



Class Model

Class Model - Goal

- Captures **static structure** of a system
- System is built around **objects** and **classes** rather than functionality – **Essence or OOP**
- Most important model in the three models
- **Describe relationship** between **objects**, **identify attributes** and **operations** for classes

Class Model - Objects

- What is Object?
 - “Concept, identity or a thing with identity”
- How to identify?
 - Appear as proper nouns in problem descriptions
- “All objects are distinguishable”
 - Two apples with same color, shape and texture are individual apples

Class Model: Classes

- What is Class?
 - A class describes a group of objects with
 - the same properties (attributes),
 - behavior (operations).
- Examples:
 - Person, Company, Process
- How to identify?
 - Appear as common nouns in problem descriptions

Why Classes, not Objects?

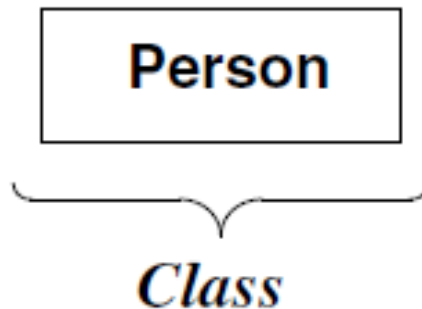
- Main Reason:
 - Abstraction
- Benefits:
 - Common things are stored only once (Per Class, not per Object)
 - Code Reuse (Same methods are shared by all objects)

Class Diagrams

- “Provide **generic notations** for modeling **classes** and their **relationships**”
- Can be used for
 - Abstract Modeling
 - Actual Program Design

Class Diagrams: Rules

- The **UML symbol** for a class also is a **box**.
- The **convention** is to list the class name in boldface, center the name in the box, and capitalize the first letter.
- **Singular nouns** are used for the names of classes.

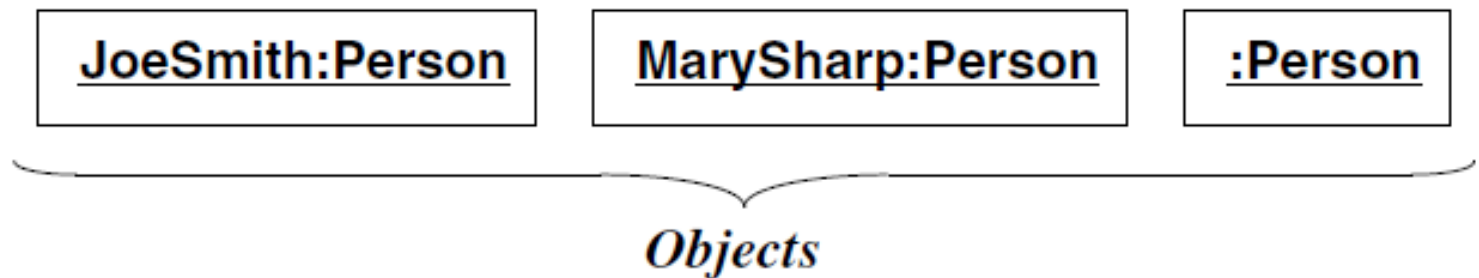


Object Diagrams

- “Represent individual objects and their relationships”
- Can be used for
 - Documenting Test Cases
 - Discussing Examples
- “Class Diagram corresponds to infinite set of object diagrams”

Object Diagrams

- Popular convention used for referring to objects is separate multiword name with intervening capital letters.



Values and Attributes

- “A **value** is a piece of data”.
 - It can be found by examining problem documentation for **examples**.
- “An **attribute** is a named property of a class that describes a value held by each object of the class”.
 - Attributes can be found by looking for **adjectives**.
 - E.g. Person object has name, birthdate and weight as attributes

Attribute: Rules

- The **UML notation** lists attributes in the **second compartment** of the **class box**.
- Optional details, such as **type** and **default value**, may follow each attribute.
- A **colon** precedes the **type**.
- An **equal sign** precedes the **default value**.
- General convention is to show the attribute name in regular face, left align the name in the box, and use a lowercase letter for the first letter.

