Module 1: Introduction to OOAD
Author:: SARMAD BALOCH

(Object-Oriented Analysis and Design)

SARMAD BALOCH

Objectives

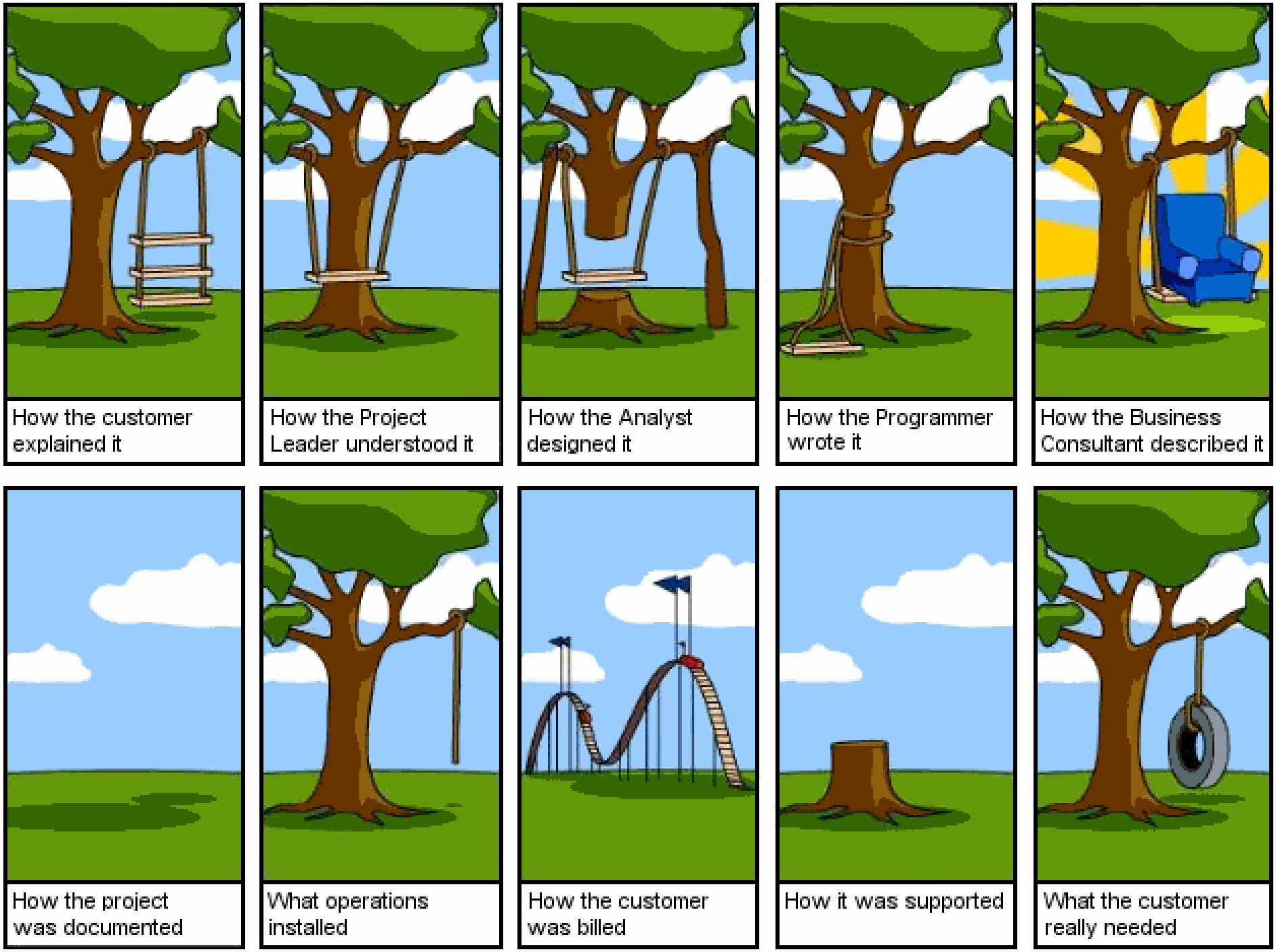
- Why OO?
- What is OOAD?
- How to do OOAD?

Why Object-Oriented?

Genesis 11:1-9 Acts 2:1-4
The Tower Of Babel

Let's become famous by building a city with a tower that reaches up to heaven (verse four).

Let's go down and confuse their speech right away, and make it so that they will not understand each other's speech. (verses five through seven).

