

### ER model

An entity is a thing that exists either physically or logically. Entities can be thought of as nouns: a company, a computer.

A relationship captures how entities are related to one another.

Relationships can be thought of as verbs, linking two or more nouns.

Entities and relationships can both have attributes.

### Instance-level r/s in class diagrams

**dependency** connection between dependent and independent model elements; exists when changes to one element may cause changes in dependent element; this relation is uni-directional

**association** association is a relationship between two classes when, that allows one instance to perform an action on behalf of another

**aggregation** aggregation is a variant of "has a" relationship; it can occur when a class is a collection of other classes

**composition** more specific version of aggregation; when container destroyed every instance if contains will be destroyed as well; composition unlike aggregation is a "whole-part" relationship

Both aggregation and composition are types of association between classes. The aggregation relationship is often "catalog" containment to distinguish it from composition's "physical" containment.

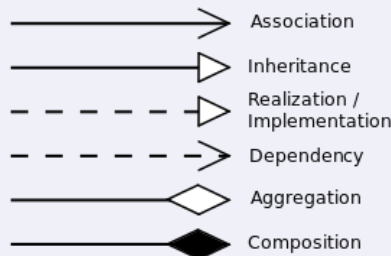
### UML diagrams

**class diagram** a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

**object diagram** shows a complete or partial view of the system at a given moment of time

**domain model** conceptual model of the domain that incorporates both: behaviour and data

### UML relations notation



### class level relationship

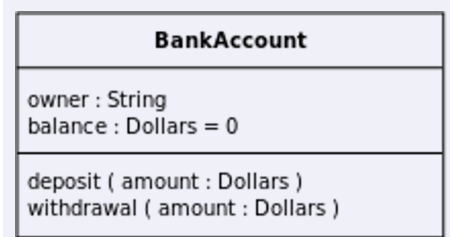
**inheritance** implements "is a" relationship;

**realization** relationship between component and its interface

### general relationship

**dependency** weaker form of bond that indicates that one class is dependent on the other; one class depends on another when the independent class is a parameter or local variable

### Class diagram



Three compartments of class diagram:-

Name of the class

Attributes of the class

Methods of the class

### Class visibility diagram

+ public  
- private  
# protected  
/ derived  
~ package

To specify the visibility of a class member (i.e. any attribute or method), these notations must be placed before the member's name