Attribute: Rules

- Attribute values can be mentioned in the second compartment of object boxes.
- The notation is to list each attribute name followed by an equal sign and the value.
- General convention is to left align attribute values and use regular type face.

Values and Attributes

Person
name: string birthdate: date

Class with Attributes

<u>JoeSmith:Person</u>
name="Joe Smith" birthdate=21 October 1983

<u>MarySharp:Person</u>
name="Mary Sharp" birthdate=16 March 1950

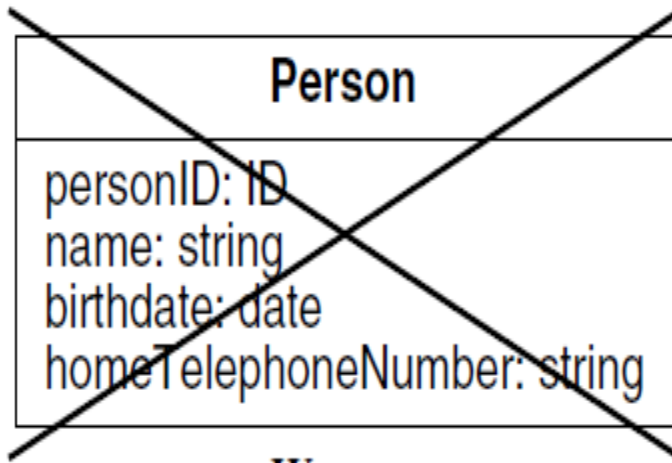
Objects with Values

Difference: Value and Object

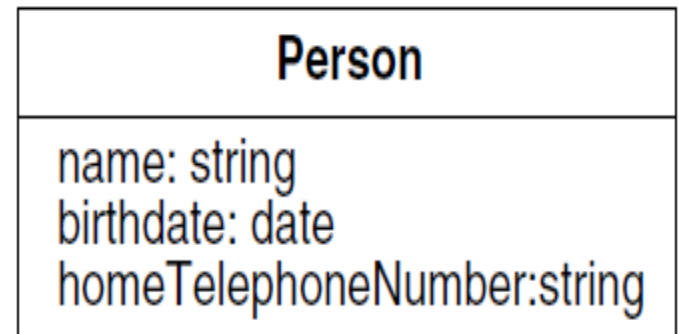
- “Values lack identity”
 - All occurrences of “India” are not distinguishable
- “Objects have identity”
 - Can be distinguished from others

Object Identifiers

- Object identifiers are **implicit** in a class model
- Most OO languages automatically generate identifiers with which to reference objects.



Wrong



Correct

Operation and Method

- **Operation:**
- “An operation is a function or procedure that may be applied to or by objects in a class.”
- All objects in a class share the same operations.
- Same operation may be applied to many different classes – Polymorphic
- **Method:**
- “Implementation of an operation for a class”
- E.g. For a class “File”, “Print” Operation has “Print Binary”, “Print ASCII” as Methods

Operation: Rules

- The **UML notation** is to list **operations** in the **third compartment** of the **class box**.
- General convention is to list the operation name in regular face, left align the name in the box, and use a lowercase letter for the first letter.
- Optional details, such as an **argument list** and **result type**, may follow each operation name.
- **Parentheses** enclose an argument list; commas separate the arguments.

Operation: Rules

- A colon precedes the result type. An empty argument list in parentheses shows explicitly that there are no arguments
- Operations are not Listed for Objects – Why?

Person
name birthdate
changeJob changeAddress

File
fileName sizeInBytes lastUpdate
print

GeometricObject
color position
move (delta : Vector) select (p : Point): Boolean rotate (in angle : float = 0.0)

Summary: Class Notations

- The attribute and operation compartments of class boxes are optional.

