SNS COLLEGE OF ENGINEERING

Kurumbapalayam(Po), Coimbatore – 641 107
Accredited by NAAC-UGC with ‘A’ Grade
Approved by AICTE, Recognized by UGC & Affiliated to Anna University, Chennai

Department of Information Technology

Object Oriented Software Engineering

Object Modelling using UML

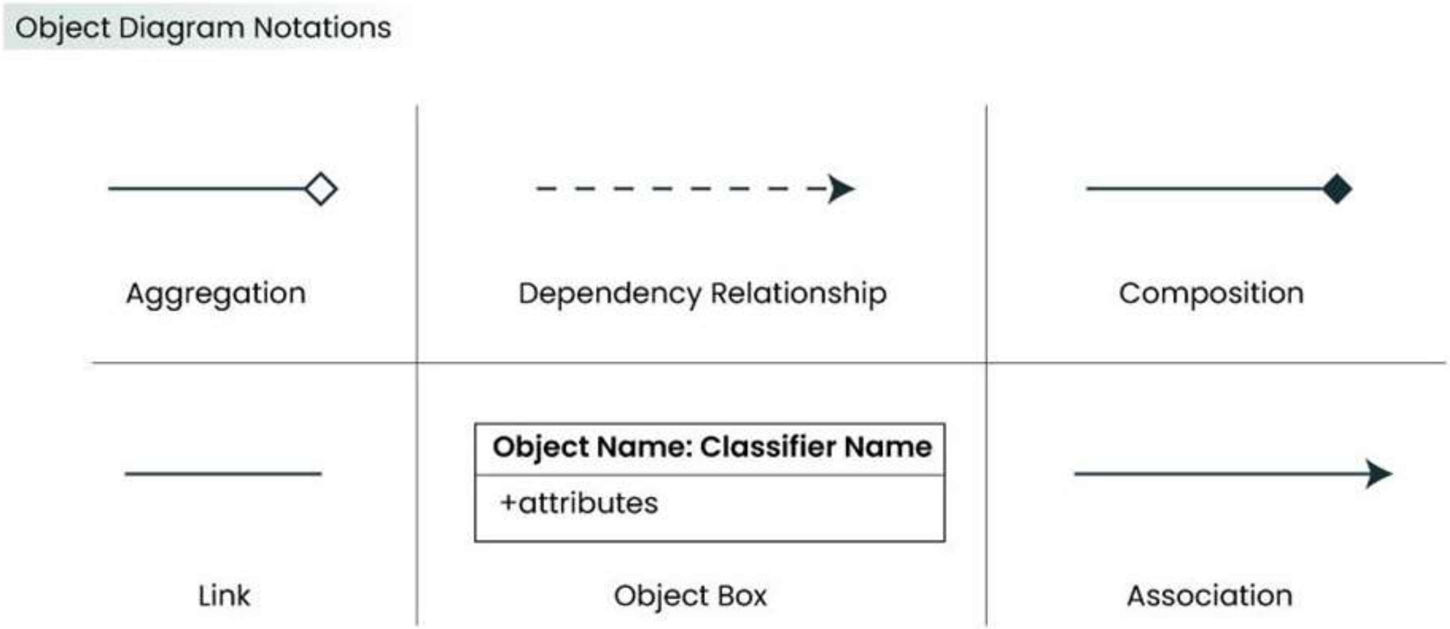
Prepared By
R.Vaishnavi.,AP/IT
SNSCE.

Object Diagram :

- An Object Diagram can be referred to as a screenshot of the instances in a system and the relationship that exists between them.
- An object diagram in UML is useful because it provides a clear and visual representation of specific instances of classes and their relationships at a particular point in time, aiding in understanding and communicating the structure and interactions within a system.
- In other words, “An object diagram in the Unified Modeling Language (UML), is a diagram that shows a complete or partial view of the structure of a modeled system at a specific time.

