

Difference Between Aggregation and Composition

www.differencebetween.com

Key Difference - Aggregation vs Composition

Object-Oriented Programming ([OOP](#)) is a common paradigm in [software](#) development. The [object](#) is an instance of a class. It is not possible to create objects at once. There should be a blueprint or a description to create an object. That blueprint is known as a class. A class contains properties and methods. Objects are created using classes. Class and object are similar to a plan and house in the real world. It is not possible to build a house without a proper plan. Likewise, a class is used to create an object. An object collaborates with other objects. A link with represents the relationship of two or more objects is called as an "association". Aggregation and composition are types of associations. They describe the relationship among classes. This article discusses the difference between aggregation and composition. The **key difference** between aggregation and composition is that **aggregation is an association between two objects that describes the "has a" relationship and composition is a more specific type of aggregation that implies ownership.**

What is Aggregation?

A relationship between two objects is shown by drawing a line in Unified Modelling Language([UML](#)). The link is an association. UML helps to get a visual representation of the system. It is different than regular [programming languages](#). Association also defines the multiplicity of objects. They are one-to-one, one-to-many and many-to-many. When a single object of class A is associated with the single object of class B, that is a one-to-one relationship. An example is an author writing a book. In that example, one author is writing a book.

When one object of class A is associated with many objects of class B, it is a one-to-many relationship. An example is, a department can have many employees. When the object of class A is associated with many objects of class B and an object of class B is associated with many objects of class A, it is a many-to-many association. One example is, an employee can work on multiple projects and a project can have multiple employees.

Aggregation is type association that further describe the relationship between objects. Aggregation describes "has a" relationship. Some examples that describe the relationship are, a student "has a" student id, a vehicle "has an"

engine. It is also possible to expand a large quantity with the relationship. Some examples are, a bank “has many” bank accounts, a class “has many” students. It can be explained using the below example.

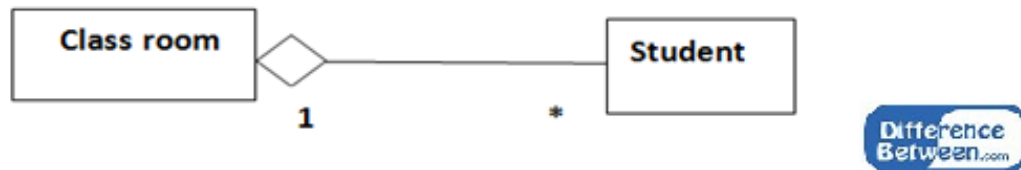


Figure 01: Aggregation

According to the above example, the classroom consists of a student or many students. Multiplicity is also used to indicate the number of objects. It describes that one classroom has multiple students. The symbol of the diamond represents the aggregation in UML. Student objects do not rely on the class object. If the class object is destroyed, it won't affect the student objects. Those objects will still exist.

What is Composition?

The composition is a more specific form of aggregation. It describes the ownership. It can be explained using the below example.

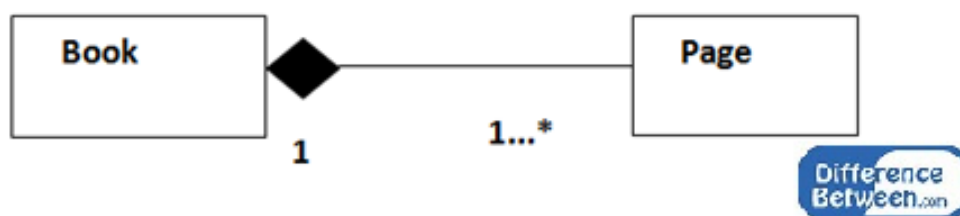


Figure 02: Composition

According to the above, the book object consists of page object or pages. Multiplicity is also used to indicate the number of objects. It describes that one classroom has multiple students. The symbol of diamond that is highlighted represents the composition in UML. As the book has a page or many pages, it is an aggregation, but it is more specified. If the book object is destroyed, then the page objects also get destroyed. Page objects cannot exist

without the book object. Therefore, the composition is a more specific form of an aggregation that implies ownership.

What are the Similarities Between Aggregation and Composition?

- Both are used in Object Oriented Programming.
- Both are used in Unified Modeling Language (UML) to get a visual understanding of the system.

What is the Difference Between Aggregation and Composition?

Aggregation vs Composition	
Aggregation is an association between two objects which describes the "has a" relationship.	The composition is the most specific type of aggregation that implies ownership.
UML Symbol	
Aggregation is denoted by a diamond.	Composition is denoted by a highlighted diamond.
Functionality	
In aggregation, if the owning object destroyed, it will not affect the containing object.	In composition, if the owning object destroyed, it will affect the containing object.

Summary - Aggregation vs Composition

Object-Oriented Programming is a major paradigm in software development. In OOP, the system is modeled using objects. These objects do not exist in isolation. Objects collaborate with other objects. The relationship among objects is known as association. Aggregation and composition are types of association. The difference between aggregation and composition is that aggregation is an association between two objects which describes the "has a" relationship and composition is the more specific type of aggregation which implies ownership. Aggregation and composition both help to understand the behaviour of the system.

Reference:

1. tutorialspoint.com. "OOAD Object Model." [The Point. Available here](#)
2.6 5 Using aggregation and composition, People career, 22 May
2015. [Available here](#)

How to Cite this Article?

APA: Difference Between Aggregation and Composition. (2018 January 18). Retrieved (date), from <http://differencebetween.com/difference-between-aggregation-and-vs-composition/>

MLA: "Difference Between Aggregation and Composition" Difference Between.Com. 18' January 2018. Web.

Chicago: "Difference Between Aggregation and Composition". Difference Between.Com. <http://differencebetween.com/difference-between-aggregation-and-vs-composition/> accessed (accessed [date]).



Copyright © 2010-2017 Difference Between. All rights reserved