

# UML: Class Diagrams and Use-Cases

Reference:  
Borland's UML tutorial

5 Mar 2010  
CMPT166  
Dr. Sean Ho  
Trinity Western University

# UML: Unified Modeling Language

- Diagrams for use in designing your programs
- Main diagram types:
  - Static: Class diagram, object, package
  - Dynamic: Use case diagram, sequence diagram, state chart
- Handy for diagramming by hand, or
- UML software tools, e.g., Visio, Sun JSEnterprise
- Developed by Booch, Rumbaugh, and Jacobson, of OMG (Object Management Group)
- Current version is 2.0: [www.uml.org](http://www.uml.org)

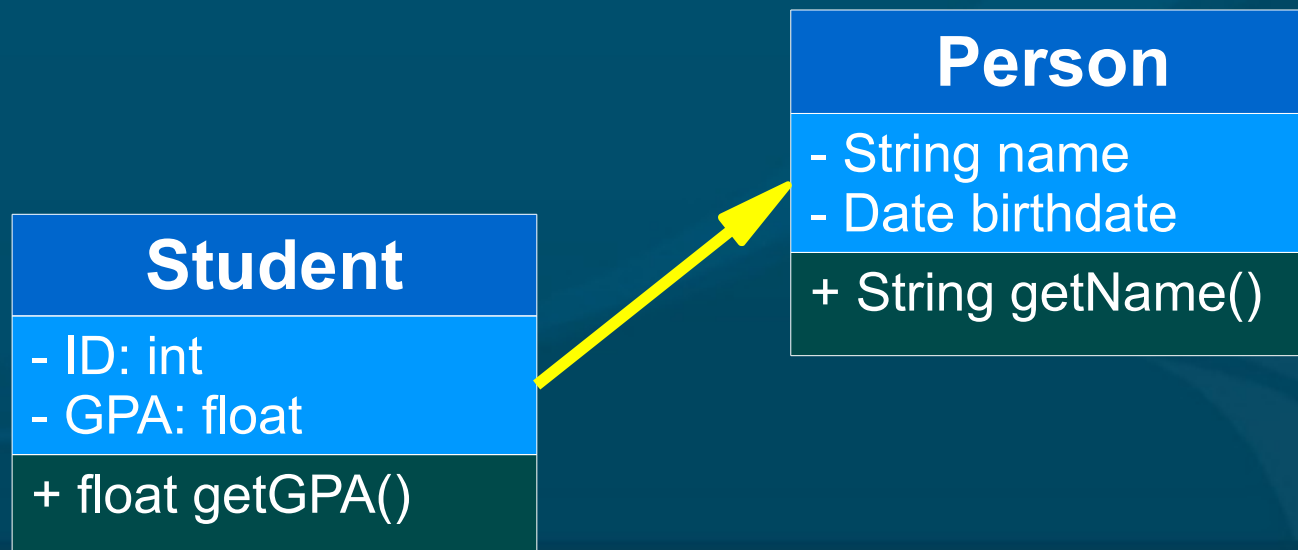
# CRC diagrams

Class Name	
Responsibilities	Collaborations
(what the class does)	(related objects)

- **Class:**
  - Short descriptive **name** for the component
- **Responsibility:**
  - **Data** stored in the class
  - Restrictions on **access** to the data
  - **Actions** the class is responsible for
- **Collaborator:**
  - e.g., types of our **attributes**/data
  - Other classes whose **methods** we call
  - Other classes who call **our methods**