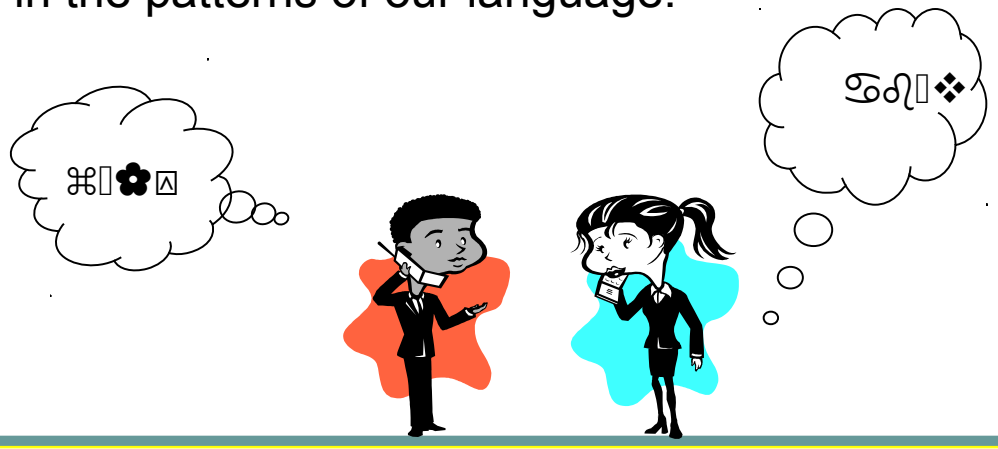

How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

Why Object-Oriented?

*"The "software crises" came about when people realized the major problems in software development were … caused by **communication** difficulties and the management of **complexity**" [Budd]*

The Whorfian Hypothesis:

Human beings … are very much at the mercy of the particular language which has become the medium of expression for their society … the 'real world' is … built upon the language habits … We cut nature up, organize it into concepts, and ascribe significances as we do, largely because we are parties to an agreement to organize it in this way … and is codified in the patterns of our language.



What kind of language can alleviate difficulties with communication & complexity hopefully well?

Why Object-Oriented?
– Consider Human Growth & Concept Formation

- Communication & complexity about the problem and the solution, all expressed in terms of ***concepts*** in a language!
- But then, What is CONCEPT? [Martin & Odell]
- *Consider Human Growth & Concept Formation*

stage	concepts
infant	*the world is a buzzing confusion*
very young age	*"blue" "sky" (individual concepts)* *"blue sky" (more complex concept)* *hypothesis: humans possess an innate capacity for perception*
getting older	*-> increased meaning, precision, subtlety, ...* *the sky is blue only on cloudless days* *the sky is not really blue* *it only looks blue from our planet Earth* *because of atmospheric effects* *elaborate conceptual constructs*

Concept formation: from chaos to order!

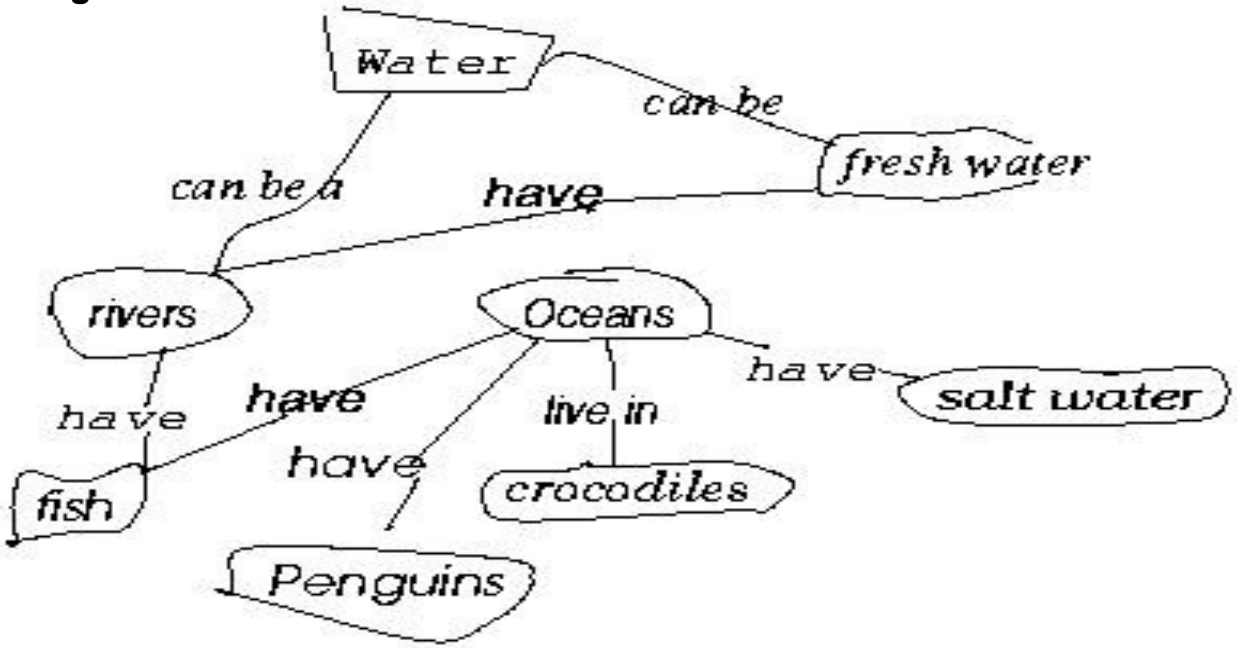
Why Object-Oriented?
- concepts and objects

