DA-IICT

UNIFIED
MODELING
LANGUAGE

IT313: Software Engineering

Advanced Class Modeling
Identification of Objects, Classes and their Relationships:
Exercises

Saurabh Tiwari

1

Enumerations
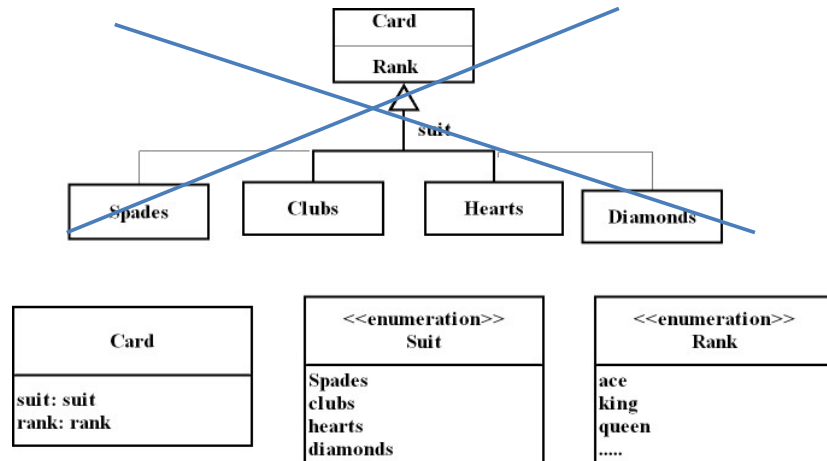
- A data type is a description of values.

- An enumeration is a data type that has a finite set of values.
  - For example, accessPermission (read, write, read-write), colors

Figure.penType

TwoDimensional.fillType

Modeling Enumerations

Card

Rank

suit

| Spades | Clubs | Hearts | Diamonds |

Card
suit: suit
rank: rank

<<enumeration>> **Suit**
Spades clubs hearts diamonds

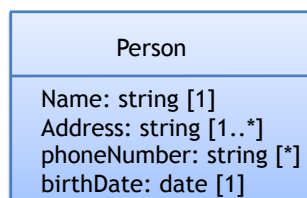| <<enumeration>> **Rank** |
|---|
| ace king queen ..... |

Do not use a generalization to capture the values of an enumerated attribute
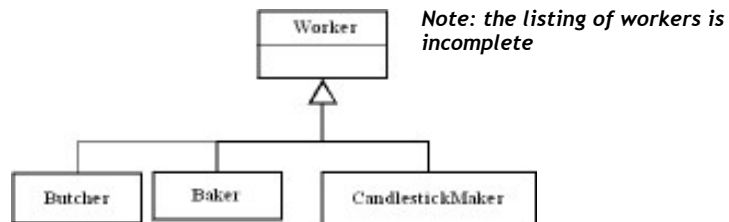

Multiplicity for an attribute

Specifies the number of possible values for each instantiation of an attribute.

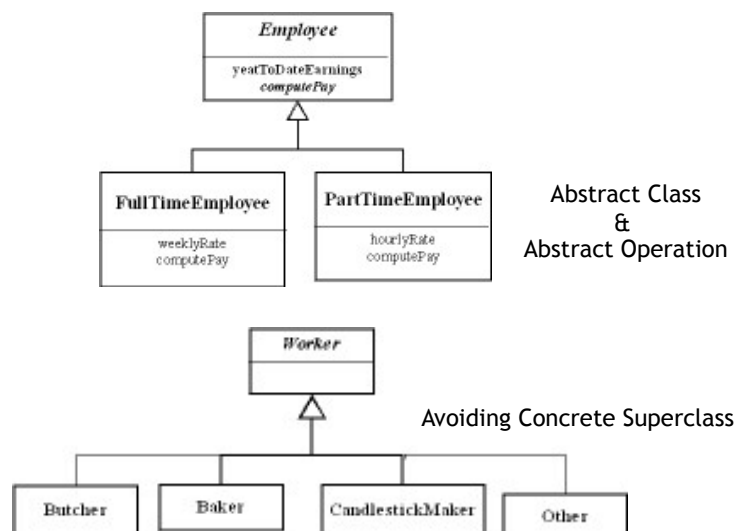- single value [1]
- optional single value [0..1]
- many [*]

Person
Name: string [1] Address: string [1..*] phoneNumber: string [*] birthDate: date [1]

Abstract Class

An abstract class is a class that has no direct instances but whose descendant classes have direct instances.

