# CS 247: Software Engineering Principles

## UML Modelling

Agenda: UML class diagrams
UML object diagrams
UML sequence diagrams

Reading: Martin Fowler, *UML Distilled, 3ed,* Addison-Wesley Professional, 2003.
(Electronic text available from UW Library Web site)

## Models

A model is an abstraction of something for the purpose of

- understanding it before building it
- communicating it to others
- answering questions about it

Metro Toronto Subway Map
Finch
North York Centre
Sheppard
York Mills
Lawrence
Eglington
Wilson
Yorkdale
Lawrence West
Glencairn
Eglington West
Ellesmere
Midland
Scarborough
McCowan
Lawrence East
Kennedy
Kipling
Islington
Royal York
Old Mill
Jane
Runnymede
High Park
Keele
Dundas West
Landsdowne
Dufferin
Ossington
Christie
Bathurst
St. Clair West
Dupont
Spadina
Bay
St. Clair
Davisville
Summerhill
Rosedale
Sherbourne
Castle Frank
Broadview
Chester
Pape
Donlands
Greenwood
Coxwell
Woodbine
Main St.
Victoria Park
Warden
St. George
Museum
Queen's Park
St. Patrick
Osgoode
St. Andrew
Spadina Ave.
Bloor/Yonge
Wellesley
College
Dundas
Queen
King
Union
Queens Quay
Rees St.
Simcoe St.
York St.
MAP LEGEND
Bloor–Danforth
Yonge–University
Scarborough RT
Harbourfront RT
Subway Interchange

## Another Model

TORONTO
Rapid Transit

http://www.toronto.ca/ttc/pdf/rideguide.pdf

# Unified Modeling Language (UML)

UML - A collection of notations for representing different views of a software design.

## Structural Diagrams

- Class diagram
- Component diagram
- Composite structure diagram
- Deployment diagram
- Object diagram
- Package diagram
- Profile diagram

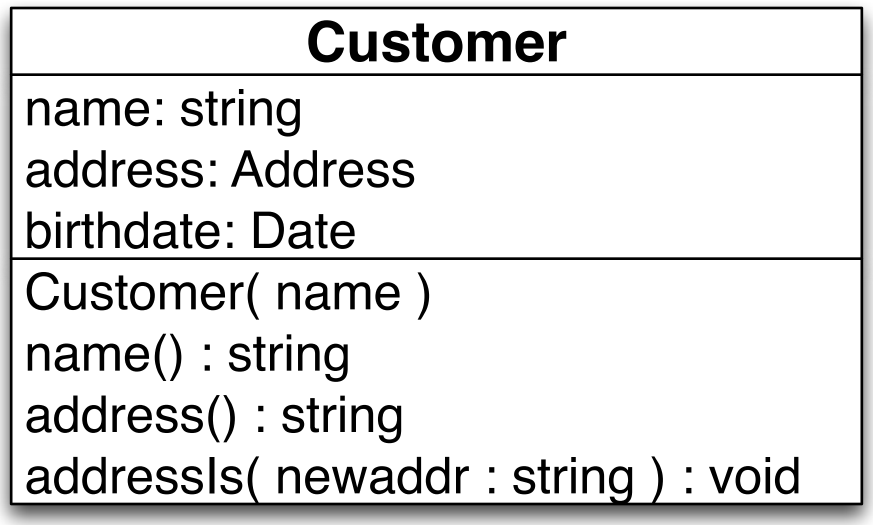
## Behaviour Diagrams

- Activity diagram
- Communication diagram
- Interaction overview diagram
- Sequence diagram
- State diagram
- Timing diagram
- Use case diagram

## UML Class Diagram Notation

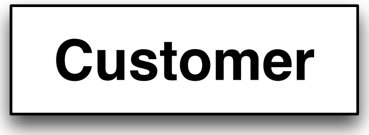
A box represents a class and defines

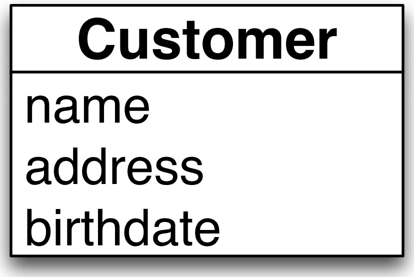
- class name
- set of attributes (data fields, types), initial values
- set of operations (routines, signatures)

