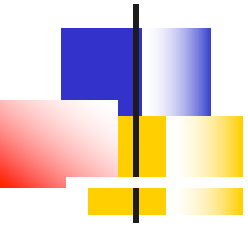
Information Systems Concepts

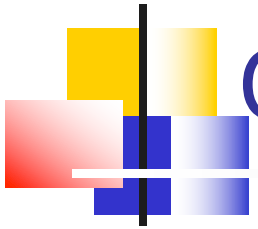
Requirements Analysis



Roman Kontchakov

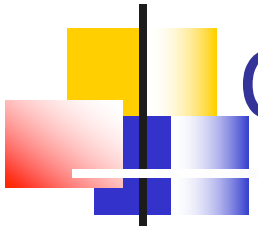
Birkbeck, University of London

Based on Chapter 7 of Bennett, McRobb and Farmer:
Object Oriented Systems Analysis and Design Using UML, (4th Edition), McGraw Hill, 2010



Outline

- Class Diagrams
  - Section 7.3 (pp. 184 – 194)
  - Section 7.5.5 – 7.5.10 (pp. 208 – 215)
  - Section 7.5.2 (pp. 198 – 201)

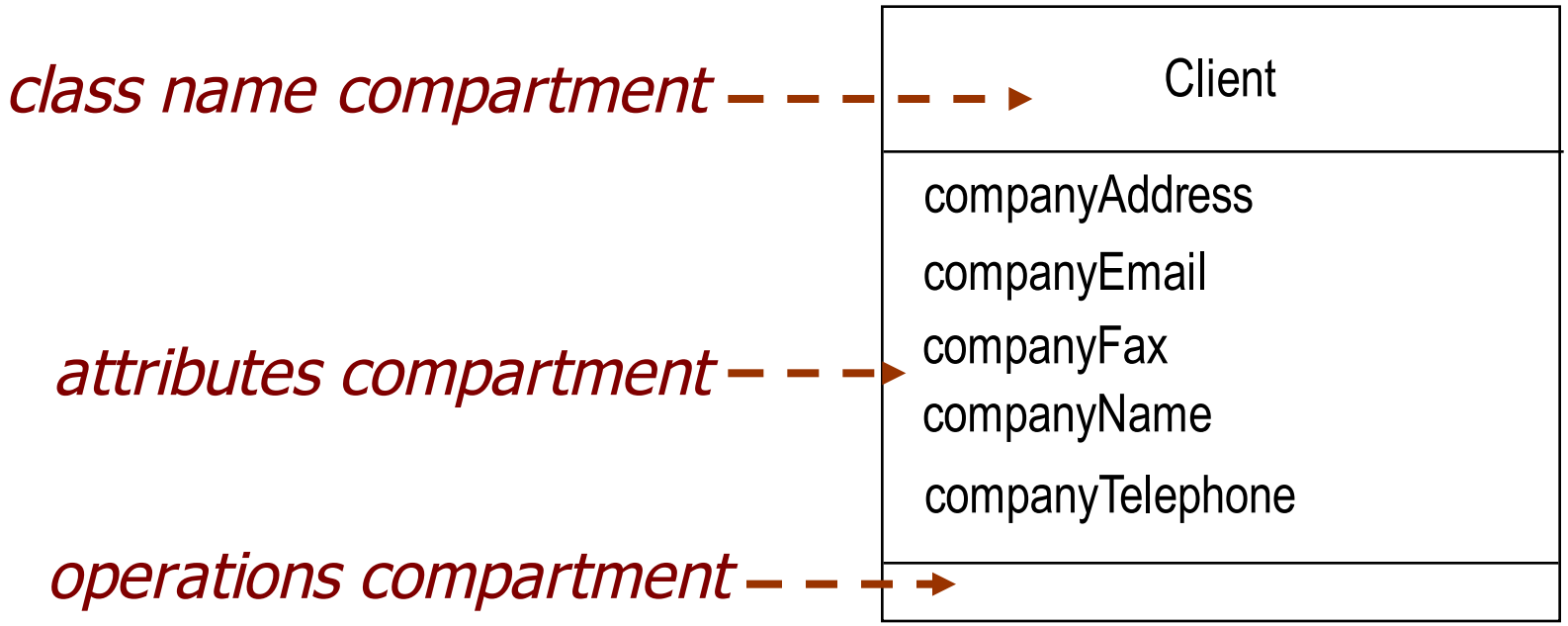


Class Diagrams

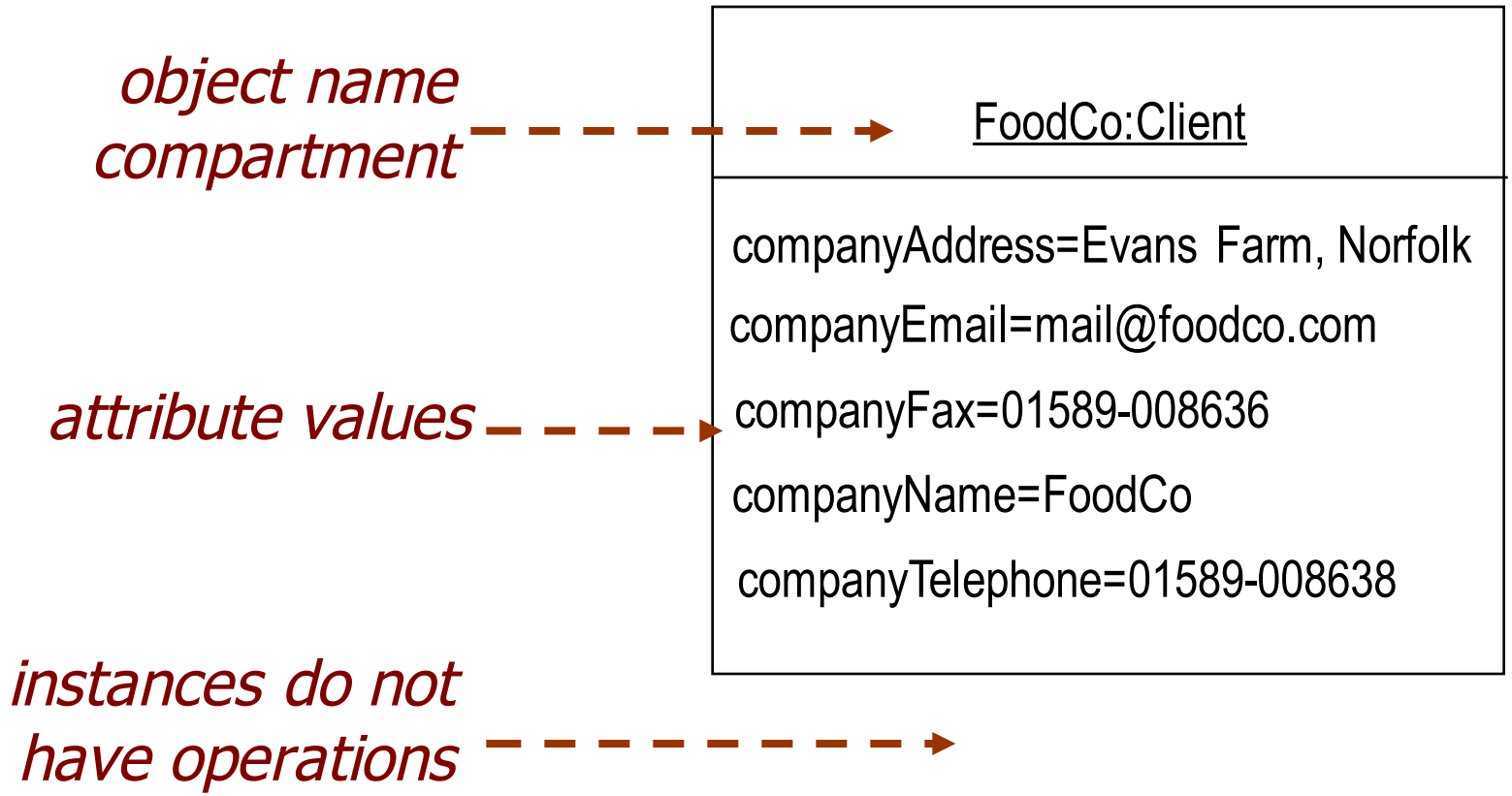
- Class (and Object/Instance)
- Stereotypes
- Attributes (and State)
- Associations (and Links)
- Multiplicity
- Operations