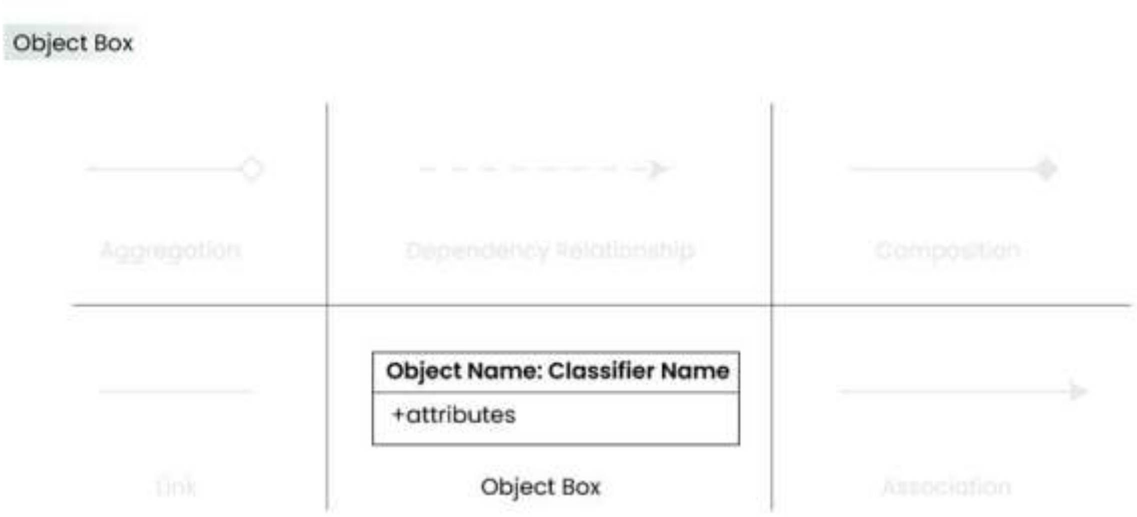
Object Diagram Notations

The object diagram in UML uses specific notations to represent instances of classes and their relationships at a particular moment in time.



1. Objects or Instance specifications

- When we instantiate a classifier in a system, the object we create represents an entity which exists in the system.
- We can represent the changes in object over time by creating multiple instance specifications.
- We use a rectangle to represent an object in an object diagram.