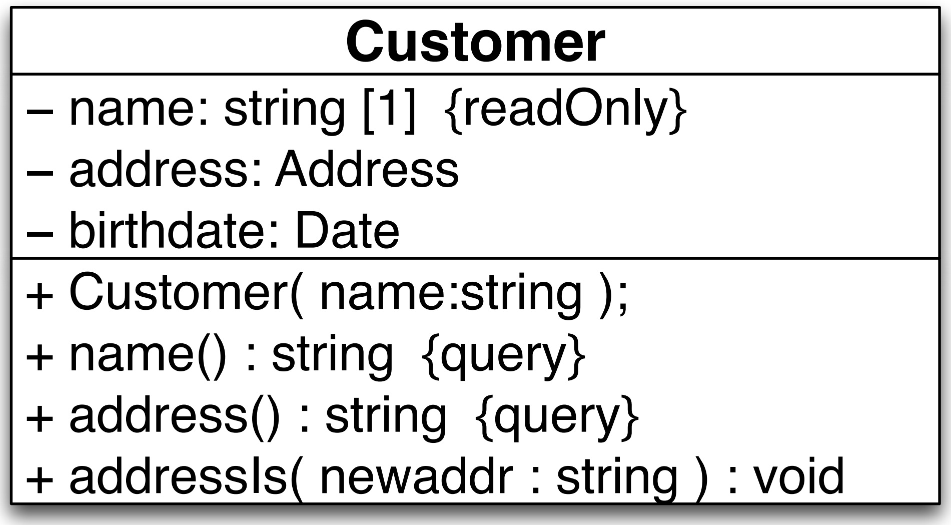


# Abstraction in Classes

Classes can be expressed at different levels of abstraction.