So, concepts are needed to bring order … into

But, What is CONCEPT? [Martin & Odell] [Novak, 1984, Cambridge University Press]

Study of a first grade class

When given a list of concepts (water, salt water, Oceans, Penguins,...),
Harry constructed a concept diagram through which he understands his world and communicates meaning

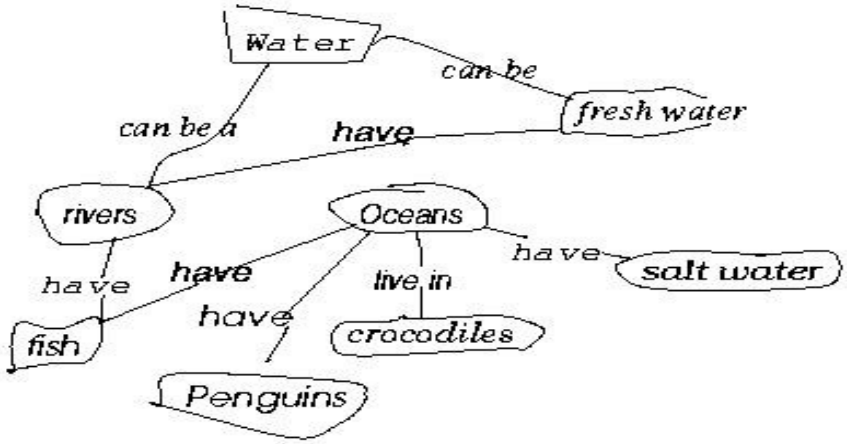


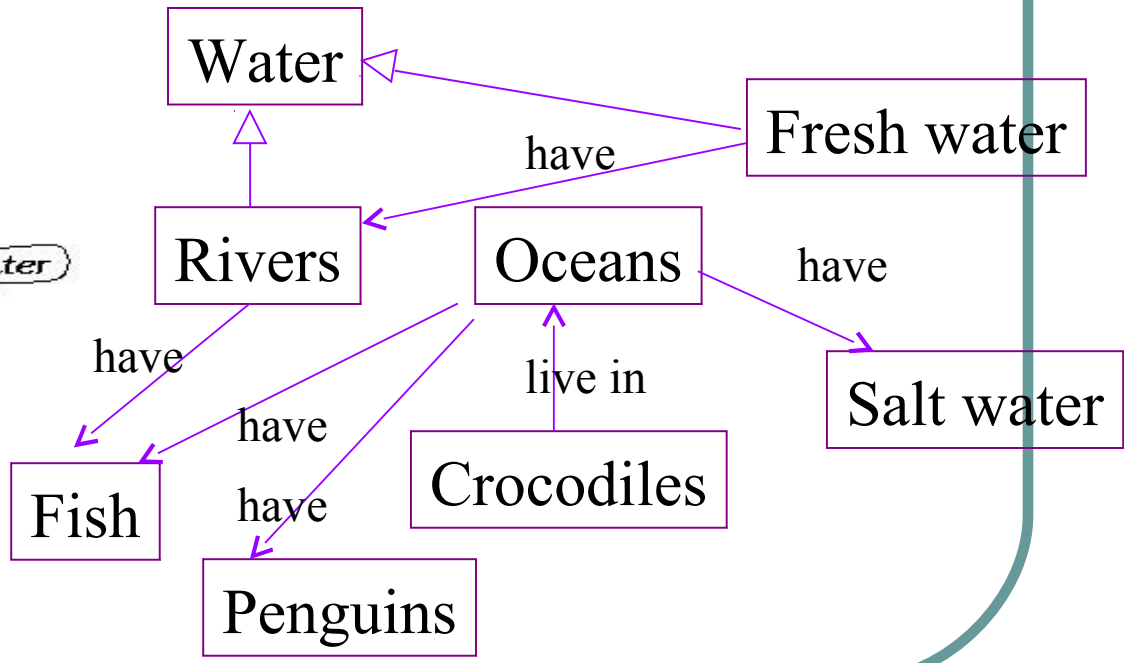
Does Harry understand the concepts? Do YOU understand what Harry understands? Agree or Diagree?

