Part 02

**1-What is the difference between class and struct in C#?**

**Classes**:

* **Reference Types**: Stored on the heap.
* **Inheritance**: Supports inheritance.
* **Garbage Collection**: Managed by the garbage collector.
* **Nullability**: Can be null.

**Structs**:

* **Value Types**: Stored on the stack.
* **No Inheritance**: Cannot inherit from other structs (except interfaces).
* **No Garbage Collection**: Not managed by the garbage collector.
* **Immutability**: Often used for small, immutable data.

**2-If inheritance is relation between classes clarify other relations between classes**

**1. Inheritance (is-a Relationship):**

* Indicates that one class derives from another.
* Example: class Car : Vehicle → "A car *is a* vehicle."

**2. Association (uses Relationship):**

* A relationship where one class uses or depends on another, without ownership.
* Example: A Customer class might use an Order class to place orders.

**3. Aggregation (has-a Relationship):**

* A stronger form of association where one class owns or contains another.
* Example: A University class *has a* collection of Department classes, but departments can exist independently.

**4. Composition (part-of Relationship):**

* A specialized form of aggregation where the lifecycle of the contained object depends on the container.
* Example: A Car class might be composed of Engine and Wheel classes. If the Car is destroyed, the Engine and Wheels are destroyed as well.

**5. Dependency (depends on Relationship):**

* A temporary relationship where one class relies on another to perform its operation.
* Example: A ReportGenerator class may depend on a DataSource class.