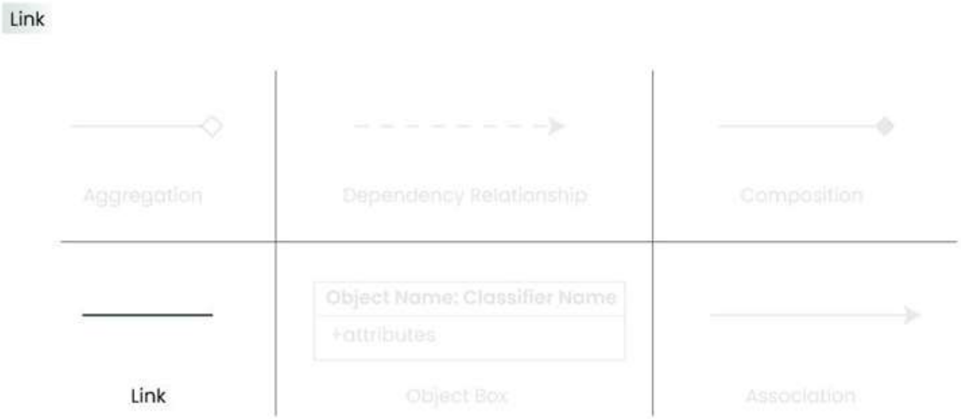


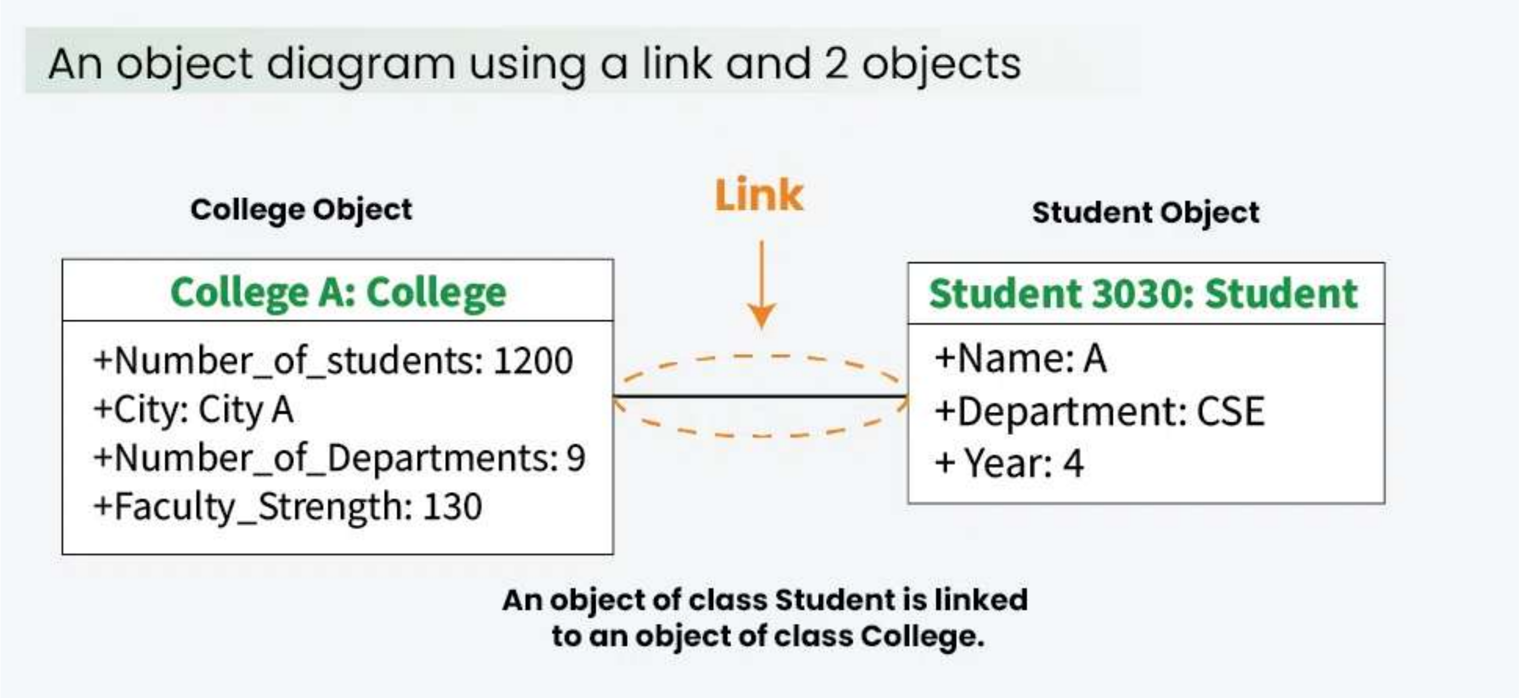
2. Attributes and Values

Inside the object box, attributes of the object are listed along with their specific values.

3. Link

We use a link to represent a relationship between two objects. We represent the number of participants on the link for each, at the end of the link. The term link is used to specify a relationship between two instance specifications or objects. We use a solid line to represent a link between two objects.





An object diagram using a link and 2 objects

College Object

College A: College

+Number_of_students: 1200
+City: City A
+Number_of_Departments: 9
+Faculty_Strength: 130

Link

Student Object

Student 3030: Student

+Name: A
+Department: CSE
+ Year: 4

An object of class Student is linked to an object of class College.