Why Object-Oriented?

… for Conceptual … Modeling Reasons

What kind of language can be used to create this concept diagram, or Harry's mental image?





Why Object-Oriented ->

What is a ***model*** **and why**?

- A model is a simplification of reality.

  E.g., a miniature bridge for a real bridge to be built

  - *Well...sort of….but not quite*
  - A model is *our* simplification of *our perception* of reality

    (that is, if it exists, otherwise it could be a mere illusion).

    communication is not about reality but about your/my/his/her perception of reality => validation and verification hard but needed

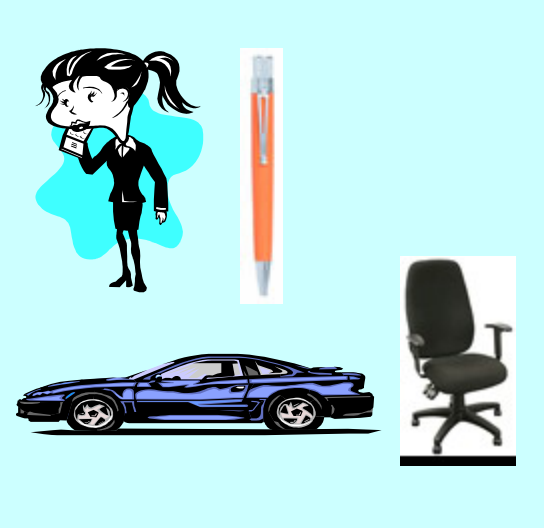
- A model is an *abstraction* of something for the purpose of *understanding*, be it the problem or a solution.

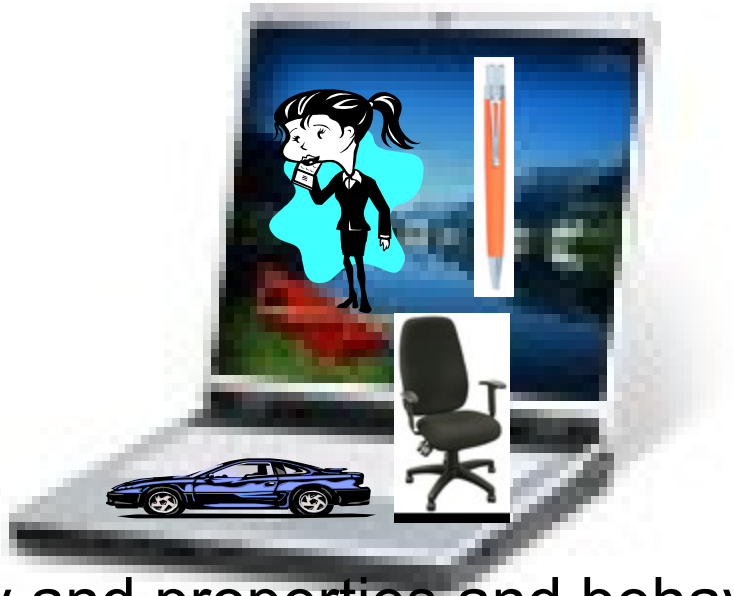
- To understand *why* a software system is needed, *what* it should do, and *how* it should do it.
- To communicate our understanding of why, what and how.
- To detect commonalities and differences in your perception, my perception, his perception and her perception of reality.
- To detect misunderstandings and miscommunications.

What is Object-Orientation?

- What is Object?

- An "object" is anything to which a concept applies, ***in our awareness***
- Things drawn from the problem domain or solution space.
  - E.g., a living person in the problem domain, a software component in the solution space.





- A structure that has identity and properties and behavior
- It is an instance of a collective concept, i.e., a class.

What is Object-Orientation
- Abstraction and Encapsulation

Abstraction

Focus on the essential

Omits tremendous amount of details

…Focus on what an object "is and does"

Encapsulation

a.k.a. information hiding

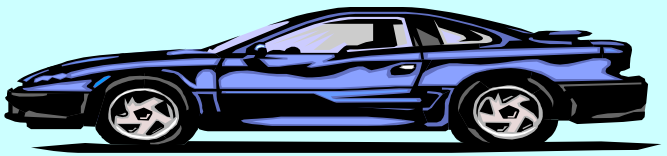
Objects encapsulate:

