**UML (Unified Modeling Language)**

# **UML Association vs Aggregation vs Composition**

* 1. **Association**

If two classes in a model need to communicate with each other, there must be a link between them, and that can be represented by an association (connector).

**Association**- A connection between two elements of a Model. This might represent a member variable in code, or the association between a personnel record and the person it represents, or a relation between two categories of workers, or any similar relationship. By default, both elements in an Association are equal, and are aware of each other through the Association. An Association can also be a Navigable Association, meaning that the source end of the association is aware of the target end, but not vice versa.

### Reflexive association

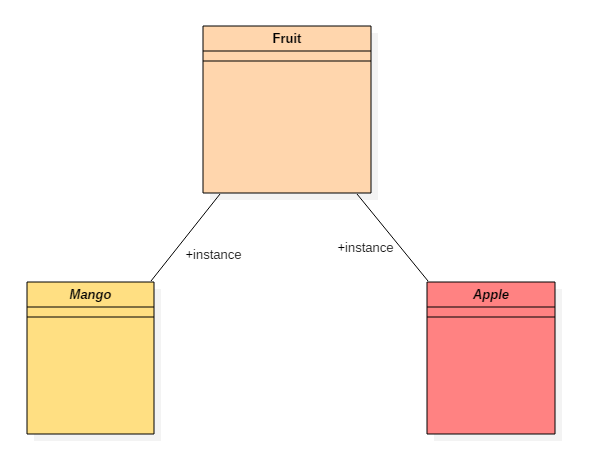
The reflexive association is a subtype of association relationship in UML. In a reflexive association, the instances of the same class can be related to each other. An instance of a class is also said to be an object.

Reflexive association states that a link or a connection can be present within the objects of the same class.

**Example**:

Let us consider an example of a class fruit.

The fruit class has two instances, such as mango and apple. Reflexive association states that a link between mango and apple can be present as they are instances of the same class, such as fruit.

[](https://www.guru99.com/images/1/062819_0756_UMLAssociat1.png)

### Directed association

As the name suggests, the directed association is related to the direction of flow within association classes.

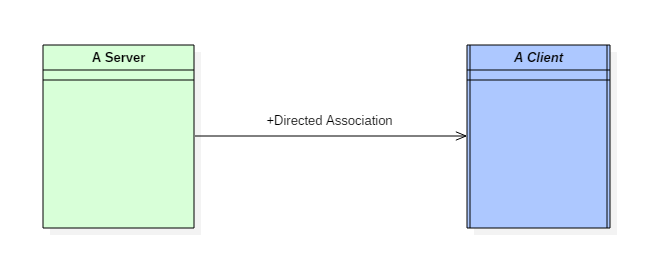
In a directed association, the flow is directed. The association from one class to another class flows in a single direction only.

It is denoted using a solid line with an arrowhead.

Example:

You can say that there is a directed association relationship between a server and a client.

A server can process the requests of a client. This flow is unidirectional, that flows from server to client only. Hence a directed association relationship can be present within servers and clients of a system.

[](https://www.guru99.com/images/1/062819_0756_UMLAssociat2.png)

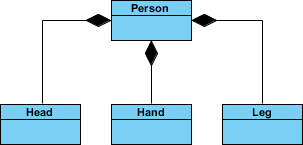
directed association

* 1. **Aggregation:**
* **Aggregation**and**Composition** are subsets of association meaning they are specific cases of association. In both aggregation and composition object of one class "owns" object of another class. But there is a subtle difference:
* **Aggregation** implies a relationship where the child can exist independently of the parent. Example: Class (parent) and Student (child). Delete the Class and the Students still exist.
* **Composition** implies a relationship where the child cannot exist independent of the parent. Example: House (parent) and Room (child). Rooms do not exist separate to a House.

Note: both aggregation and composition comes under aggregation but behavior changes in terms of uses as described below.

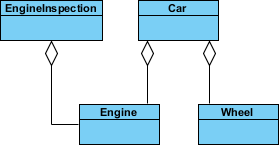
### **Composition Example:**

* We should be more specific and use the composition link in cases where in addition to the part-of relationship between Class A and Class B - there's a strong lifecycle dependency between the two, meaning that when Class A is deleted then Class B is also deleted as a result



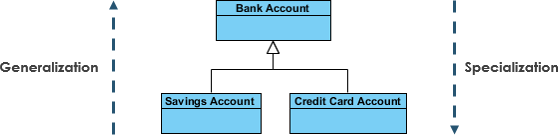
### **Aggregation Example:**

* It's important to note that the aggregation link doesn't state in any way that Class A owns Class B nor that there's a parent-child relationship (when parent deleted all its child's are being deleted as a result) between the two. Actually, quite the opposite! The aggregation link is usually used to stress the point that Class A instance is not the exclusive container of Class B instance, as in fact the same Class B instance has another container/s.



## Generalization vs Specialization

* Generalization is a mechanism for combining similar classes of objects into a single, more general class. Generalization identifies commonalities among a set of entities. The commonality may be of attributes, behavior, or both. In other words, a superclass has the most general attributes, operations, and relationships that may be shared with subclasses. A subclass may have more specialized attributes and operations.
* Specialization is the reverse process of Generalization means creating new sub-classes from an existing class.
* For Example, a Bank Account is of two types - Savings Account and Credit Card Account. Savings Account and Credit Card Account inherit the common/ generalized properties like Account Number, Account Balance, etc. from a Bank Account and also have their specialized properties like unsettled payment etc.



**Class Diagram** - Shows the system classes and relationships between them.