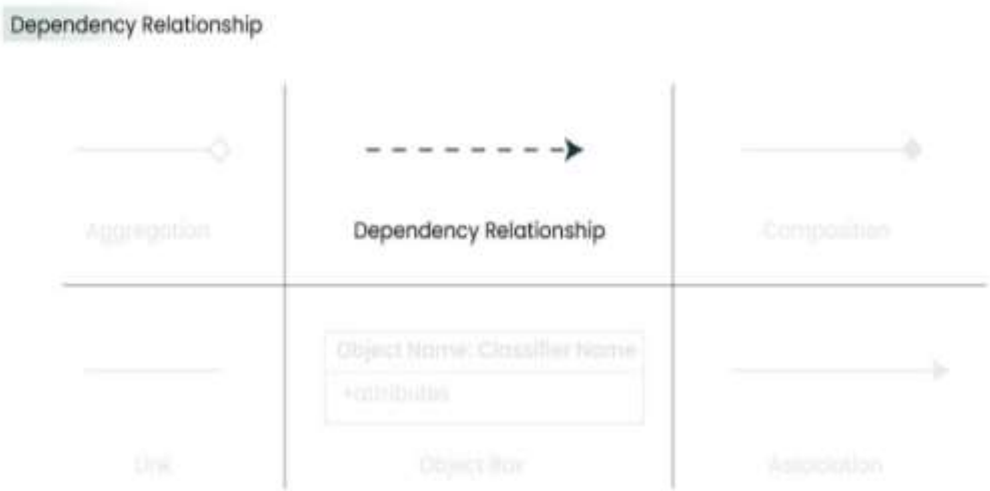
4. Dependency Relationships

We use a dependency relationship to show when one element depends on another element. A dependency is used to depict the relationship between dependent and independent entities in the system.

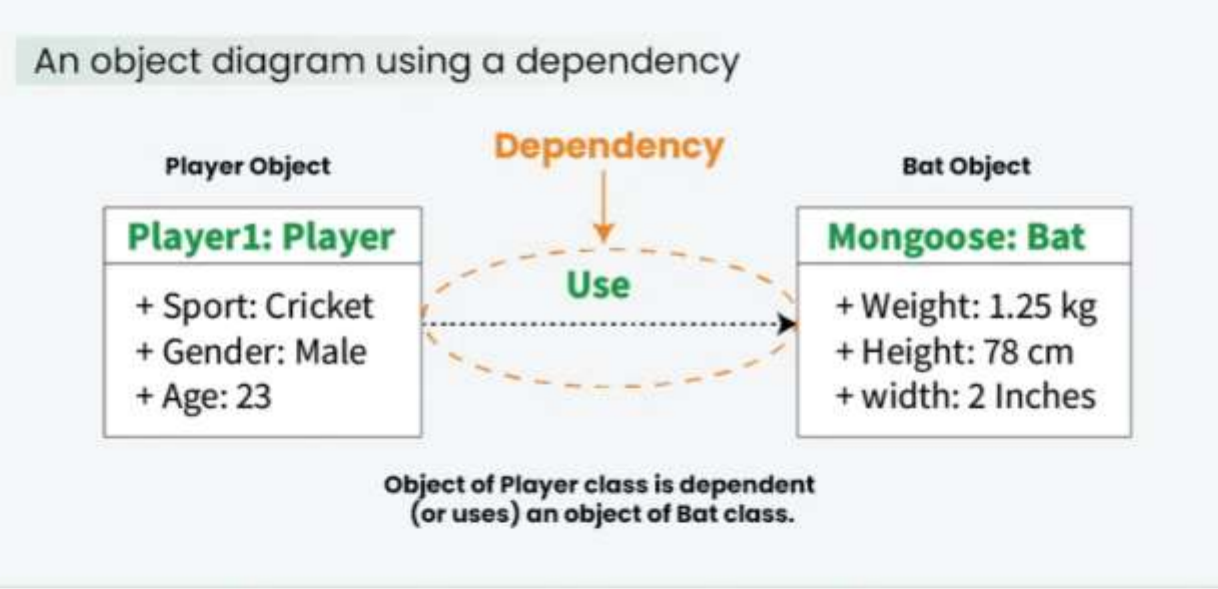
Any change in the definition or structure of one element may cause changes to the other.