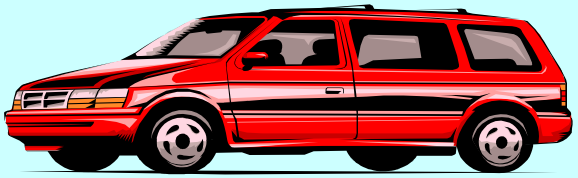
property

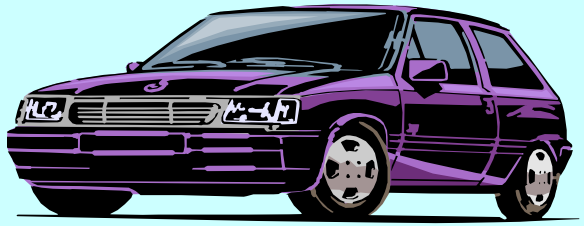
behavior as a collection of methods invoked by messages

…state as a collection of instance variables

What is Object-Orientation
- Another Example of Abstraction and Encapsulation







<<instanceOf>>

<<instanceOf>>

<<instanceOf>>

Class Car

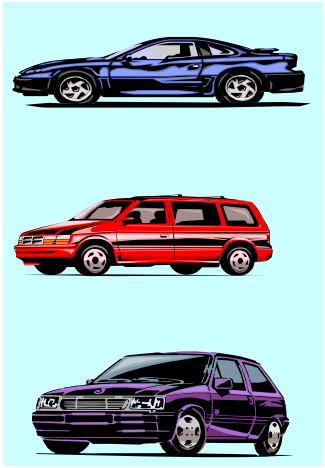
Attributes
- Model
- Location
- #Wheels = 4

Operations
- Start
- Accelerate

What is generalization?
What is over-generalization??
Forall x [Car(x) -> …]

What is Object-Orientation?

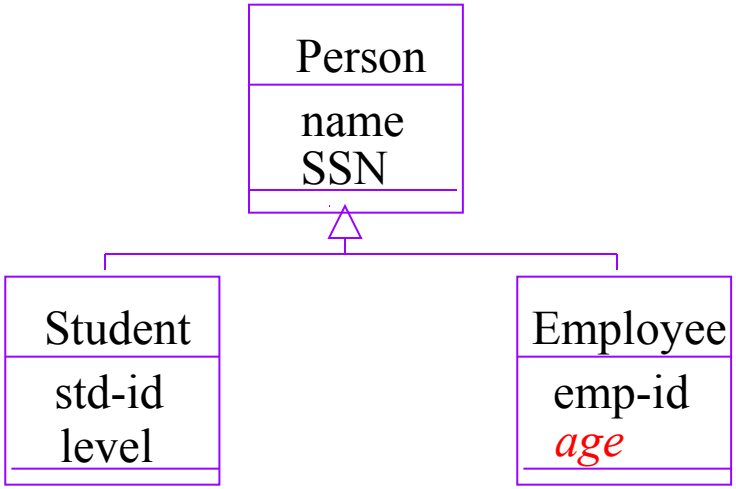
- Class



- **What is *CLASS*?**
  - a collection of objects that share common properties, attributes, behavior and semantics, in general. ***What are all these???***
  - A collection of objects with the same data structure (attributes, state variables) and behavior (function/code/operations) in the solution space.
- **Classification**
  - Grouping of common objects into a class
- Instantiation.
  - The act of creating an instance.