ClassName
attributeName1 : dataType1 = defaultValue1 attributeName2 : dataType2 = defaultValue2 • • •
operationName1 (argumentList1) : resultType1 operationName2 (argumentList2) : resultType2 • • •

direction argumentName : type = defaultValue

Class Model: Relationships

- Links and Associations
- Generalization
- Aggregation
- Composition

Links

- “A **link** is a physical or conceptual connection among objects.”
- For example, Joe Smith “WorksFor” Simplex company.
- “**Most links relate two objects**”
 - Links relating three or more objects are also possible
- “**Link is a tuple – List of objects**”
- “**A link is an instance of an association.**”

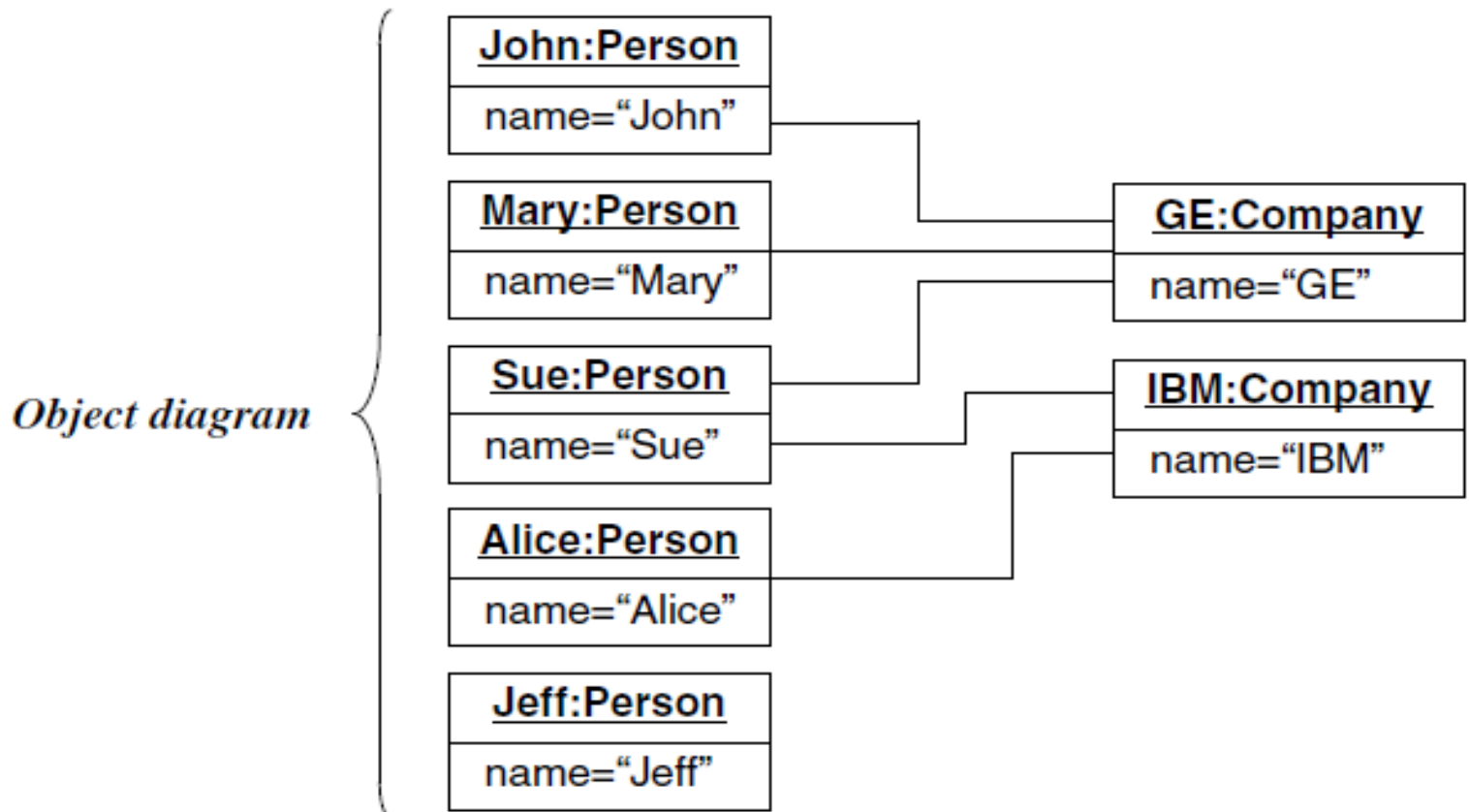
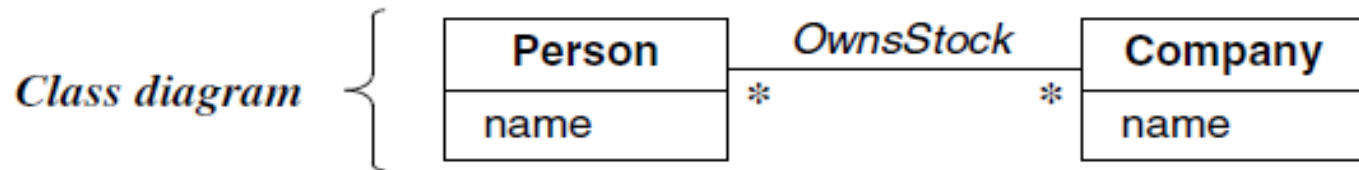
Associations