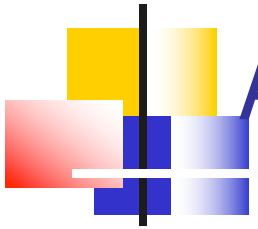
Symbols of *Instances, States, and Links* are used in other UML diagram types
(Object Diagrams, Communication Diagrams, etc.)

Notation: Class



Notation: Object/Instance

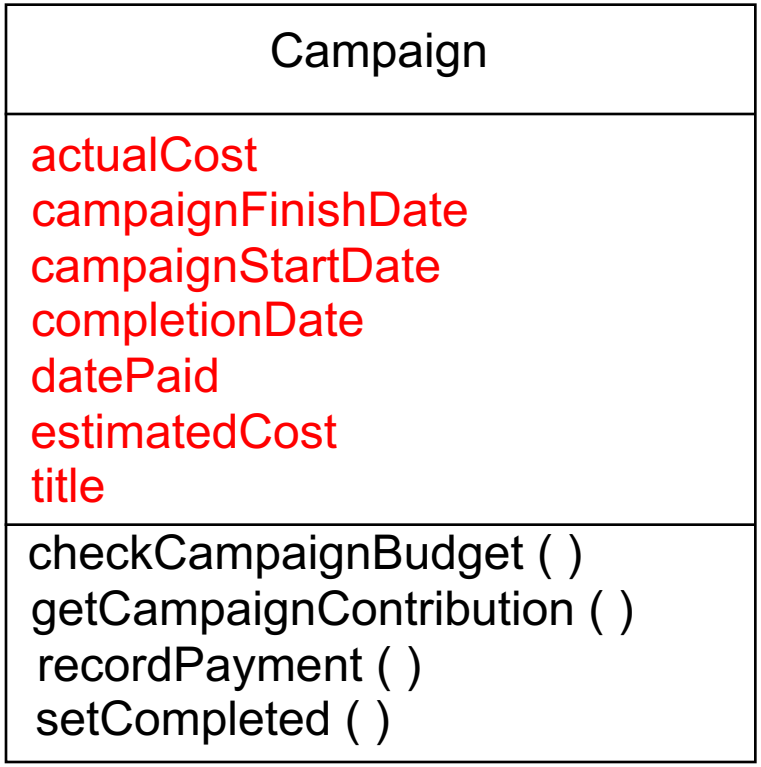




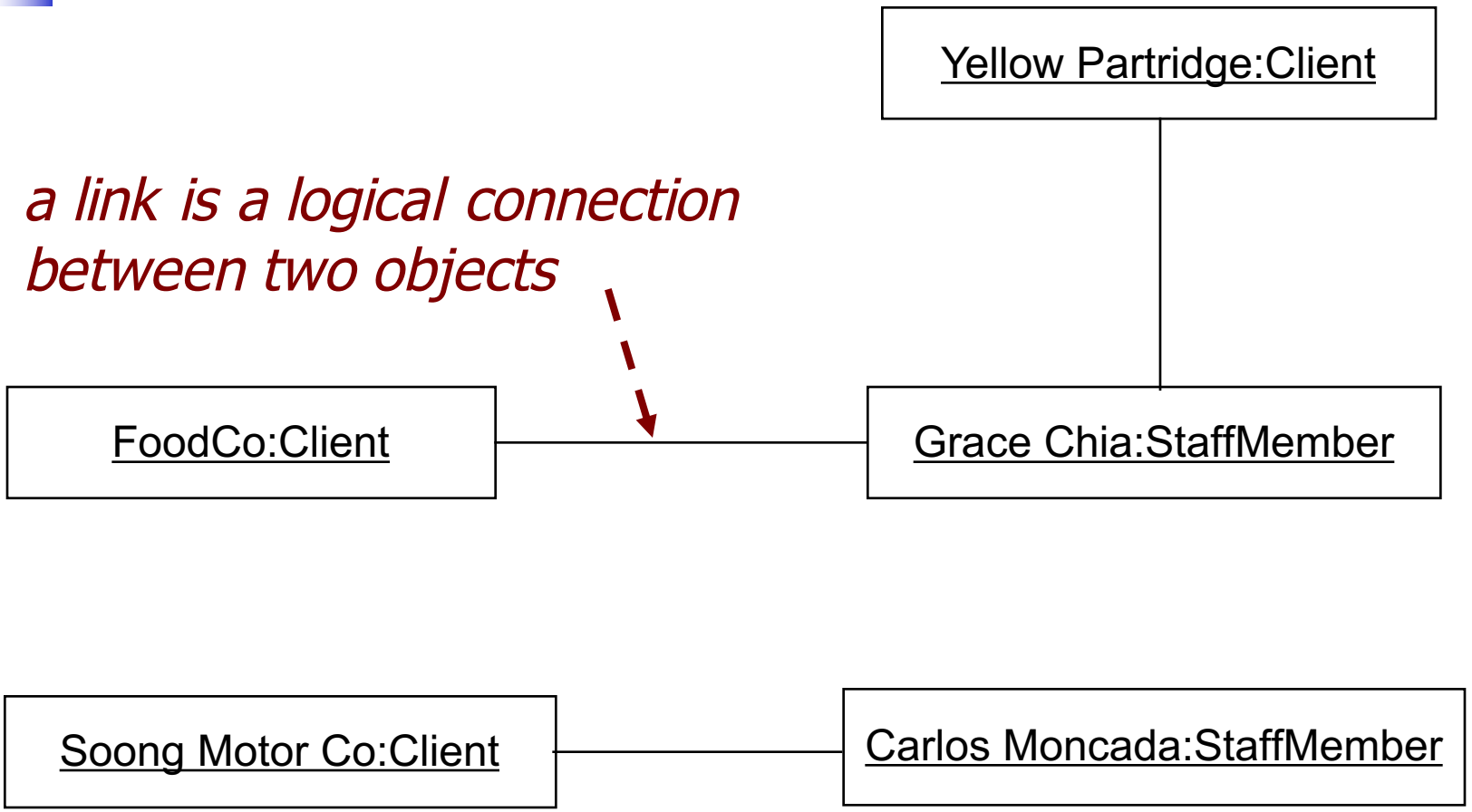
Attributes

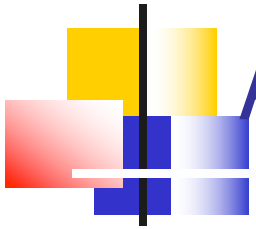
- Attributes are:
  - part of the essential description of a class
  - the common structure of what all objects of the class can 'know'
    - each object has its own value for each attribute of its class
    - attribute values characterize state of the object

