Association vs. Aggregation vs. Composition

Association represents the relationship between two classes. It can be unidirectional(one way) or bidirectional(two way):
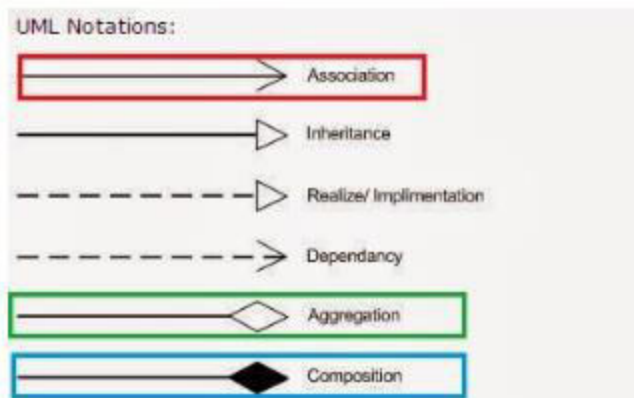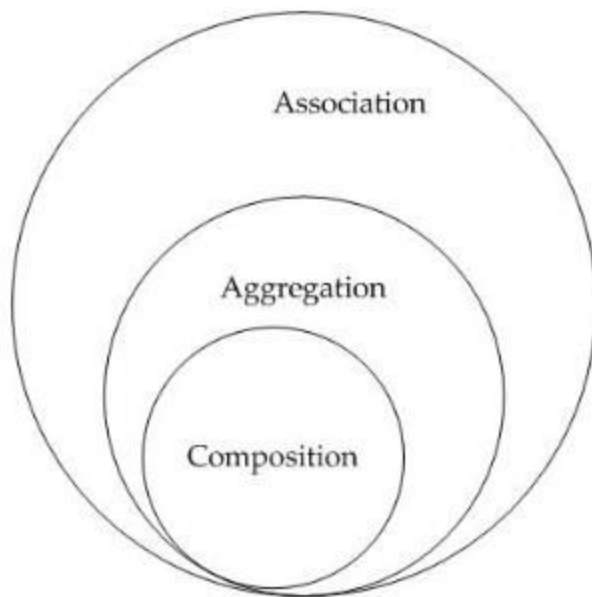
Unidirectional:
Customer places orders
Bidirectional: A is married to B , B is married to A

Aggregation : In Aggregation the dependent object is standalone and can exist even if the object of owning class is dead.

Car		Wheel
	◇———————	

Company		Employee
	◇———————	

Composition: In Composition the object of class that is owned by the object of its owning class **cannot live on it's own**

Car		Specific Engine
	◆———————	

Company		Accounts
	◆———————	

Association

Aggregation

Composition

UML Notations:

———————————▷ Association

———————————▷ Inheritance

– – – – – – – –▷ Realize/ Implimentation

– – – – – – – –> Dependancy

———————————◇ Aggregation

———————————◆ Composition

Simple rules to Remember:

A "owns" B = **Composition** : B has no meaning or purpose in the system without A
A "uses" B = **Aggregation** : B exists independently (conceptually) from A
A "belongs/Have" B= **Association**; A and B exists just have a relation

Example:

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.