- “An **association** is a description of a group of links with common structure and common semantics.”
- For example, a person “WorksFor” a company.
- “**Associations are inherently bidirectional.**”
- **Links and associations** often appear as **verbs** in problem statements.

Links and Associations : Rules

- The UML notation for a link is a line between objects.
- An association connects related classes and is also denoted by a line.
- It is good to arrange the classes in an association to read from left-to-right, if possible.

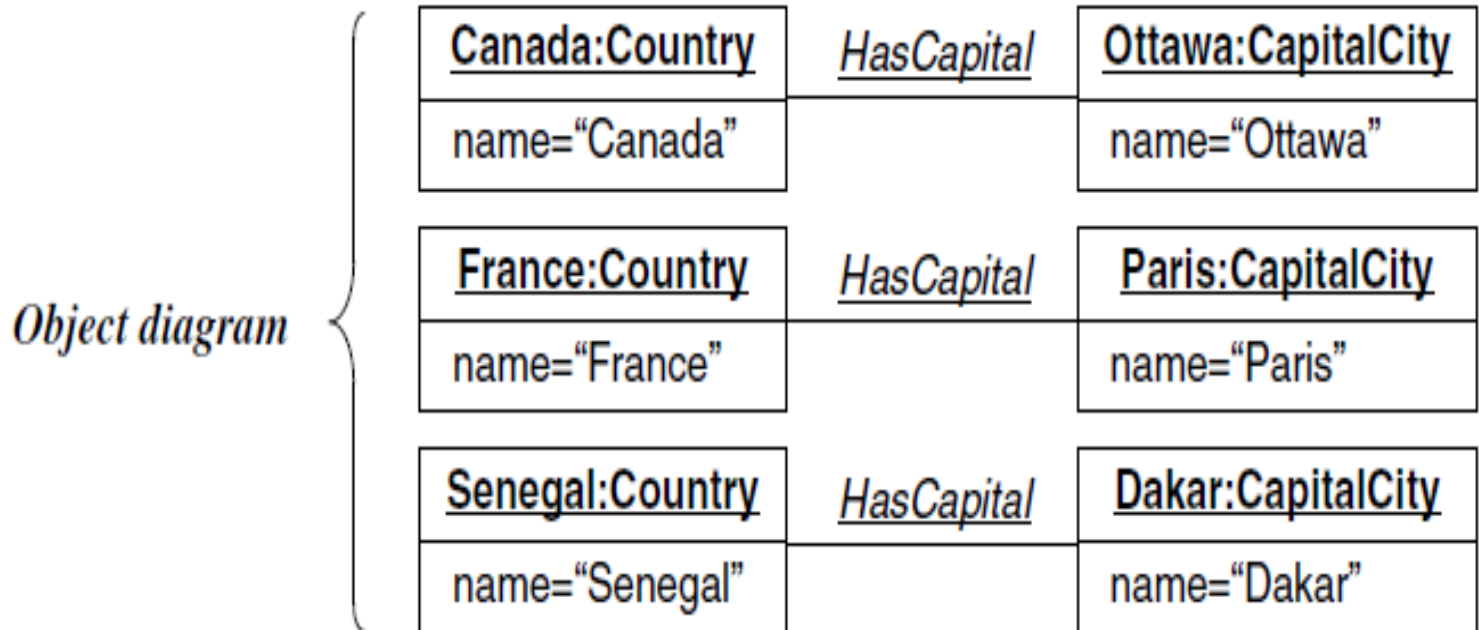
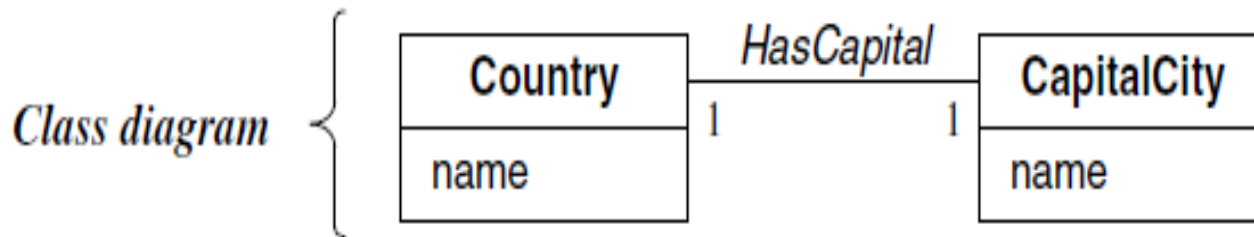
Links and Associations



