Association, Aggregation & Composition

Association is a relationship where all objects have their own lifecycle and there is no owner. Let's take an example of Teacher and Student. Multiple students can associate with single teacher and single student can associate with multiple teachers, but there is no ownership between the objects and both have their own lifecycle. Both can create and delete independently.

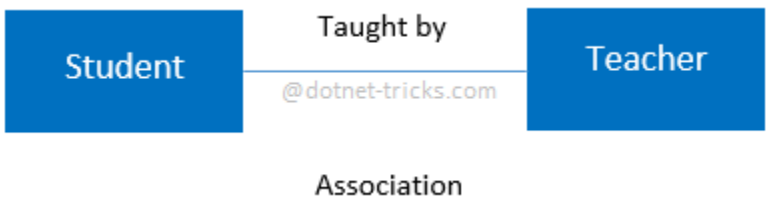
Aggregation is a specialized form of Association where all objects have their own lifecycle, but there is ownership and child objects can not belong to another parent object. Let's take an example of Department and teacher. A single teacher cannot belong to multiple departments, but if we delete the department teacher object will *not* be destroyed. We can think about it as a "has-a" relationship.

Composition is again specialized form of Aggregation and we can call this as a "death" relationship. It is a strong type of Aggregation. Child object does not have its lifecycle and if parent object is deleted, all child objects will also be deleted. Let's take again an example of relationship between House and Rooms. House can contain multiple rooms - there is no independent life of room and any room cannot belong to two different houses. If we delete the house - room will automatically be deleted. Let's take another example relationship between Questions and Options. Single questions can have multiple options and option cannot belong to multiple questions. If we delete questions options will automatically be deleted.

Association

It represents a relationship between two or more objects where all objects have their own lifecycle and there is no owner. The name of an association specifies the nature of relationship between objects. This is represented by a solid line.

Let's take an example of relationship between Teacher and Student. Multiple students can associate with a single teacher and a single student can associate with multiple teachers. But there is no ownership between the objects and both have their own lifecycle. Both can be created and deleted independently.



Aggregation

It is a specialized form of Association where all object have their own lifecycle but there is ownership. This represents "whole-part or a-part-of" relationship. This is represented by a hollow diamond followed by a line.

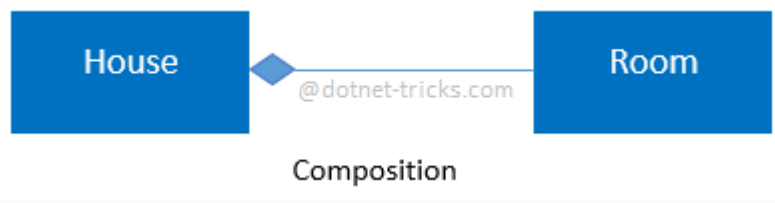
Let's take an example of relationship between Department and Teacher. A Teacher may belongs to multiple departments. Hence Teacher is a part of multiple departments. But if we delete a Department, Teacher Object will not destroy.

Aggregation

Composition

It is a specialized form of Aggregation. It is a strong type of Aggregation. In this relationship child objects does not have their lifecycle without Parent object. If a parent object is deleted, all its child objects will also be deleted. This represents "death" relationship. This is represented by a solid diamond followed by a line.

Let's take an example of relationship between House and rooms. House can contain multiple rooms there is no independent life of room and any room cannot belongs to two different house if we delete the house room will automatically delete.



Composition

Let's take another example of relationship between Questions and options. Single questions can have multiple options and option cannot belong to multiple questions. If we delete questions options will be automatically deleted.

To avoid confusion henceforth for these three terms, I have put forward a table below which will help us compare them from three angles: owner, lifetime, and child object.

	Association	**Aggregation**	**Composition**
Owner	No owner	Single owner	Single owner
Life time	Have their own lifetime	Have their own lifetime	Owner's life time
Child object	Child objects all are independent	Child objects belong to a single parent	Child objects belong to a single parent

Aggregation and Composition:

Simple rules:

1. A "owns" B = Composition : B has no meaning or purpose in the system without A
2. A "uses" B = Aggregation : B exists independently (conceptually) from A

Example 1:

A Company is an aggregation of People. A Company is a composition of Accounts. When a Company ceases to do business its Accounts cease to exist but its People continue to exist.

Example 2: (very simplified)

A Text Editor owns a Buffer (composition). A Text Editor uses a File (aggregation). When the Text Editor is closed, the Buffer is destroyed but the File itself is not destroyed.