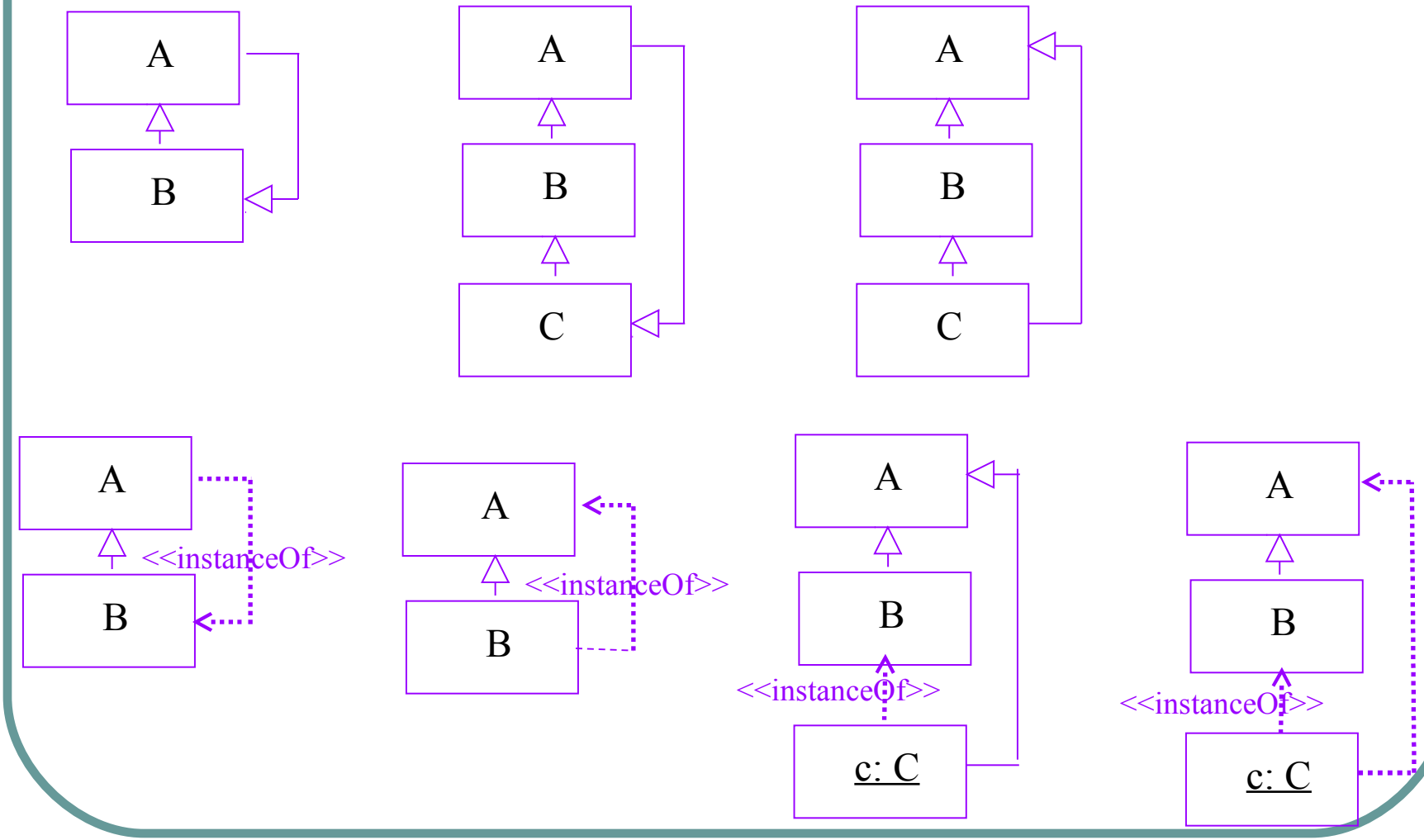
What is Object-Orientation
- Subclass vs. Superclass

- Specialization: The act of defining one class as a refinement of another.
- Subclass: A class defined in terms of a specialization of a superclass using inheritance.
- Superclass: A class serving as a base for inheritance in a class hierarchy
- Inheritance: Automatic duplication of superclass attribute and behavior definitions in subclass.



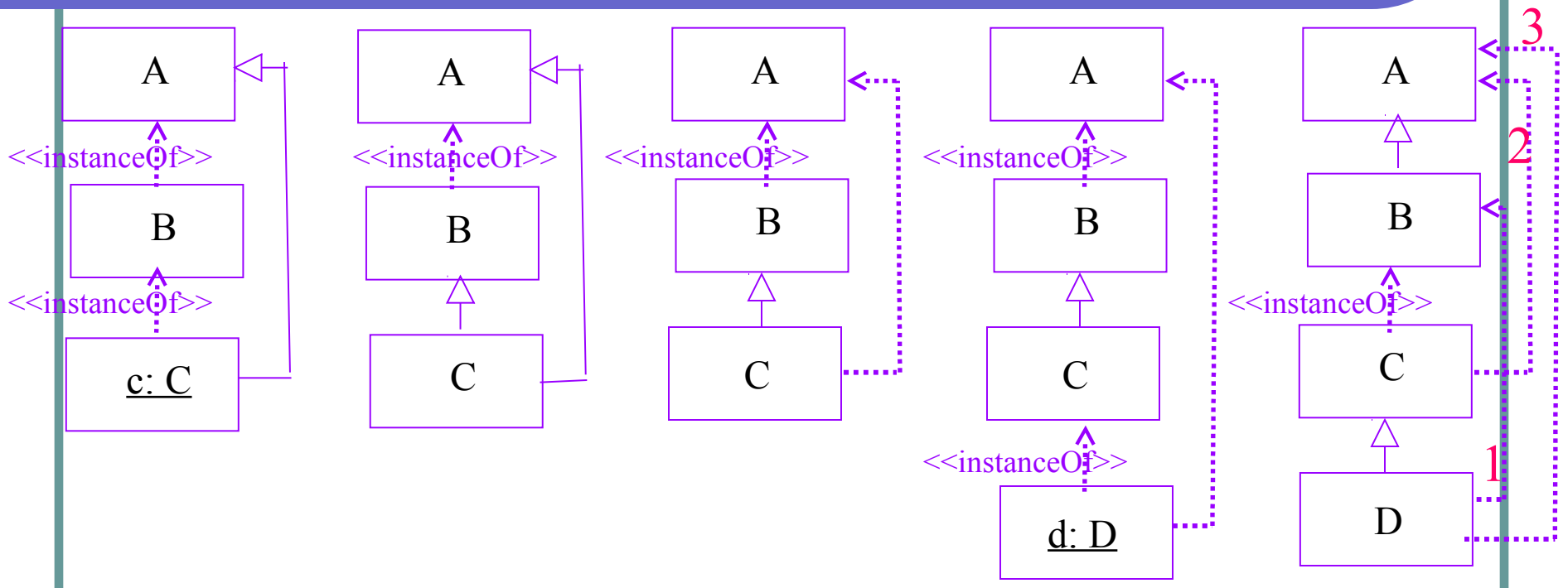
multiple inheritance?