Associations: Multiplicity

- Multiplicity specifies the number of instances of one class that may relate to a single instance of an associated class.
- UML diagrams explicitly list multiplicity at the ends of association lines.
- The UML specifies multiplicity with an interval, such as “1” (exactly one), “1..*” (one or more), or “3..5” (three to five, inclusive).

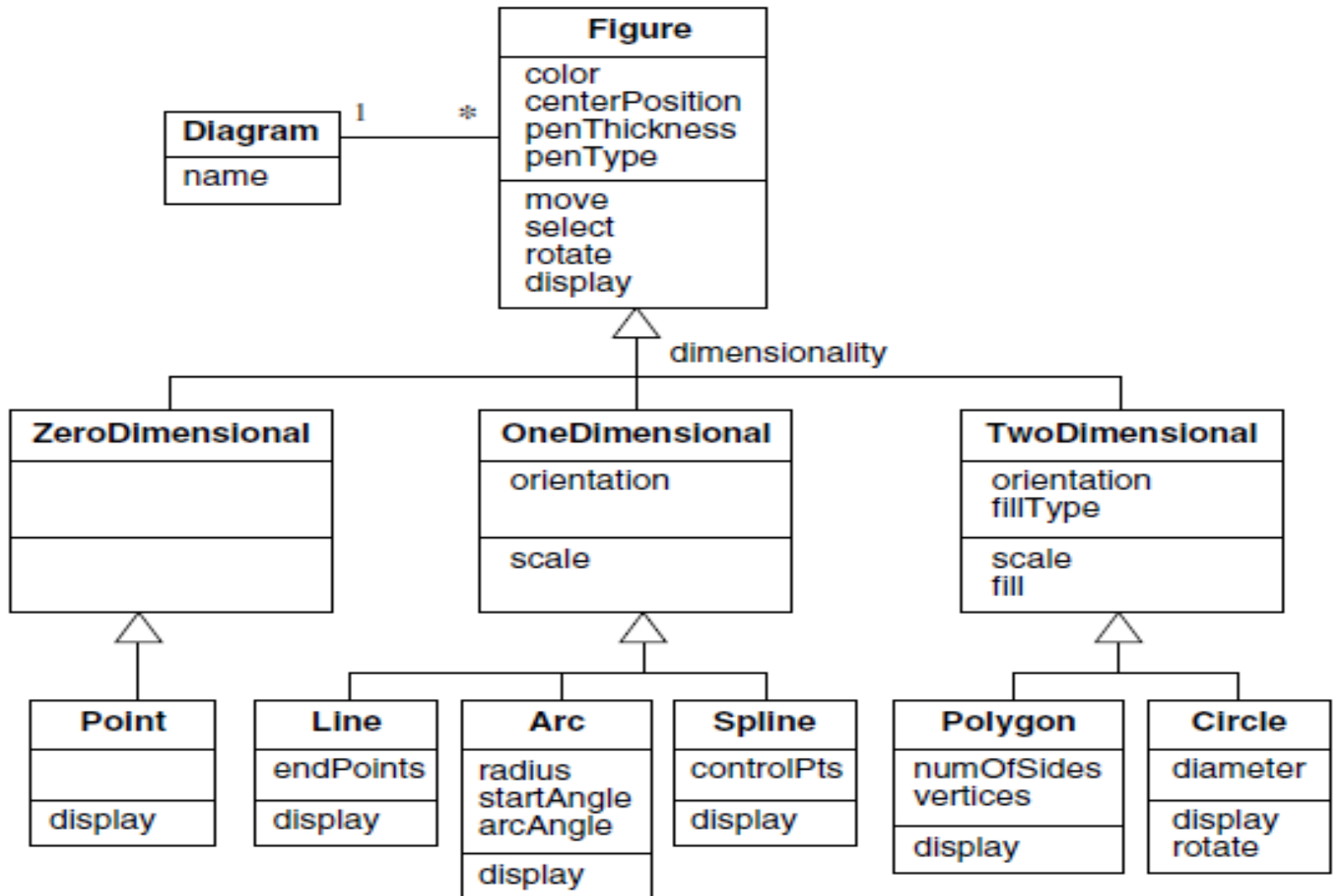
Associations: Multiplicity



Generalization