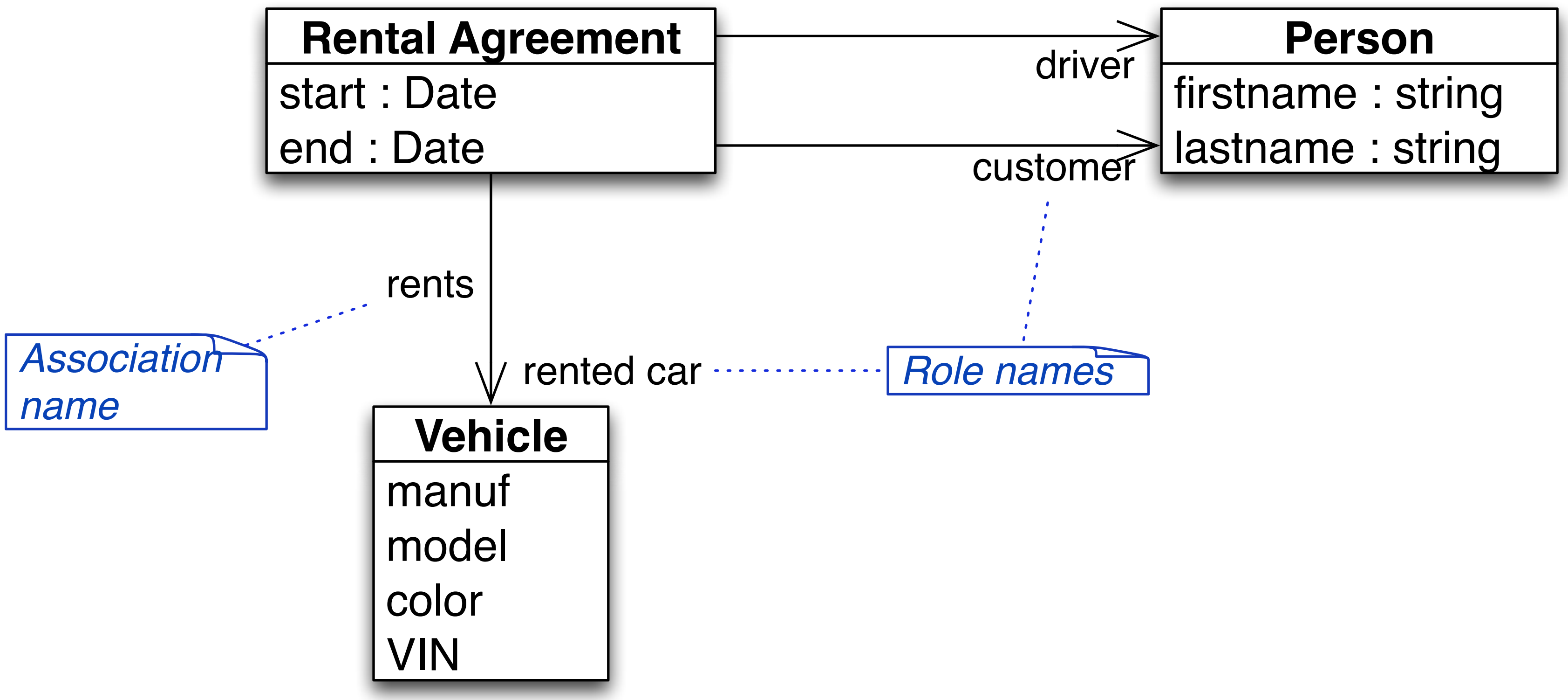



Customer
– name: string [1] {readOnly}
– address: Address
– birthdate: Date
+ Customer( name:string );
+ name() : string {query}
+ address() : string {query}
+ addressIs( newaddr : string ) : void

KEY:
- + public
- – private
- # protected
- <u>static</u>
- *pure virtual*

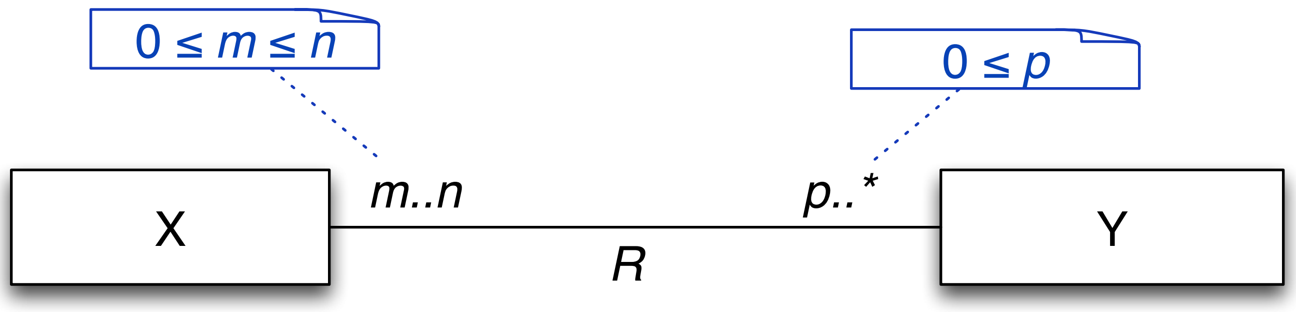
## Associations

An association between two classes indicates that there exists a physical or conceptual link between objects of those classes.



# Multiplicities

Multiplicity annotations constrain the number of allowable links in an association.