What is Object-Orientation
- Subclass vs. Superclass



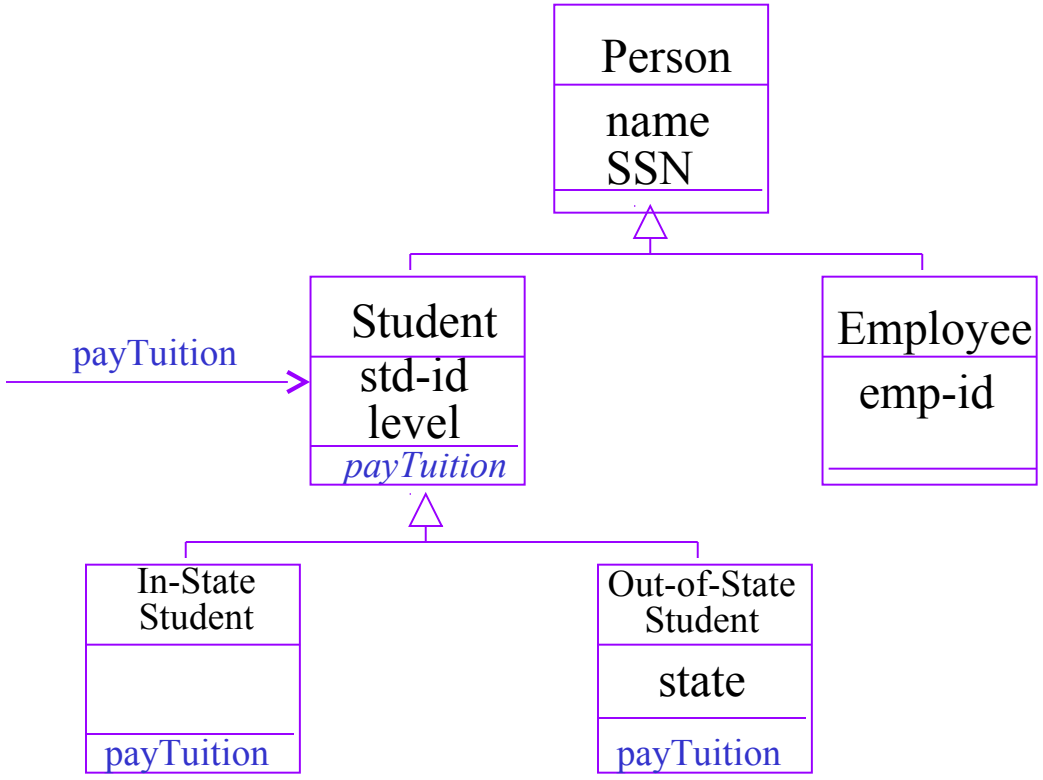
What is Object-Orientation

- Subclass vs. Superclass and …



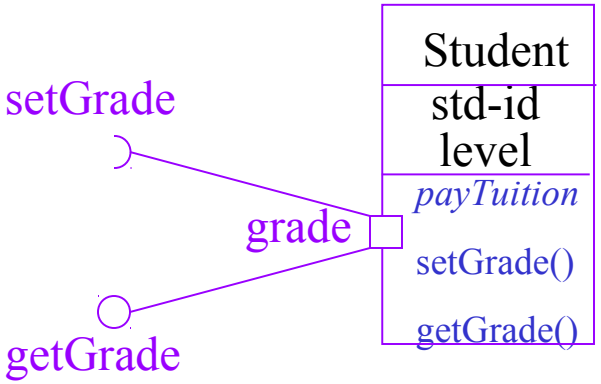
What is Object-Orientation
- Polymorphism

Objects of different classes respond to the same message differently.