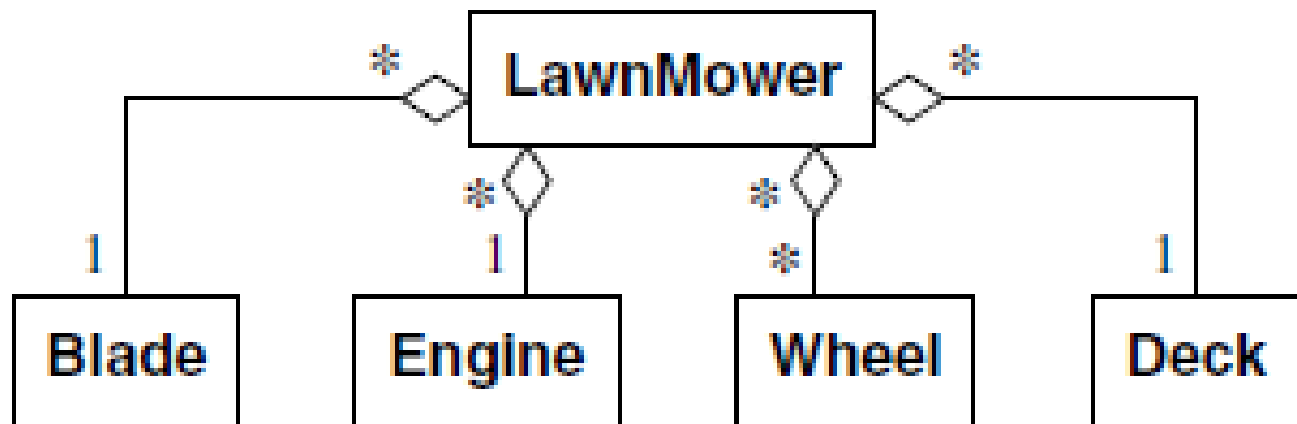
- “Generalization is the relationship between a class(super class) and one or more variations of the class(sub classes)”.
- Known as “is a” relationship
 - Each instance of subclass is an instance of super class as well
 - E.g. “Diagram is a figure”

Generalization



Aggregation

- “Aggregation is a strong form of association in which an aggregate object is made of constituent parts.”
- Constituents are part of the aggregate.