Note: the listing of workers is incomplete

A concrete class is a class that is instantiable; it can have direct instances.

Abstract Class

Abstract Class
&
Abstract Operation

Avoiding Concrete Superclass
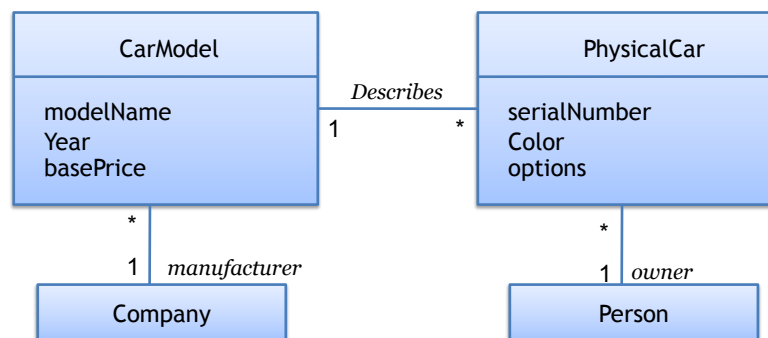
Metadata

Metadata is data that describes other data

- A Class Definition is Metadata.
- Models are inherently metadata, describe the things being modeled
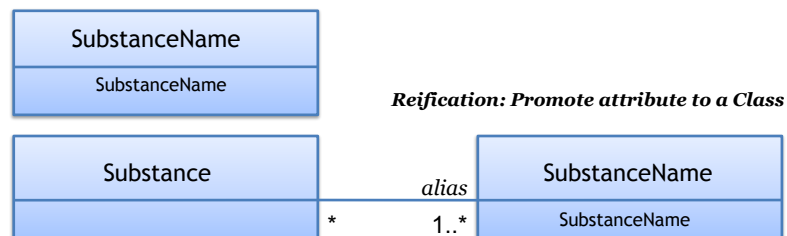- Programming Language implementations also use metadata.

CarModel	
modelName Year basePrice	

Describes 1   *

PhysicalCar	
serialNumber Color options	

*

1   *manufacturer*

Company

*

1   *owner*

Person

Reification

Reification is the promotion of something that is not an object into an object.

- It lets you to shift the level of abstraction

- For example, Consider a Database Manager. A developer could write code for each application to read and write files.

Writing for all other applications????
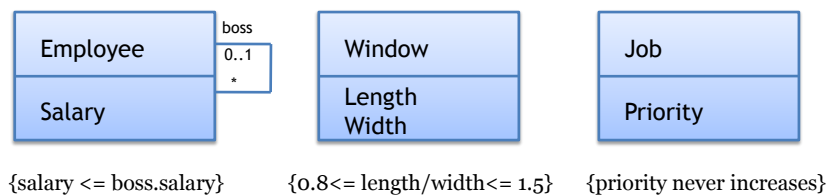*Reify: Data services and use DATA Manager.*

SubstanceName
SubstanceName

Reification: Promote attribute to a Class

Substance

alias   *   1..*

SubstanceName
SubstanceName

Constraints

A *Constraint* is a Boolean condition involving model elements, such as objects, classes, attributes, links, associations, and generalization sets.

- It restricts the values that elements can assume.
- Natural Language or a formal language **(OCL; Object Constraint Language)**
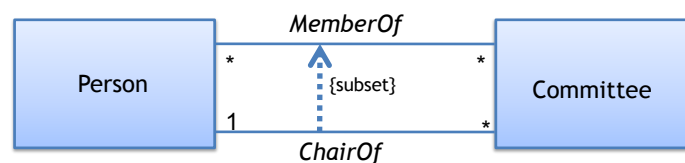
Constraints on Objects

Employee	boss
Salary	0..1 *

Window
Length   Width

Job
Priority

{salary <= boss.salary}   {0.8<= length/width<= 1.5}   {priority never increases}

Constraints

Constraints on Links

Multiplicities, association names, association end names, qualification, {ordered}, {sequence}and many more….

