

UML Association (Composition vs Aggregation)

Association is generalized concept of relations. It includes both Composition and Aggregation.

Composition(*mixture*) is a way to wrap simple objects or data types into a **single unit**. Compositions are a critical building block of many basic data structures

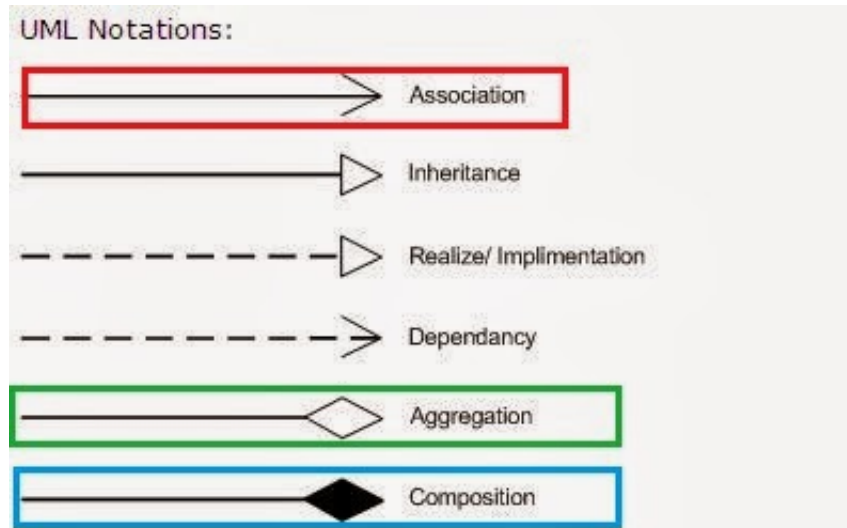
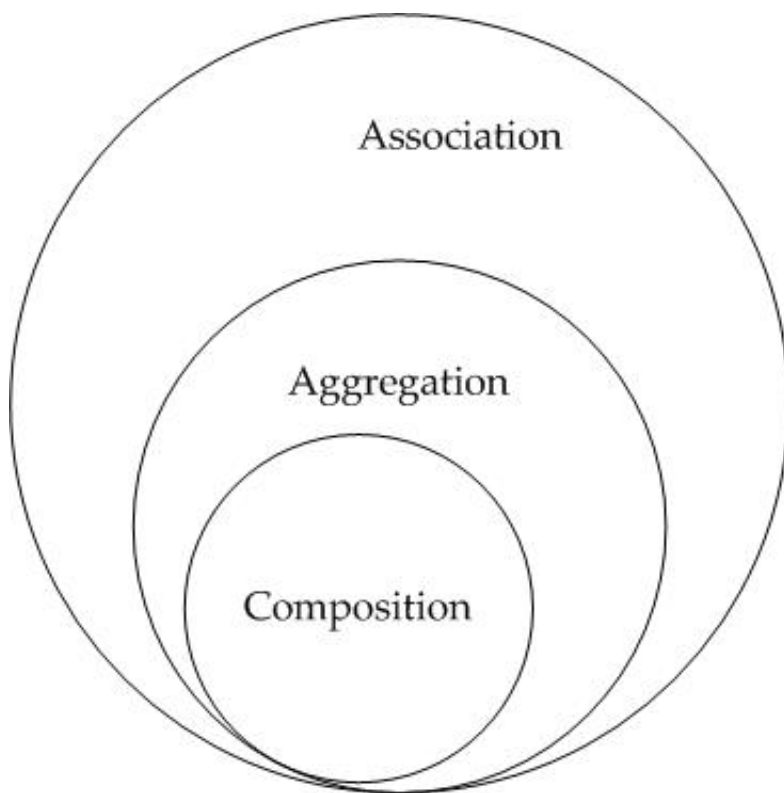
Aggregation(*collection*) differs from ordinary composition in that it does not imply ownership. In composition, when the owning object is destroyed, so are the contained objects. In aggregation, this is not necessarily true.

Both denotes relationship between object and only differ in their strength.

Trick to remember the difference : has **A** -Aggregation and **O**wn - cOmpositoin

	Aggregation	Composition
Life time	Have their own lifetime	Owner's life time
Child object	Child objects belong to a single parent	Child objects belong to a single parent
Relation	Has-A	Owns
Example	Car and Driver	Car and Wheels

Now let observe the following image

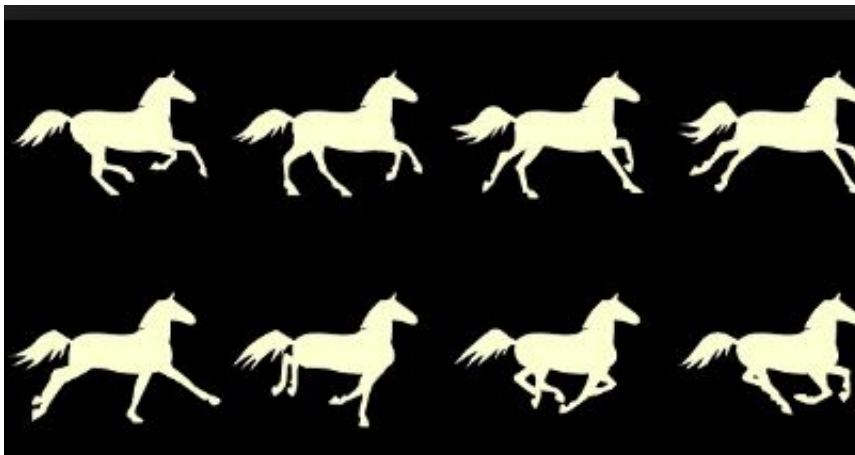


Analogy:

Composition: The following picture is image composition i.e. using individual images making one image.



Aggregation : collection of image in single location



[For example](#), A university owns various departments, and each department has a number of professors. If the university closes, the departments will no longer exist, but the professors in those departments will continue to exist. Therefore, a University can be seen as a composition of departments, whereas departments have an aggregation of professors. In addition, a Professor could work in more than one department, but a department could not be part of more than one university.