Notation: Attributes



Links

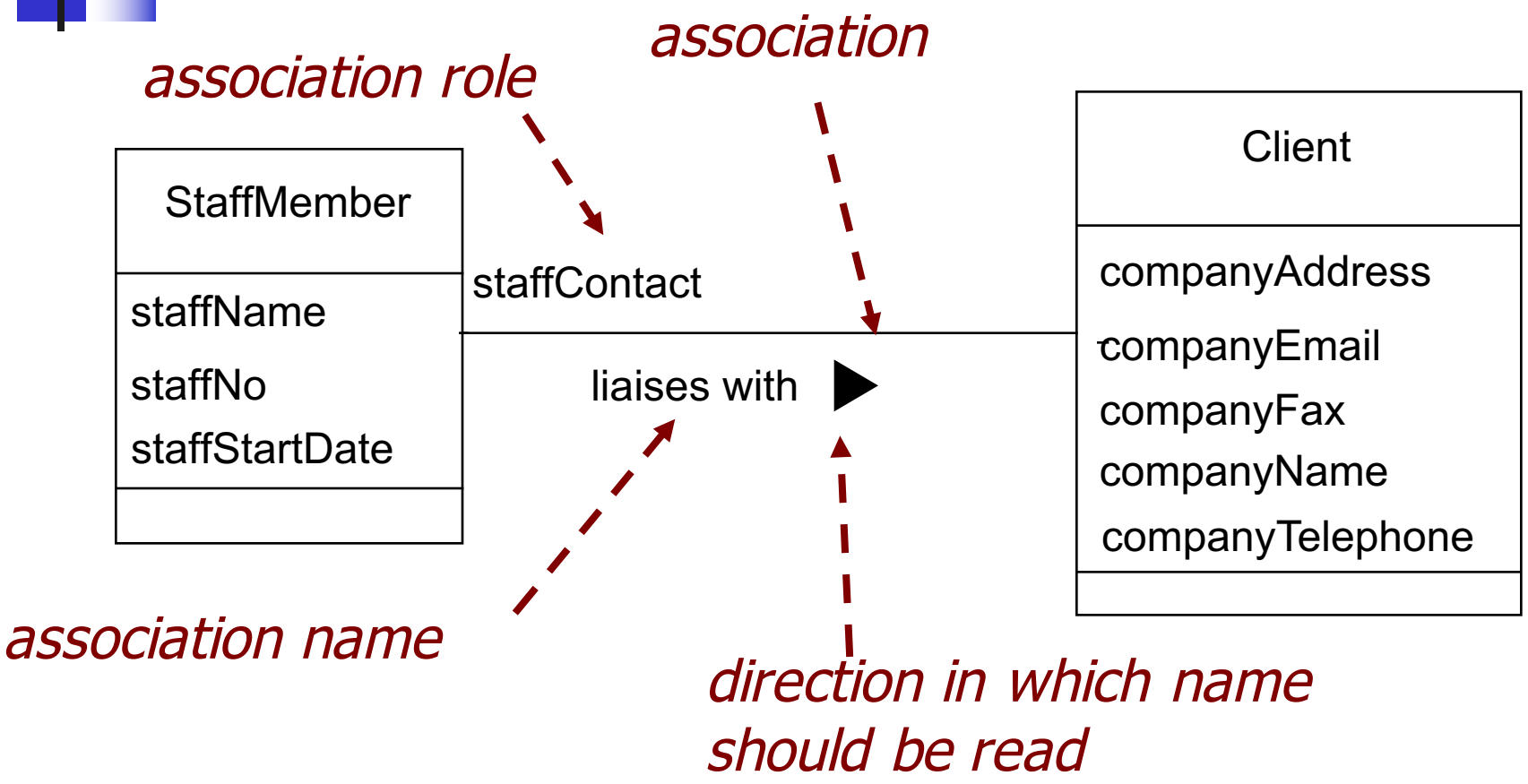

Yellow Partridge:Client
a link is a logical connection between two objects
FoodCo:Client
Grace Chia:StaffMember
Soong Motor Co:Client
Carlos Moncada:StaffMember