This is a unidirectional kind of relationship between two objects.

Dependency relationships are of various types specified with keywords like Abstraction, Binding, Realization, Substitution and Usage are the types of dependency relationships used in UML.



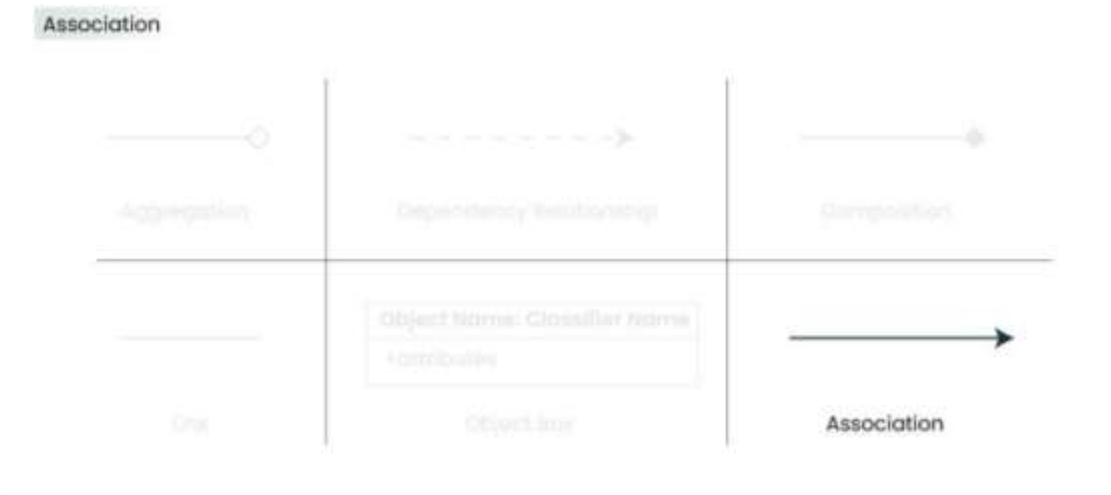
For example – In the figure below, an object of Player class is dependent (or uses) an object of Bat class.