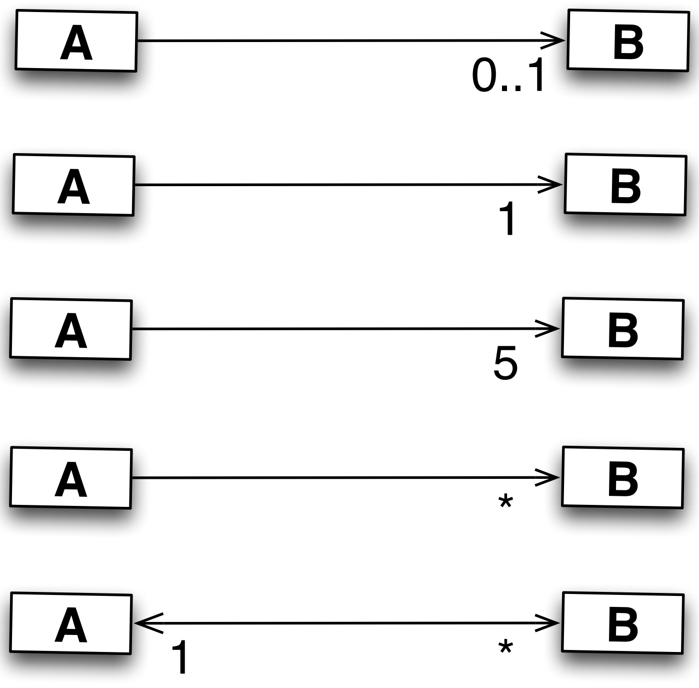


• For each object x of class X, there must be at least p links of association R linking x to object of class Y; and

• For each object y of class Y, there must be between m and n links of association R linking y to object of class X.

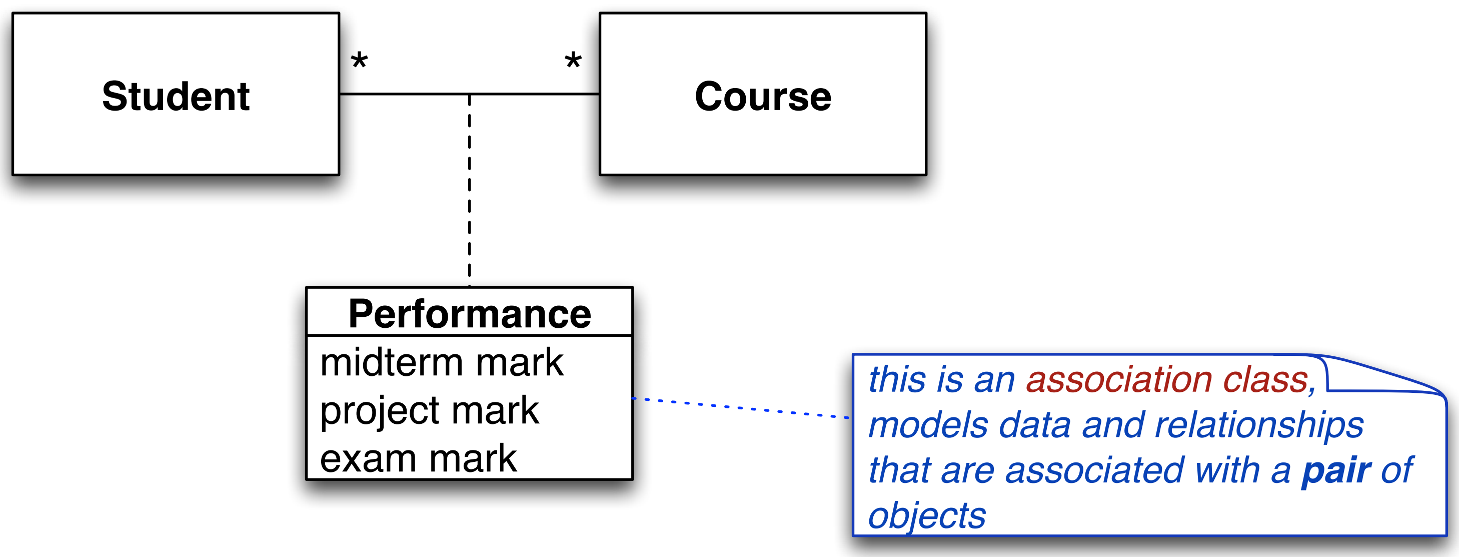
No annotation means that the multiplicity is unspecified.