Aggregation - Properties

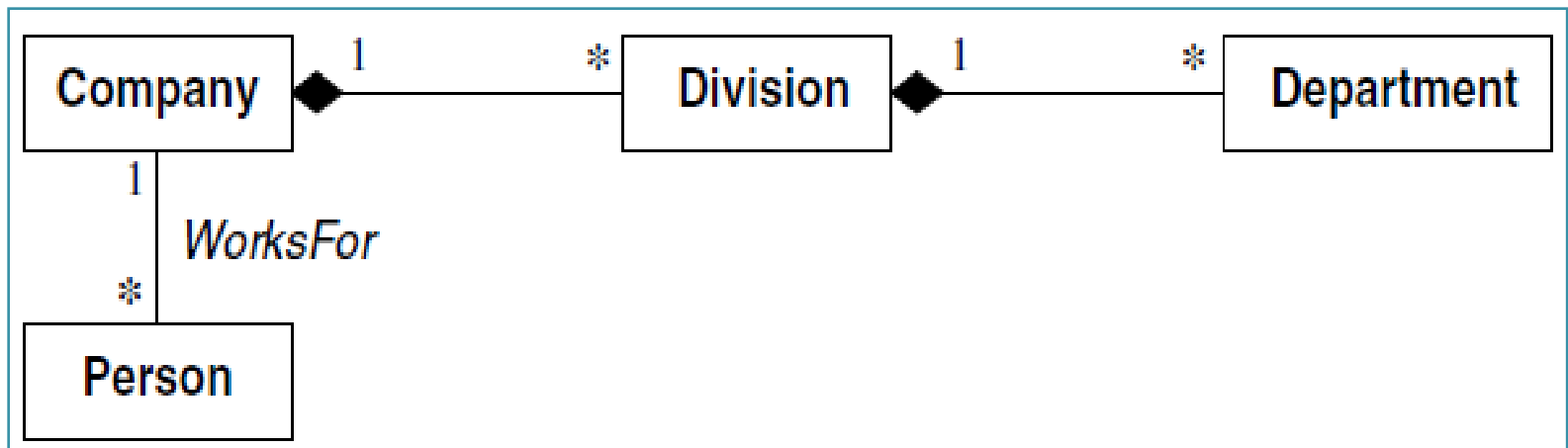
- Aggregation is transitive
 - That is, if A is part of B and B is part of C, then A is part of C.
 - E.g. Line is part of Paragraph, Paragraph is part of Document, then Line is part of Document
- Aggregation is anti-symmetric
 - A is part of B, B is not part of A
 - Engine is part of Car, Car is not part of Engine
- Aggregation is part-whole relationship

Aggregation vs. Association

- Aggregation is a special form of association, not an independent concept.
- If two **objects** are tightly bound by a **part-whole relationship**, it is an **aggregation**.
- If the two **objects** are usually considered as **independent**, even though they may often be linked, it is an **association**.

Composition

- **Composition** is more **restrictive** form of aggregation
 - Part can belong to at most one whole
 - Part has coincident lifetime with whole



Time for Questions!

- Identify the type of relationships based on following:
- “Manager is a type of Employee.”
- “Manager has a swipe card to enter company premises.”
- “Manager has many workers working under him.”
- “Company has many departments.”

Class Diagram Exercises

- Hospital Management System
 - Hospital, Department, Patient, Staff, Administration Staff, Hospital Staff, Technical Staff, Doctor, Nurse...
- College Management System
 - College, Department, Staff, Library, Student, Faculty, Subject, Result...
- Online Shopping System
 - Order, Customer, Account, Cart, Product...
- Try to draw complete class diagrams for the above mentioned systems.