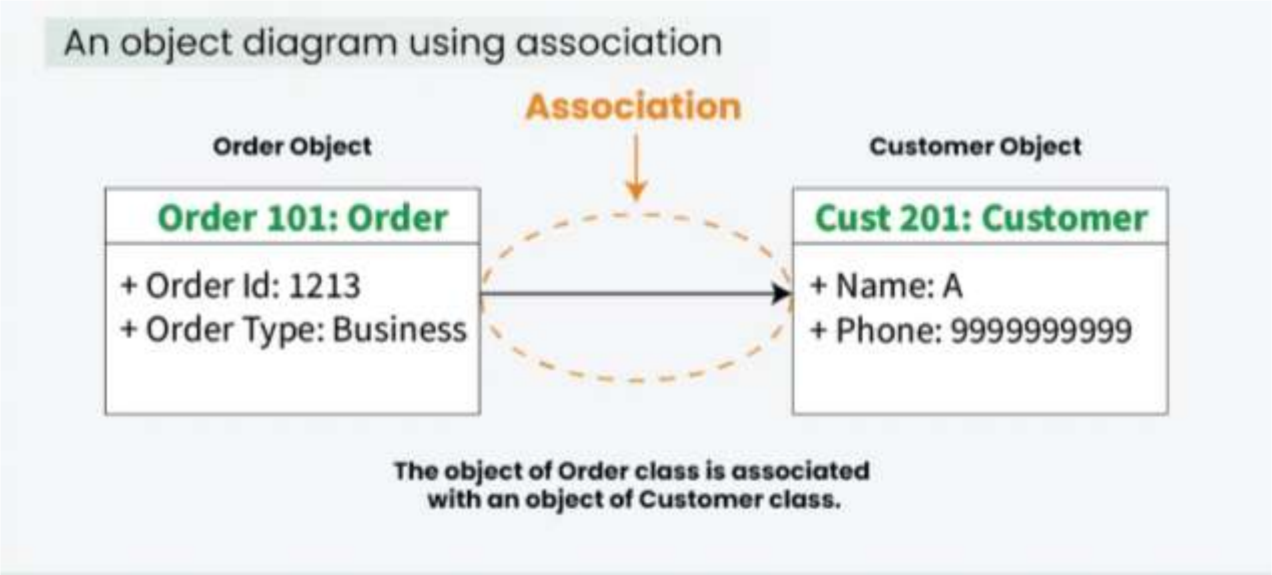
5. Association

Association is a reference relationship between two objects (or classes). An association line connects two object boxes, representing a relationship between instances of two classes.

We use association when one object references members of the other object. Association can be uni-directional or bi-directional. We use an arrow to represent association.



For example – The object of Order class is associated with an object of Customer class.



An object diagram using association

Order 101: Order

+ Order Id: 1213
+ Order Type: Business

Cust 201: Customer

+ Name: A
+ Phone: 9999999999

The object of Order class is associated with an object of Customer class.

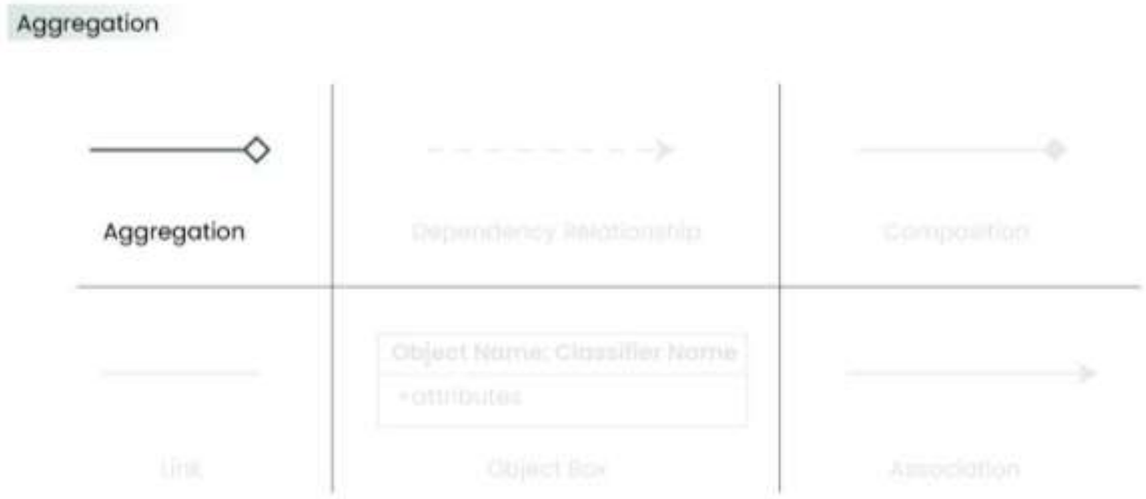
6. Aggregation

Aggregation represents a “has a” relationship. We use a hollow diamond on the containing object with a line which joins it to the contained object.