## Implementing Associations

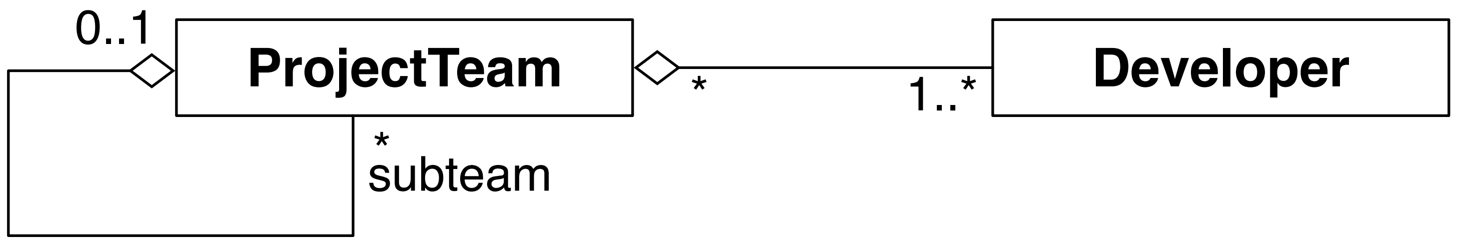


# Association Class

A class association represents link attributes
- properties of the link, because they cannot be attributed to either of the end objects


Student
*
*
Course
Performance
midterm mark
project mark
exam mark
this is an association class, models data and relationships that are associated with a **pair** of objects

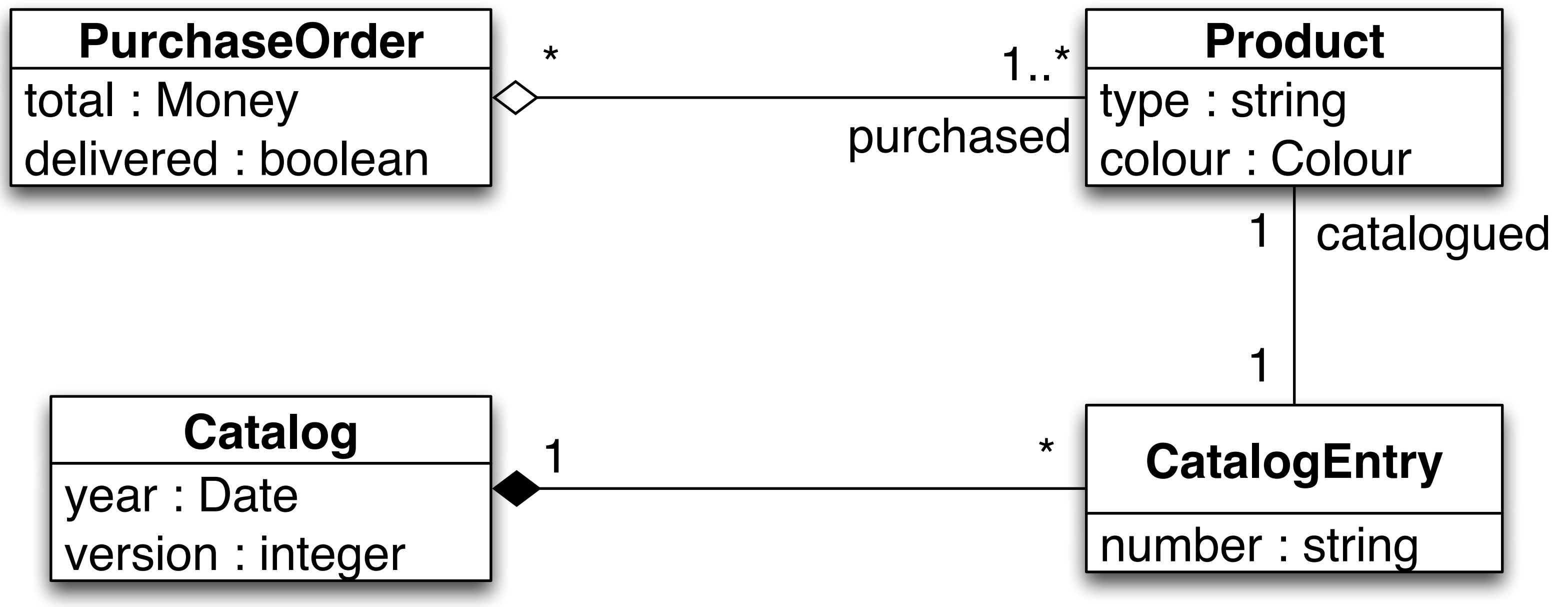
## Aggregation



Aggregation is a “part-of” relation between an aggregate (collection) and its members.