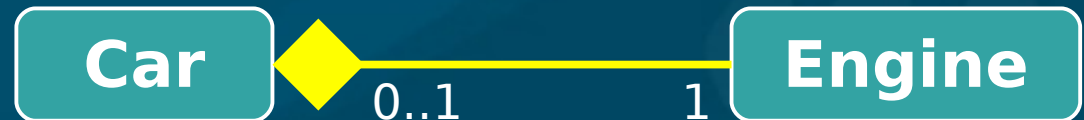
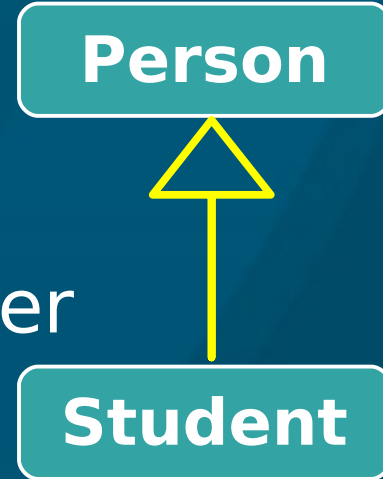
# UML class diagram

- Each box represents a **class** (type)
  - Name, attributes, methods
  - **Static** (class) members are **underlined**
  - Flag: **public** (+), **private** (-), **protected** (#)
- Lines show **relationships** between classes



# Class diagram: relationships

- Class relationships: e.g., superclass:
  - Hollow arrow-head pointing to super
- Instance relationships:
  - e.g., every Car has an instance of Engine
  - relationship between instances of the classes, not entire classes



- Multiplicity: e.g., “\*”: any number of instances
  - “1..\*”: one or more instances
  - “0..1”: optional one instance

# Instance relationships

- **Association:** label with the relationship; arrow indicates **direction** of dependency



- **Aggregation:** container  
"A is **part of** B" (but can exist apart from B)

