

Relationship

There are five different types of relationships: **Inheritance**, **Association**, **Aggregation**, **Composition** and **Dependency**.

Association


- Association is a semantically weak relationship (a semantic dependency) between otherwise unrelated objects.
- An association is a "using" relationship between two or more objects in which the **objects have their own life time and there is no owner**
- **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. In other words, the objects that are part of the association relationship are created and destroyed independently.

- **N association relationship** can be represented as (also known as **cardinality**)
 - one-to-one
 - one-to-many
 - Many-to-many.
- An association relationship between two or more objects denotes a path of communication (also called a link) between them so that one object can send a message to another.
- This is represented by a solid line in UML(Unified Modelling language)



Aggregation


- Aggregation is a specialized form of association between two or more objects in which the objects have their **own life-cycle** but there exists an **ownership** as well.
- Aggregation is a typical **whole/part relationship**
- Let's take an example of relationship between **Department and Teacher**. A Teacher may belong to multiple departments. Hence Teacher is a part of multiple departments. But if we delete a Department, Teacher Object will destroy relation from that department only.

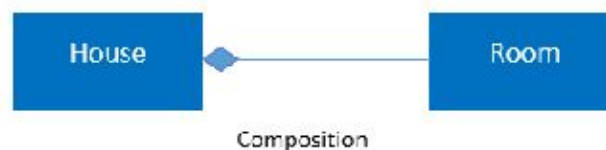
- This represents "whole-part or a-part-of" relationship. This is represented by a hollow diamond followed by a line. 



- Let us take another example, The Tires can be taken off from the car object and installed on a different one. Also, if the car gets totalled, the tires do not necessarily have to be destroyed.

Composition


- Composition is a specialized form of aggregation in which if the parent object is destroyed, the child objects would be destroyed.
- It is actually a strong type of aggregation and is also referred to as a **death relationship**.
- Composition is also a whole/part relationship but unlike aggregation, here the lifetime of the "part" is controlled by the "whole".
- 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.



- 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.
- One more example : Body -> Blood Cell, When the Body object is destroyed the BloodCells get destroyed with it.


Dependency

- It represents a relationship between two or more objects where an object is dependent on another object(s) for its specification or implementation. This is represented by a dashed arrow. ----->
- Let's take another example of relationship between a **client** and a **supplier**. A client is dependent on the supplier for supplying products. If the supplier will not supply the products, client cannot use those products.

ahead of the curve.

Web Designing	Java	C, C++	Python	Project Training
	Php	DBMS, Oracle		
	.NET, C#, ASP		CDAC-CAT	
	Hardware Programming			



ahead of the curve.

CDAC - CAT

webDesigning
Java, MS.net
Oracle, Training
Python, C, C++

Akhillesh Gupta
9981315087