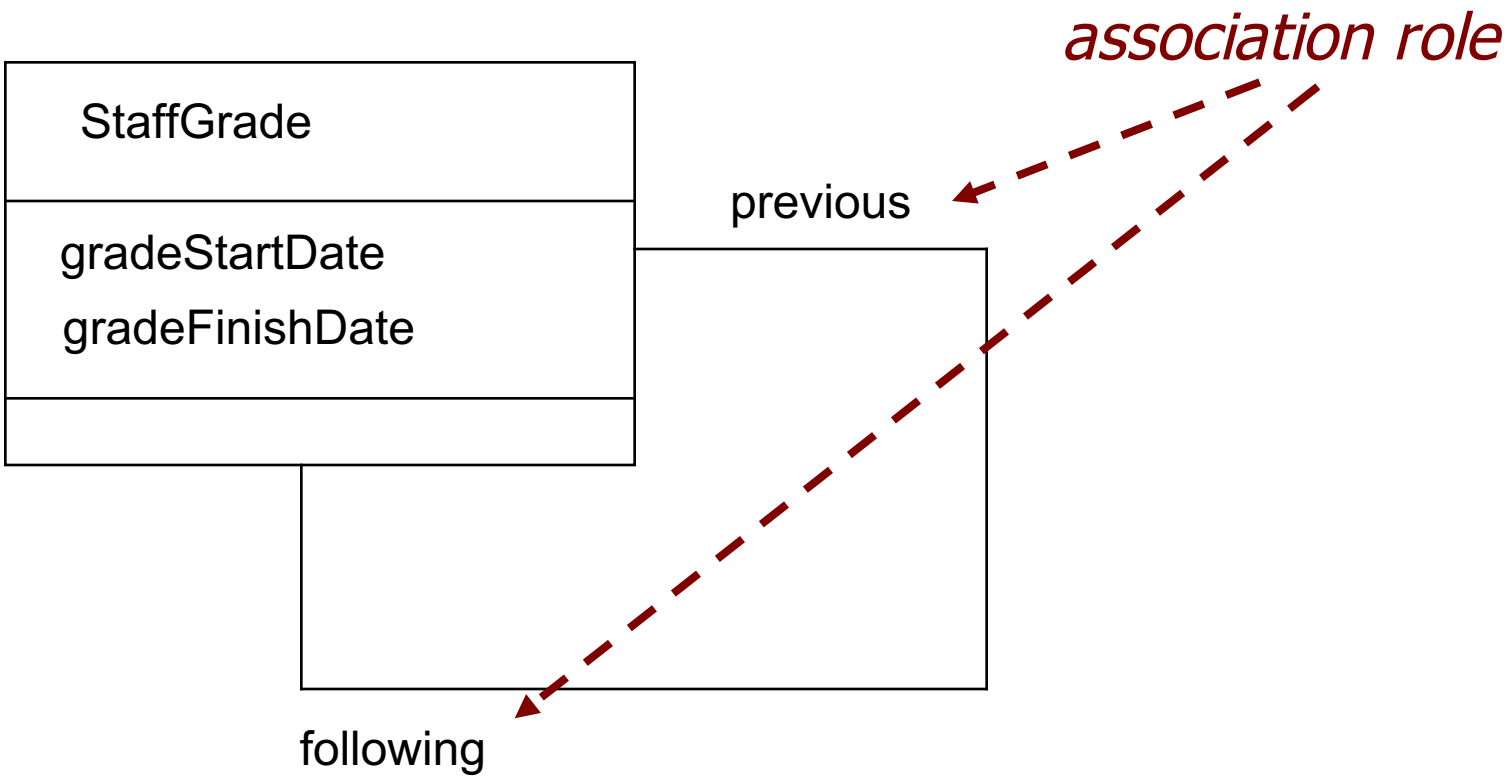
Associations

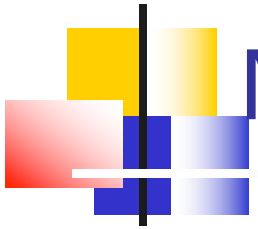
- An associations represent the *possibility* of a logical relationship or connection between objects of one class and objects of another
  - If two objects can be linked, their classes have an association