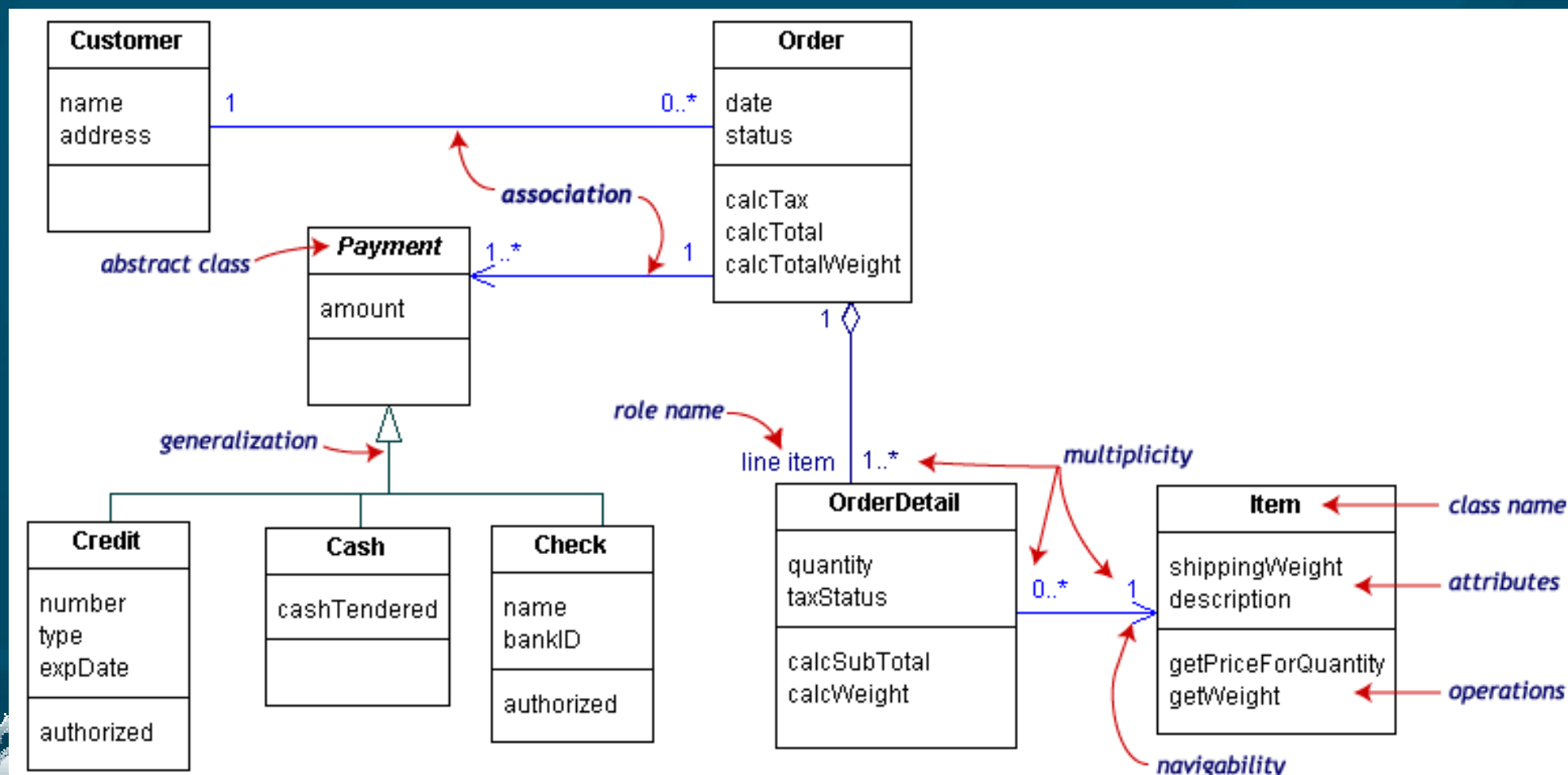


- **Composition:** "B **owns** an A"  
**Life-cycle** dependency:  
when B dies, so should its instance of A



# An example class diagram

- Ordering system:  
each Order has multiple OrderDetail line items



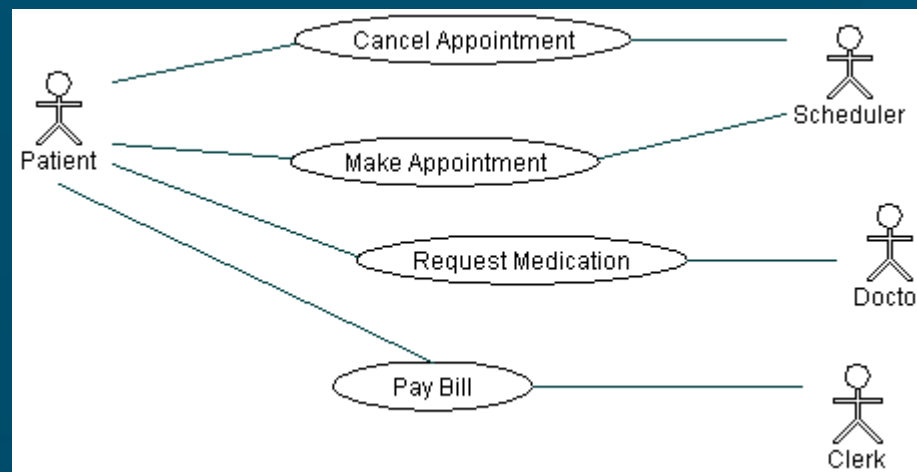
# Steps to OO design: wADes

- (Prereq: understand client requirements)
- System behaviour
  - Use-case scenarios
  - User interface mockups
- Components
  - Self-contained blocks with narrow interactions
- From components to classes
  - Attributes, methods, relationships



# UML: Use case diagram

- Describes **relationships** between **actors**:
  - ◆ **Patient** calls the clinic to make an appointment
  - ◆ **Receptionist** books timeslot
  - ◆ **Patient** sees **doctor** and requests medication
  - ◆ **Patient** pays bill to **clerk**



- See Borland's UML tutorial for more details

# System behaviour: use-case

- UML **use-case** diagrams show:
  - The **actors** involved (may be nonhuman!)
  - Ways in which the actors **interact**: relationships, actions, use cases, etc.

- Example: **ATM**  
(thanks to  
ArgoUML)