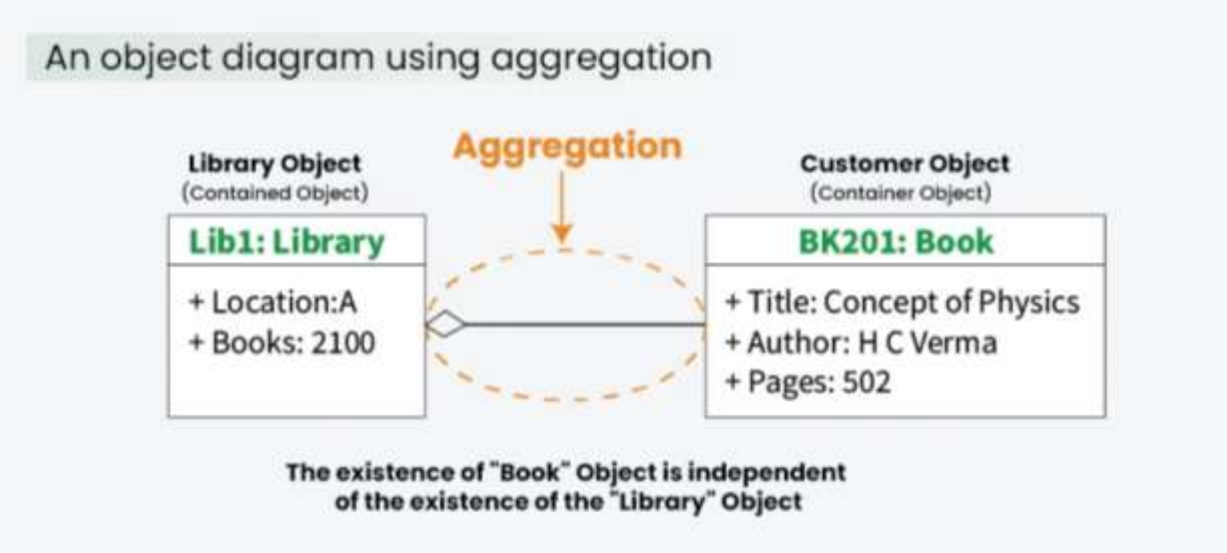
Aggregation is a specific form of association.

It is a kind of parent-child relationship however it isn’t inheritance.

Aggregation occurs when the lifecycle of the contained objects does not strongly depend on the lifecycle of container objects.



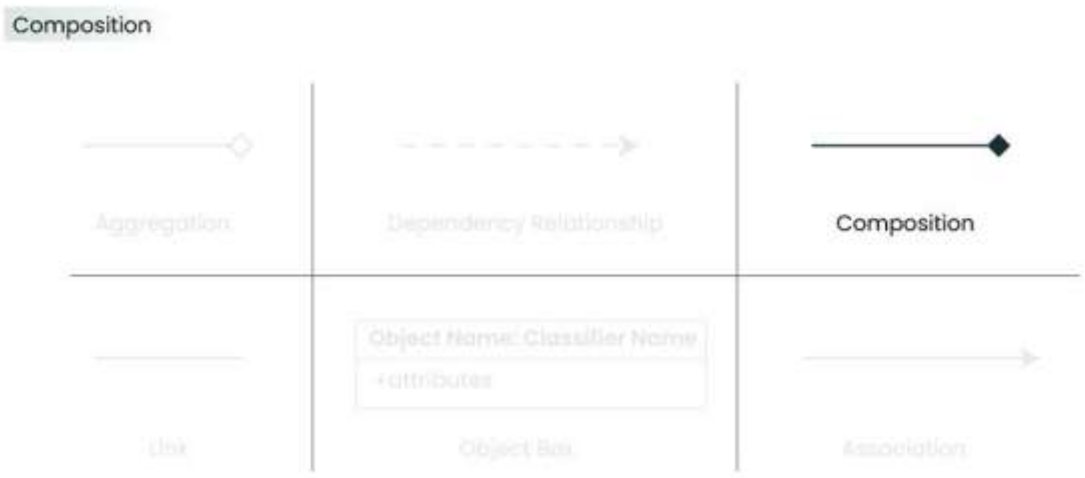
For example – A library has an aggregation relationship with books. Library has books or books are a part of library. The existence of books is independent of the existence of the library.