Explicit constraint:
The chair of a committee is a member of the committee; the ChairOf association is a subset of the MemberOf association.

Person	*MemberOf*	Committee
*	{subset}	*
1	*ChairOf*	*

Subset constraints between associations

Exercises
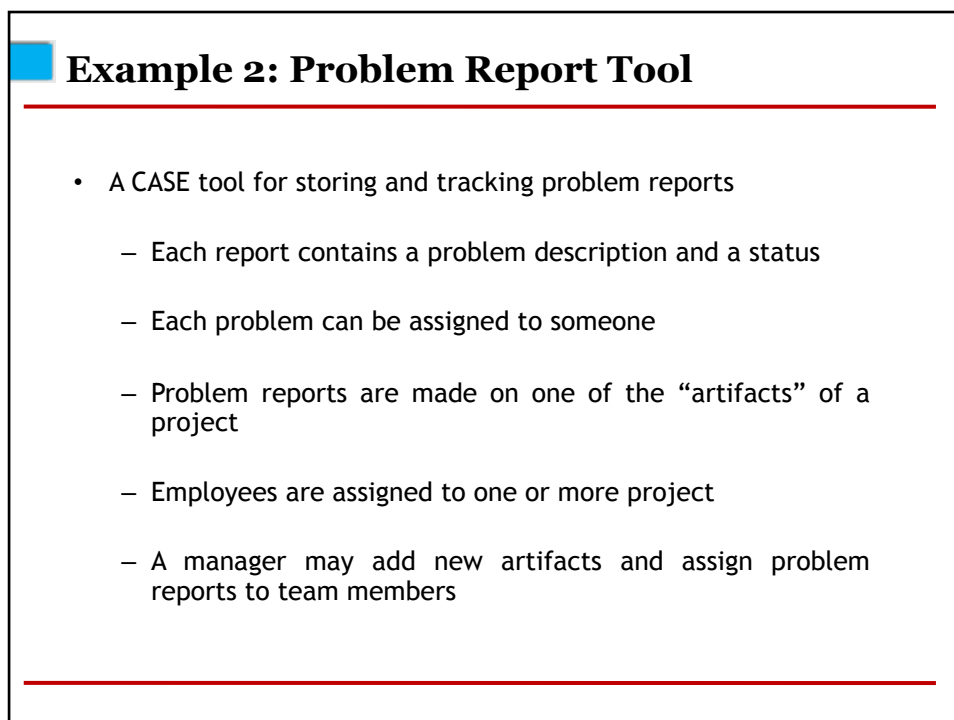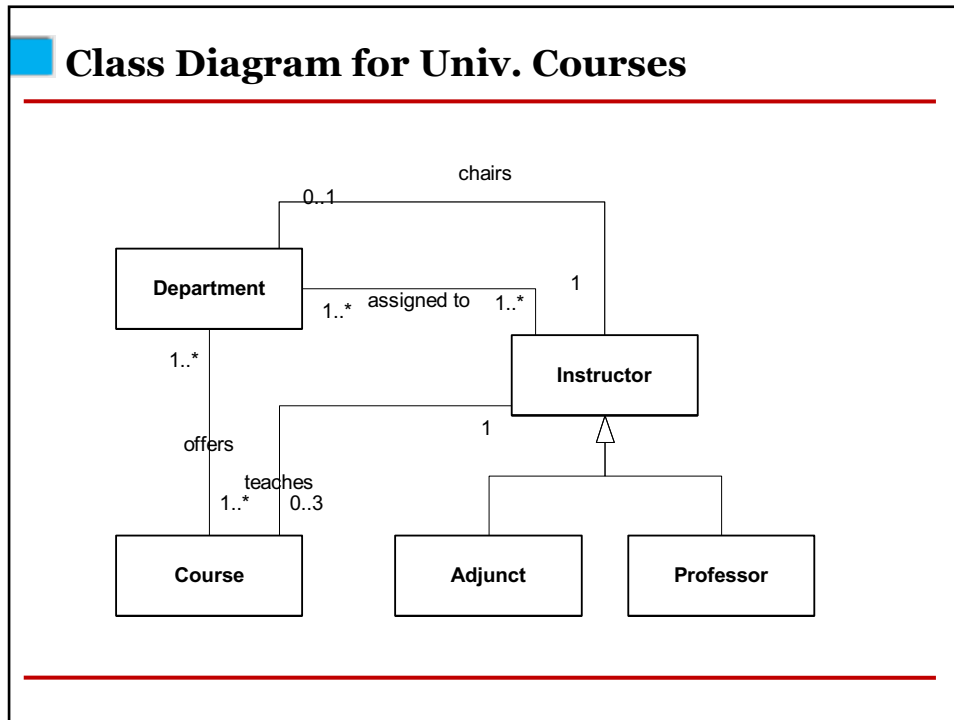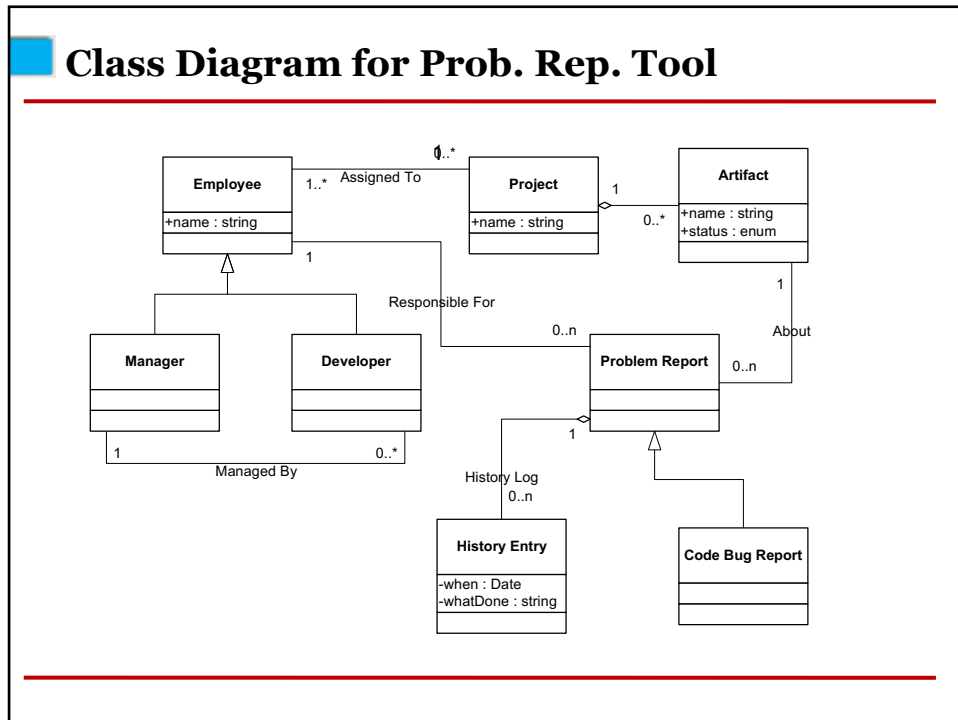
Example 1: University Courses

- Some instructors are professors, while others have job title adjunct

- Departments offer many courses, but a course may be offered by >1 department

- Courses are taught by instructors, who may teach up to three courses

- Instructors are assigned to one (or more) departments

- One instructor also serves a department chair

Class Diagram for Univ. Courses

Example 2: Problem Report Tool

- A CASE tool for storing and tracking problem reports

  – Each report contains a problem description and a status

  – Each problem can be assigned to someone

  – Problem reports are made on one of the "artifacts" of a project

  – Employees are assigned to one or more project

  – A manager may add new artifacts and assign problem reports to team members

Class Diagram for Prob. Rep. Tool

Questions??