- part can be a member of more than one aggregate
  e.g., students can be members of more than one class roster
- part has an identity outside of the aggregate

## Composition



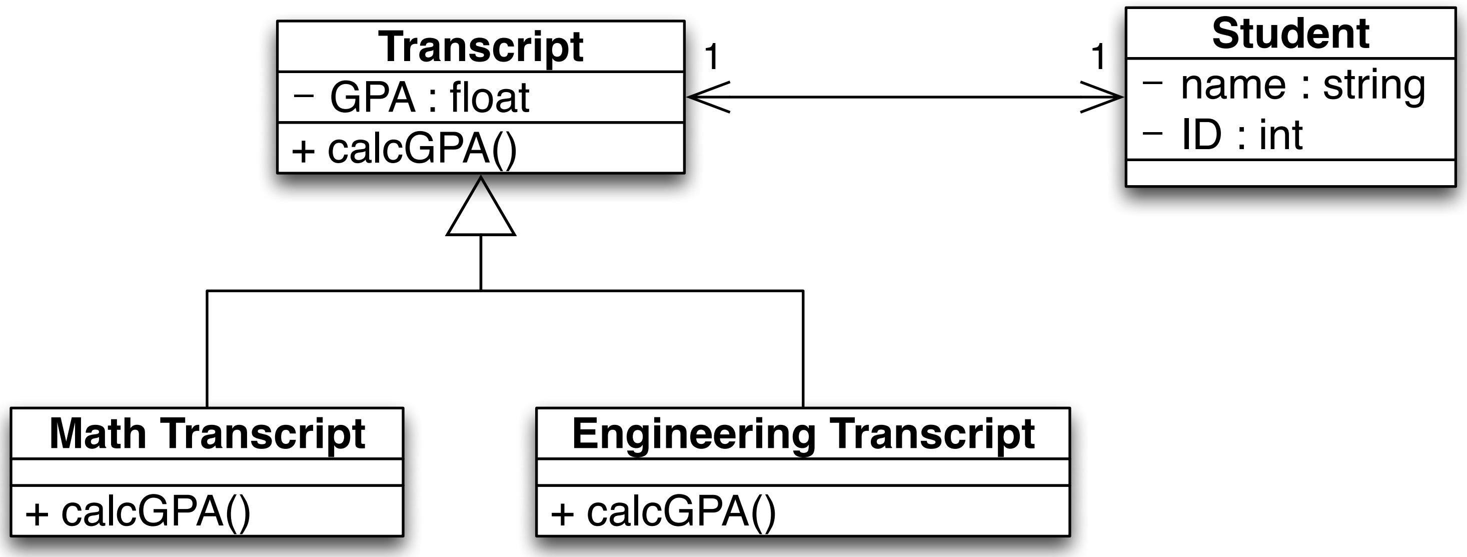
Composition is a stronger “part of” relation between a composite object and its components:

- a part does not exist without its composite
- a part belongs to at most one composite
- the composite is responsible for creating, destroying members

# Generalization

The UML uses the term generalization for the subtype relationship between a base class and its derived classes.

- Every member of a derived class is a member of its base class
- Attributes and associations of the base class are attributes and associations of the derived class



