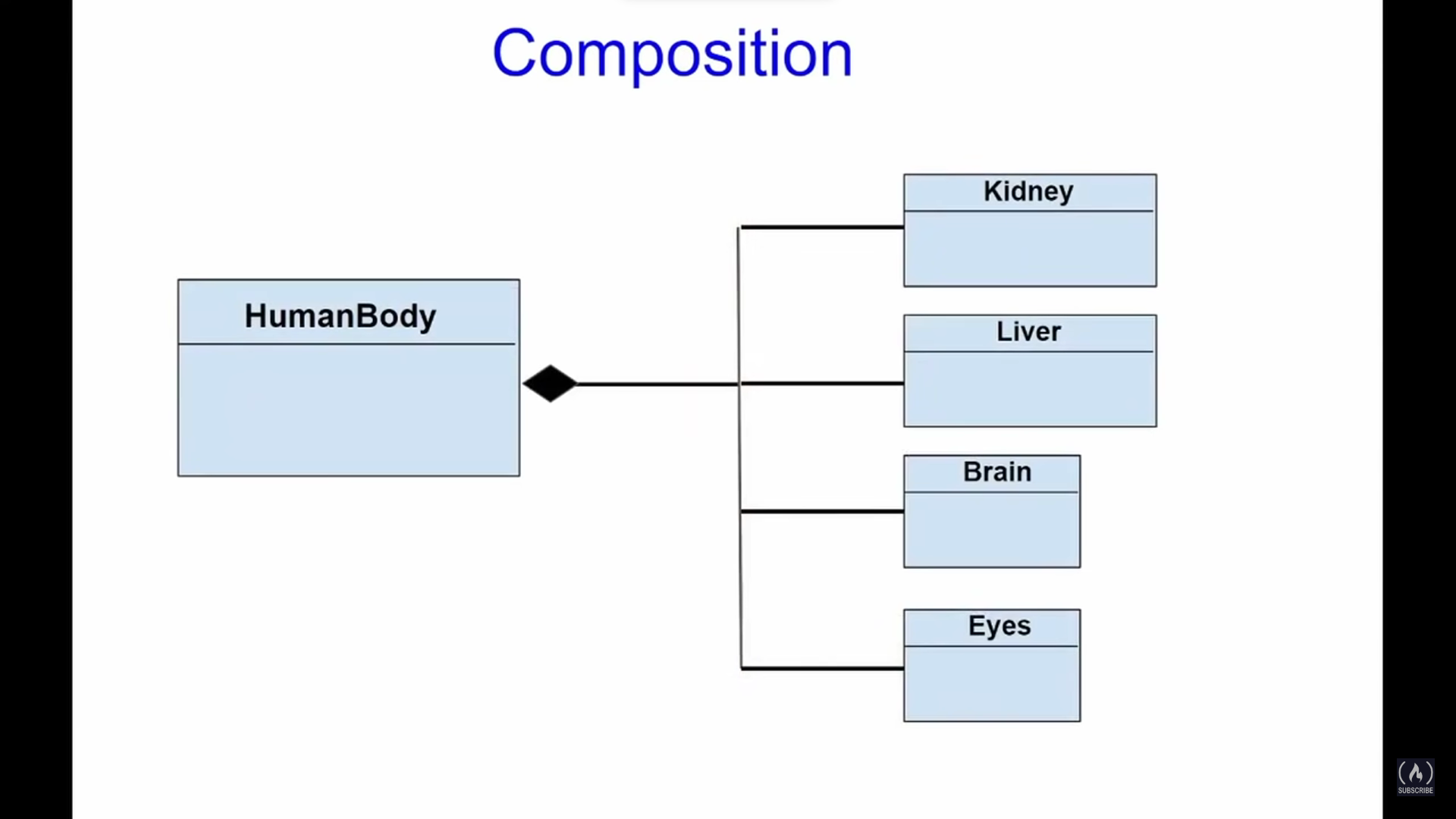
1. **Dependency** – when object of one class uses object of another class, and this object is not stored in any field.



1. **Aggregation** – when classes that form aggregator class can live by their own without aggregator (main) class.

****

1. **Composition** – when classes that form aggregator class CAN NOT live by their own without aggregator (main) class (are destroyed with it).

****

1. **Dgd**