Notation: Associations



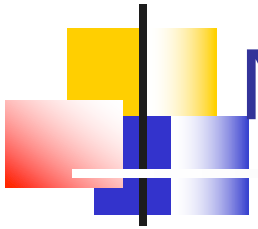
Notation: Associations





Multiplicity

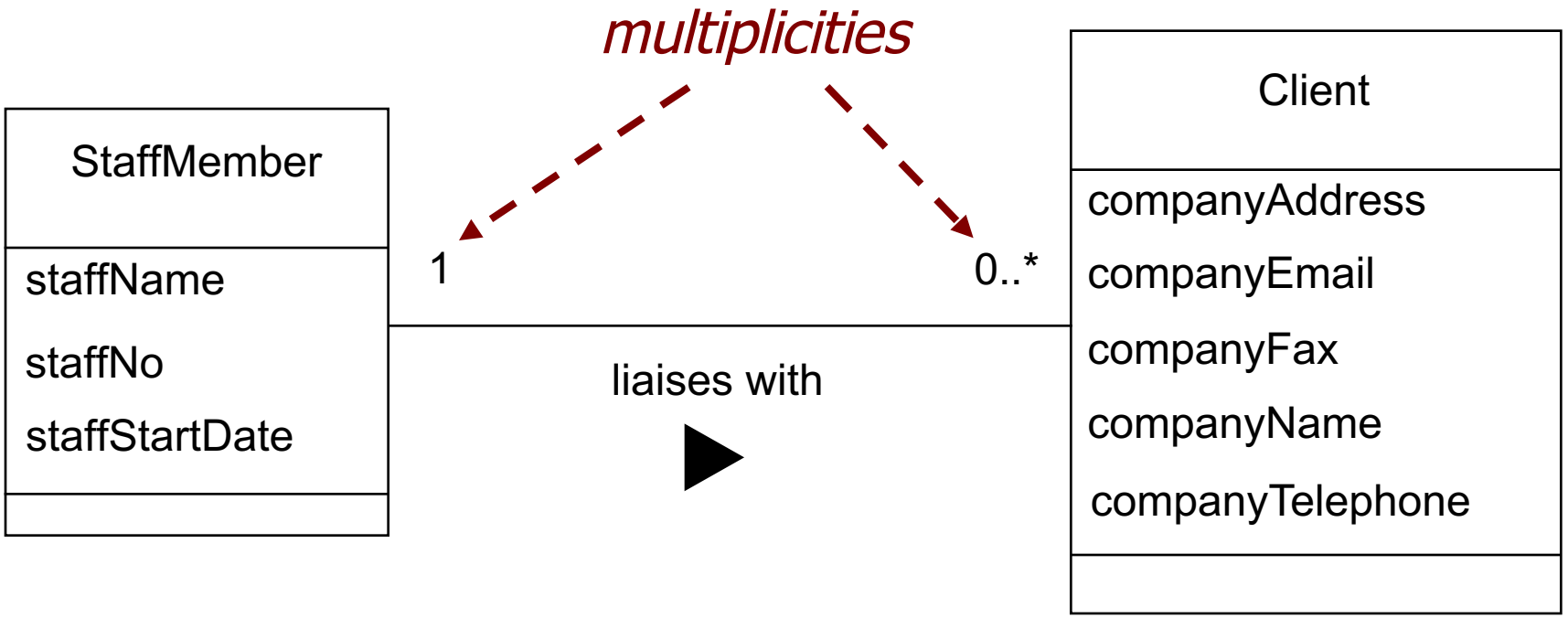
- The multiplicity of an association is the range of permitted *cardinalities* of its participating objects, according to or business rules.
  - for example:
    - any bank customer may have one or more accounts
    - every individual account is for one, and only one, customer



Notation: Multiplicity

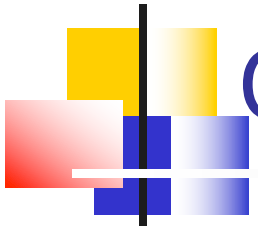
- ***n***: exactly n
- *: any number
- ***m..n***: any number in the range m to n (inclusive)
  - **0..1**: optional (i.e., either none or 1)
  - **1..***: at least one

Notation: Multiplicity



A staff member may liaise with any number of clients (including 0)
Each client is liaised with exactly one staff member