## From Assignment 1

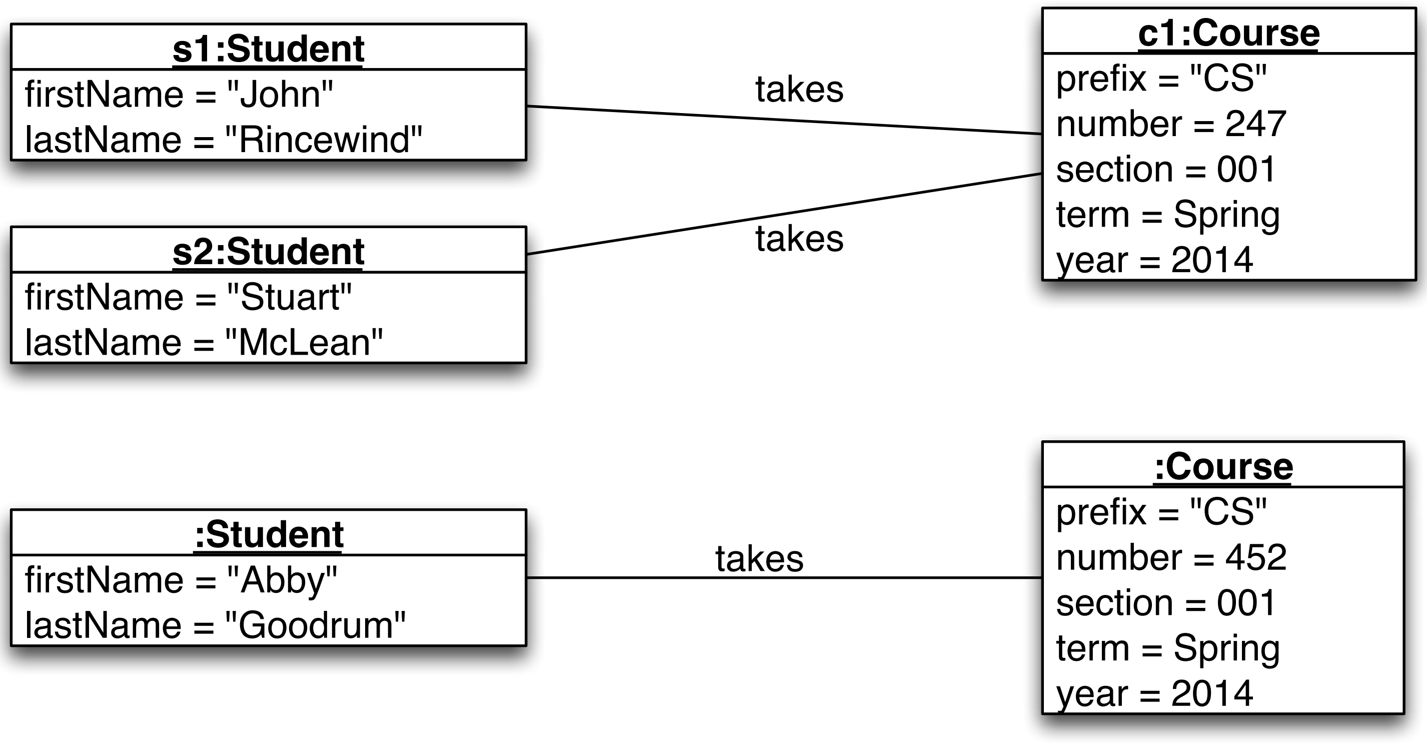
A cellphone account keeps track of information needed to bill a customer for their cellphone usage. There are two types of cellphone plans that a customer might subscribe to, and the plan type affects what information the account needs to keep track of:

- Cheap Plan: has a monthly service fee of $30 and gives the customer 200 free minutes of calls each month. If the customer makes more than 200 minutes of calls in a month, the customer is charged $1 for each minute over and above the 200 free minutes.
- Expensive Plan: has a monthly service fee of $100 and gives the customer unlimited free minutes of call time.

# UML Object Models

An object model is a run-time instance of a class model

- Every object is an instantiation of a specific class
- Every link between two objects is an instantiation of a specific association



## Snapshot of Execution

Object models visualize snapshots of a program's execution.