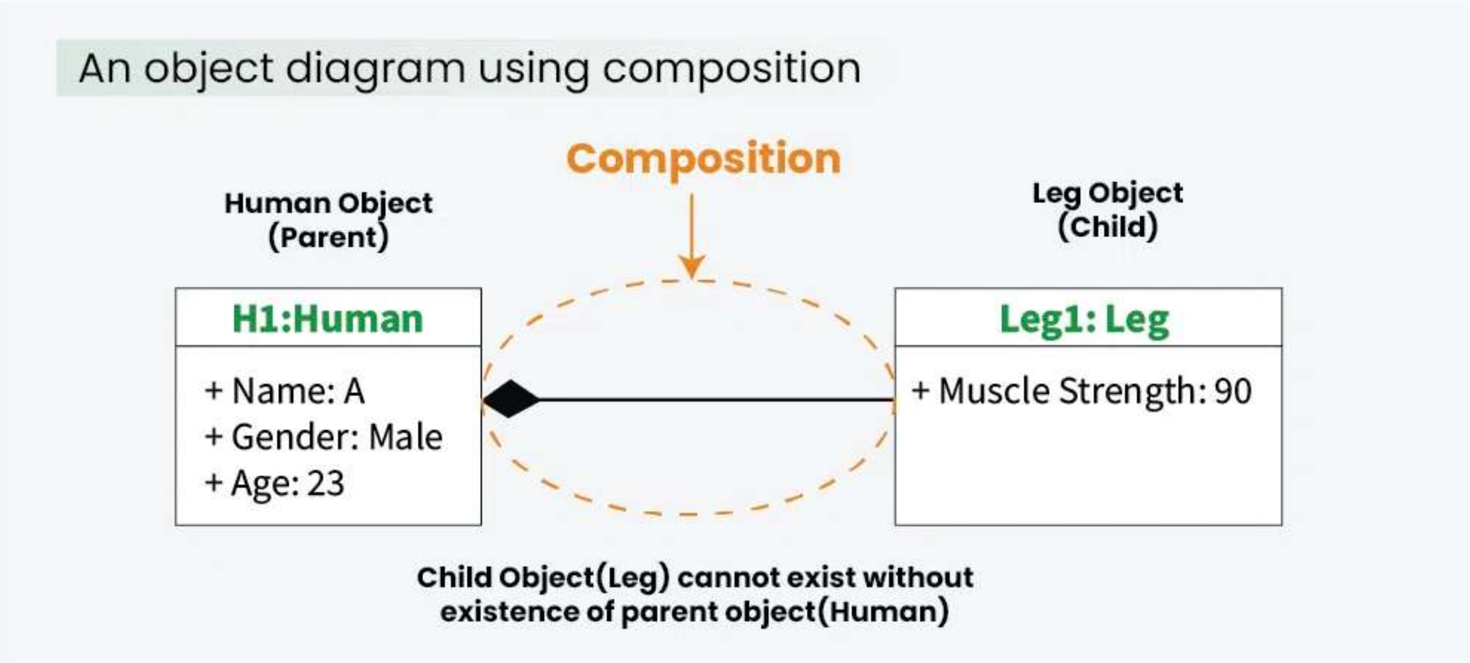
7. Composition

Composition is a type of association where the child cannot exist independent of the other. We use a filled diamond on the containing object with a line which joins it to the contained object.

Composition is also a special type of association. It is also a kind of parent child relationship but it is not inheritance. So whenever independent existence of the child is not possible we use a composition relationship.



Consider the example of a boy Gurkaran: Gurkaran is composed of legs and arms. Here Gurkaran has a composition relationship with his legs and arms. Here legs and arms can’t exist without the existence of their parent object.



Purpose of Object Diagrams

The main purpose of using object diagrams is:

- They offer a detailed view of how objects interact with each other in specific scenarios.
- Proper design and analysis of applications can be faster and efficient.
- Object diagrams are beneficial during the implementation phase of software development.
- Promoting a shared understanding of specific instances and their relationships, facilitating collaboration among team members.

Benefits of Object Diagrams

- Detailed Insight into Relationships
- Implementation Guidance
- Integration Testing Assistance
- Validation of Code Implementation
- Scenario Illustration

How to draw an Object Diagram?

- Identify Classes
- Identify Objects
- Create Object Boxes
- Add Attributes and Values
- Draw Relationships
- Label Relationships
- Review and Refine
- Use Tools for Digital Drawing