

# CS315 - September 30, 2015

## Class Diagram Lab

---

A Class Diagram has a few items on it:

- Properties
- Attributes

There is visibilities on properties (who can read this property)

- '+' {public} visible to any class
- '#' {protected} visible to class and its subclasses
- '-' {private} visible only to the class itself
- '~' {package} visible to any class within the enclosing package

Static features are features in which all instances share the same attribute slot of the class.

Class operations applies to the entire class and only depends on static members themselves.

## Abstract Classes

- Abstract classes are not instantiable.
- To be meaningful, abstract classes must be specialized.
- Concrete classes must implement inherited abstract operations.
- If a class extends an abstract class, they can inherit members in the abstract class.

## Arrows in UML

- Arrows in UML can have various purposes.
  - Closed arrows most often describe an inheritance relation
  - Open arrows are used for describing associations
  - Square arrows represent composite classes

## Generalizations

- (Generally!) Used for an inheritance relationship

## Association

- Describe which & how classes interact with each others
  - **Name** verb (maybe a noun)
  - **Reading Direction** full black arrow head
  - **Role Name**
  - **Cardinalities/Multiplicities** one to one, one to many, many to many

Associations can be both ways.

See HE's slides on Blackboard for more detailed examples.

Indicator	Meaning
1	exactly 1
0..1	Optional
0..*	Zero or more
1..*	at least one instance
5..5	Exactly 5
m..n	At least m but no more than n instances

## Class and Object Diagrams

- **Class Diagrams**
  - Classes and Associations
  - Describes declaration time elements
  - Specifies constraints on the objects and their relationships
  - Defines the set of all possible object/link configurations
- **Object Diagram**
  - Describes a snapshot at a particular time of the system.
  - Used object diagram to formally understand the constraints and multiplicities defined by the associations in a class diagram.

## Composition

- Composite, nesting.
- A composite object does not exist without its components.
- Each component is a **part** of a *single composite*.
- Typically, components are of different types.

## Warehouse Example

- Company XYZ is a manufacturing company that produces cartoon action figures for big entertainment companies.
- This company needs an inventory and tracking system
- The inventory system keeps track of how many of each **figurine** is stored in each **warehouse**
- **Figurines** are stored in **cases**
- **Client order** the figurines and cases are eventually **shipped** to clients.

Huseyin went through how to use VPP for class Diagram using the ATM Example.