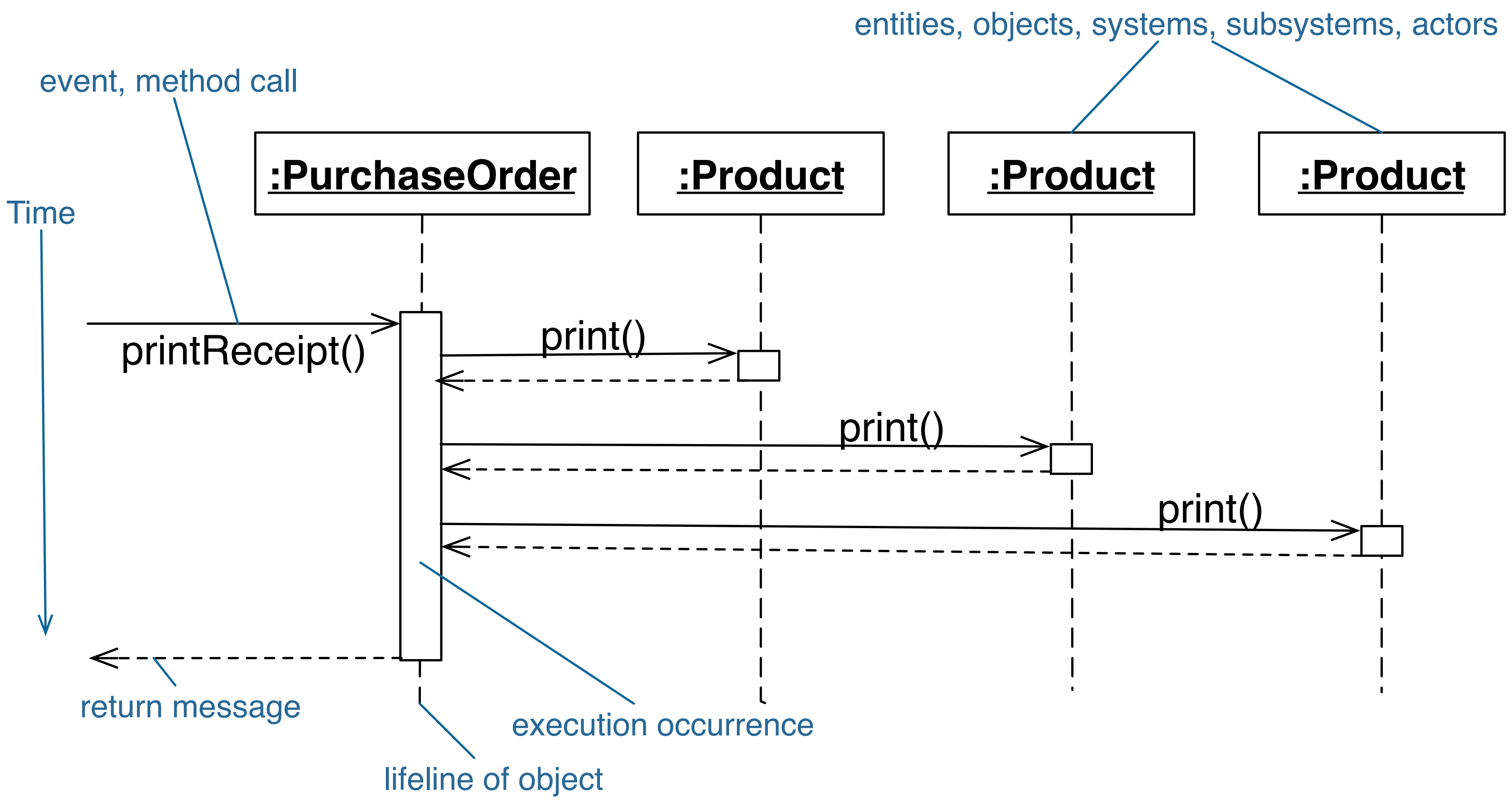
```
// C++ code.  Warning: contains disasters.

Circle *c1;
c1 = new Circle("Green");

if (pigsCanFly) {
    Circle *c2, *c3, *c4;
    c2 = new Circle("Red");
    c3 = new Circle("Blue");
    c1 = c2; // object model at this point in program
     ...
}
```

# UML Sequence Diagram Notation

A UML Sequence Diagram is a graphical model of communication events between objects, as exhibited in one execution trace.



## Some UML Drawing Tools

Can use any UML modelling or drawing tool that you would like
- must be able to output PDF files.

- Visio
- OmniGraffle (Mac only)
- UMLet (open source, Windows / OS X / Linux)
  http://www.umlet.com/