Course Name - Object Oriented Programming using Java

Lecture 2– Relationships among objects and classes-Generalization, Specialization, Aggregation, Association, Composition, links, Meta-class.

Presented By
Dr. Sudipta Sahana
Associate Professor
Dept. of CSE
UEM - Kolkata

Topic of Interest

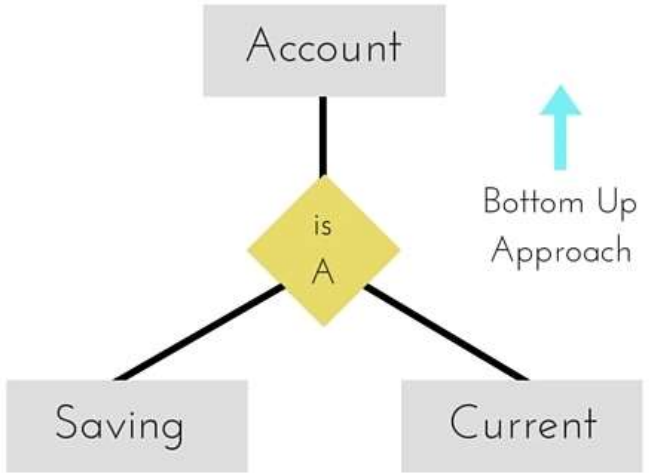
- Relationships among objects and classes
- Generalization
- Specialization
- Aggregation
- Association
- Composition
- links
- Meta-class

Generalization

Generalization uses a “is-a” relationship from a specialization to the generalization class. Common structure and behavior are used from the specialization to the generalized class. At a very broader level we can understand this as inheritance.

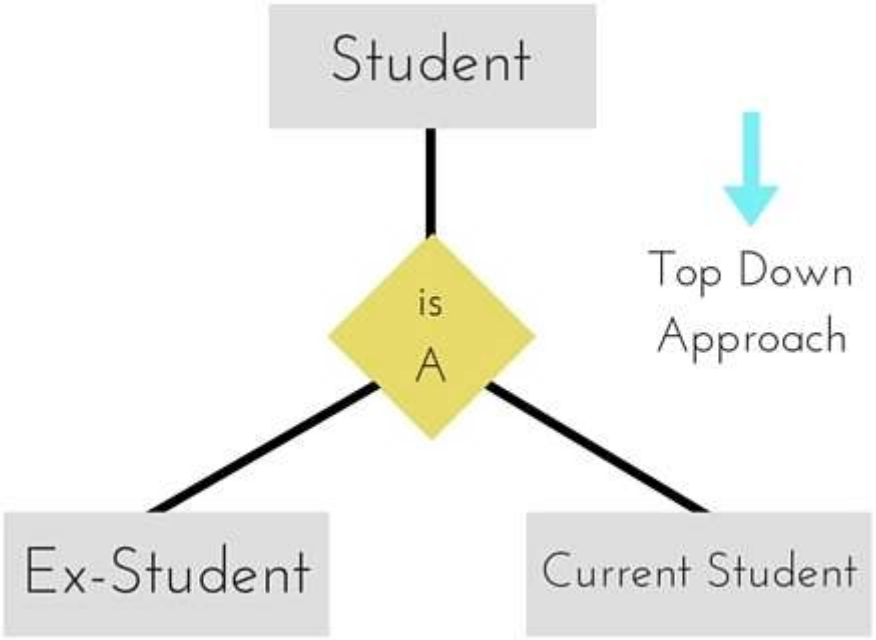
Generalization is also called a “Is-a” relationship.

Generalization is a bottom-up approach in which two lower level entities combine to form a higher level entity. In generalization, the higher level entity can also combine with other lower level entities to make further higher level entity.



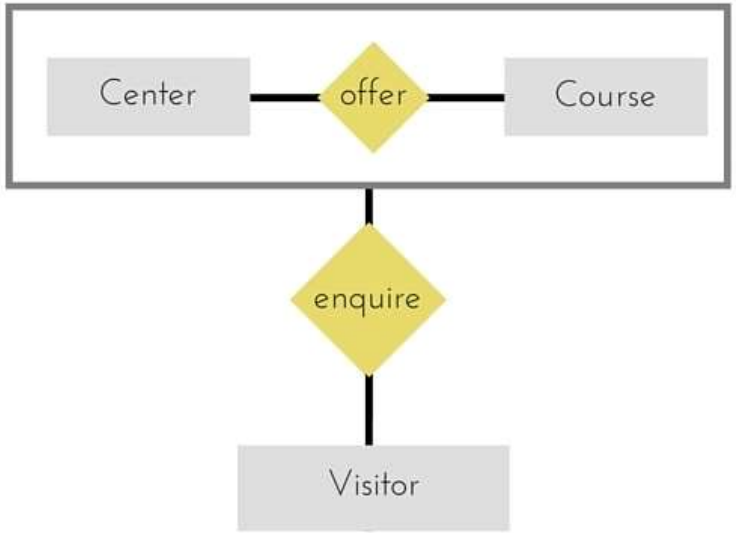
Specialization

Specialization is opposite to Generalization. It is a top-down approach in which one higher level entity can be broken down into two lower level entity. In specialization, a higher level entity may not have any lower-level entity sets, it's possible.



Aggregation

Aggregation is a process when relation between two entities is treated as a single entity.



In the diagram above, the relationship between **Center** and **Course** together, is acting as an Entity, which is in relationship with another entity **Visitor**. Now in real world, if a Visitor or a Student visits a Coaching Center, he/she will never enquire about the center only or just about the course, rather he/she will ask enquire about both.

Association

Association is a relationship between two objects. In other words, association defines the multiplicity between objects. You may be aware of one-to-one, one-to-many, many-to-one, many-to-many all these words define an association between objects. Aggregation is a special form of association. Composition is a special form of aggregation.

Example: A Student and a Faculty are having an association.

Composition

Composition is a special case of aggregation. In a more specific manner, a restricted aggregation is called composition. When an object contains the other object, if the contained object cannot exist without the existence of container object, then it is called composition.

Example: A class contains students. A student cannot exist without a class. There exists composition between class and students.

Difference between aggregation and composition

Composition is more restrictive. When there is a composition between two objects, the composed object cannot exist without the other object. This restriction is not there in aggregation. Though one object can contain the other object, there is no condition that the composed object must exist.

Example: A Library contains students and books. Relationship between library and student is aggregation. Relationship between library and book is composition. A student can exist without a library and therefore it is aggregation. A book cannot exist without a library and therefore its a composition.

Links

In object modeling links provides a relationship between the objects. These objects or instance may be same or different in data structure and behavior. Therefore a link is a physical or conceptual connection between instances (or objects).

For example: Ram works for HCL company. In this example “works for” is the link between “Ram” and “HCL Company”. Links are relationship among the objects (instance).

Meta-class

In object-oriented programming, a meta-class is a class whose instances are classes. Just as an ordinary class defines the behavior of certain objects, a meta-class defines the behavior of certain classes and their instances.

A MetaClass describes a real Class with the purpose of providing to an IDE class level information, and delaying the loading of that class to the last possible moment: when an instance of the class is required.

A MetaClass binds the Class object from its class name using the appropriate class loader. Once the class is bound, new instances can be created using the newInstance() method.

UEM
श्रद्धावान लभते ज्ञानम्
UNIVERSITY OF ENGINEERING & MANAGEMENT
Good Education, Good Jobs

Thank You