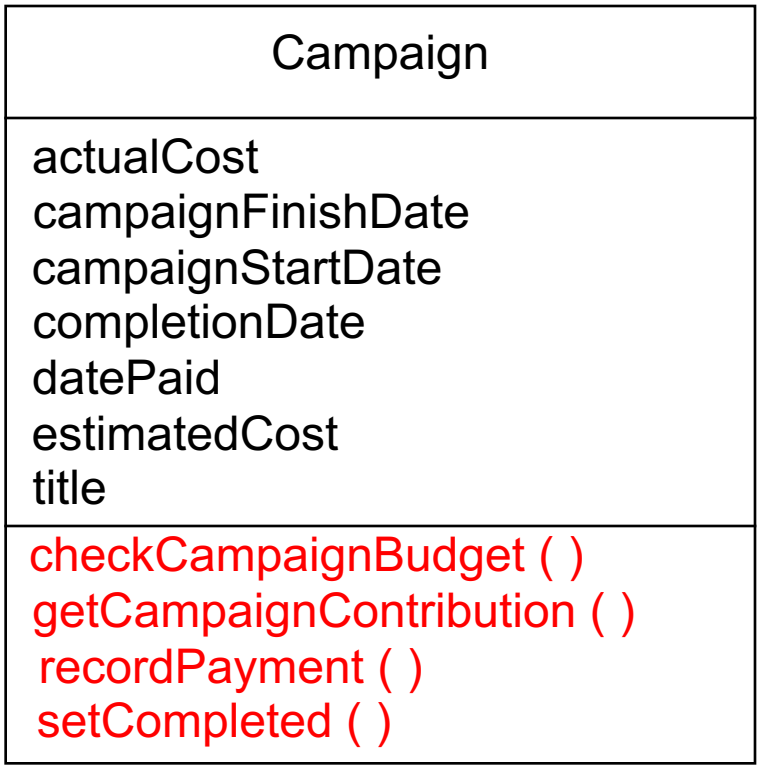
See Also: Figure 7.9-7.11 (p. 191)

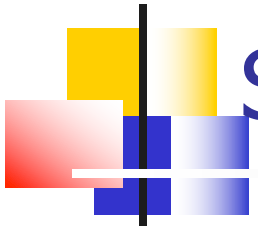


Operations

- Operations are:
  - part of the essential description of a class
  - the common behaviour that all objects of the class can 'do'
    - get or set attribute values (not specified in Analysis Model)
    - perform calculations
    - send messages to other objects
    - create or destroy links (not specified in Analysis Model)
  - services that objects of a class can provide to other objects

Notation: Operations





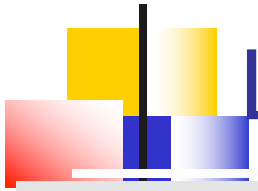
Static Analysis with UML

- Requirements Model

  → Analysis Model

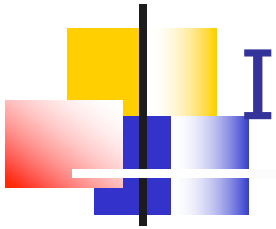
  → Design Model

- To Draw an Analysis Class Diagram
  - Identify Classes
  - Determine Stereotypes
  - Find and Locate Attributes
  - Add Associations
  - Determine Multiplicity
  - Find and Locate Operations



Looking for Potential Classes

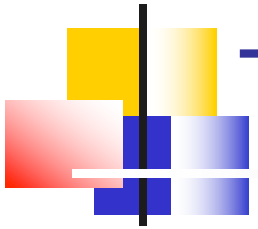
Category	Examples
People	Mr Harmsworth (a campaign manager), Dilip (a copywriter).
Organisations	Jones & Co (a forklift truck distributor), the Soong Motor Company, Agate's Creative Department.
Structures	Team, project, campaign, assembly.
Physical things	Fork-lift truck, electric drill, tube of toothpaste.
Abstractions of people	Employee, supervisor, customer, client.
Abstractions of physical things	Wheeled vehicle, hand tool, retail goods.
Conceptual things	Campaign, employee, rule, team, project, customer, qualification.
Enduring relationships between members of other categories.	Sale, purchase, contract, campaign, agreement, assembly, employment.



Identifying Classes

- Should *this* really be considered as a class?
  - Is it beyond the scope of the system?
  - Does it refer to the system as a whole?
  - Does it duplicate another class?
  - Is it too vague?
  - Is it too specific?
  - Is it too tied up with physical inputs and outputs?
  - Is it really an attribute?
  - Is it really an operation?
  - Is it really an association?

If any answer is 'Yes', consider modelling the potential class in some other way (or do not model it at all).



Take Home Messages

- Class Diagrams
  - Class (and Object/Instance)
  - Attributes (and State)
  - Associations (and Links)
  - Multiplicity
  - Operations