What is Object-Orientation
- Interfaces

- Information hiding - all data should be hidden within a class, at least in principle.
- make all data attributes private
- provide public methods to get and set the data values (cf. Java design patterns)
- e.g. Grade information is usually confidential, hence it should be kept private to the student. Access to the grade information should be done through *interfaces*, such as setGrade and getGrade



What is Object-Orientation
- Abstract Class vs. Concrete Class

- Abstract Class.
  - An *incomplete* superclass that defines common parts.
  - Not instantiated.
- Concrete class.
  - Is a *complete* class.
  - Describes a concept completely.
  - Is intended to be instantiated.

Work out an example!

What is Object-Orientation?
-State

- **What is *STATE*?**

  "State" is a collection of association an object has with other objects and object types.

- **What is *STATE CHANGE*?**

  A "state change" is the ***transition*** of an object from one state to another.

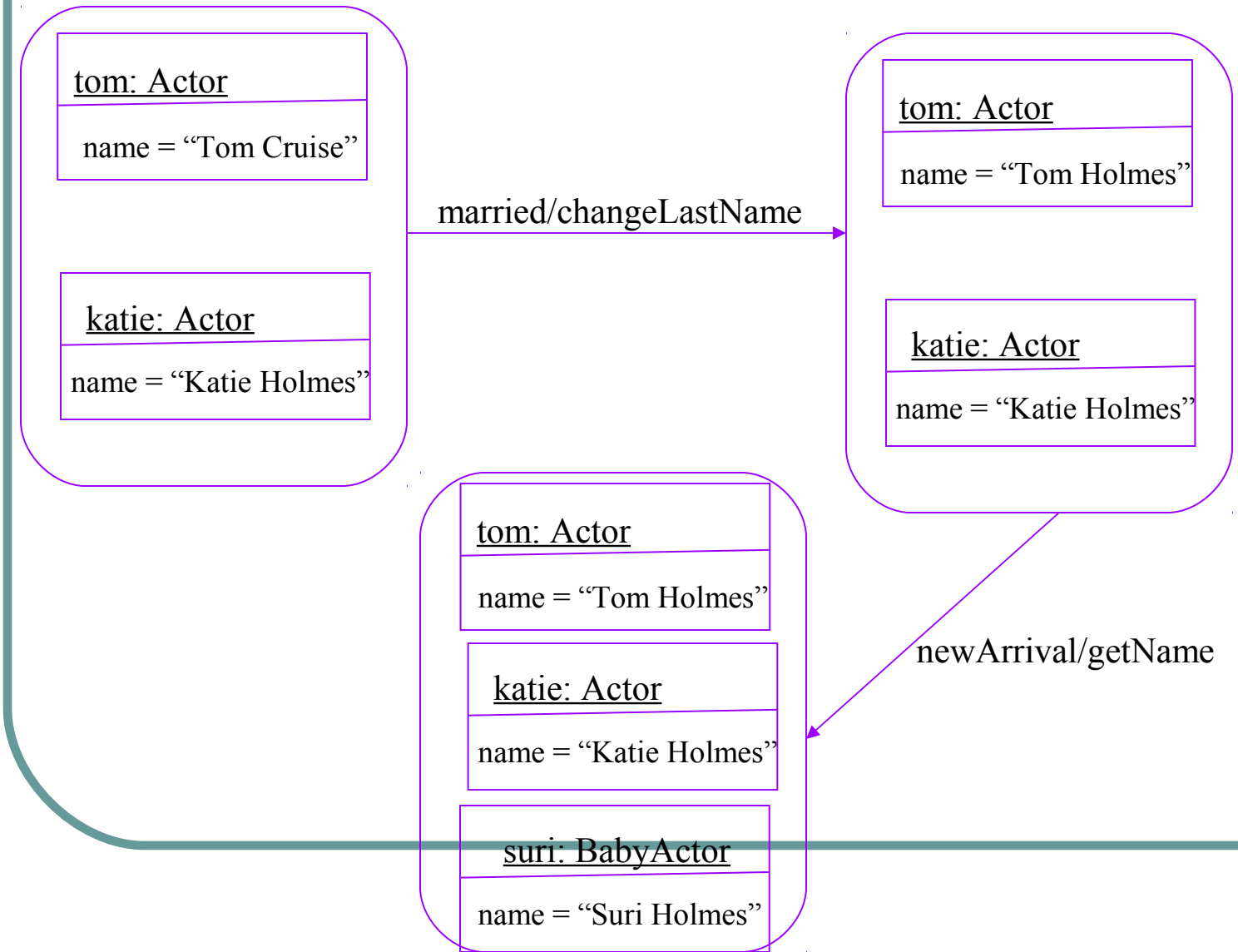
- **What is *EVENT*?**

  An "event" is a noteworthy change in state [Rumbaugh]

Work out an example!

What is Object-Orientation?

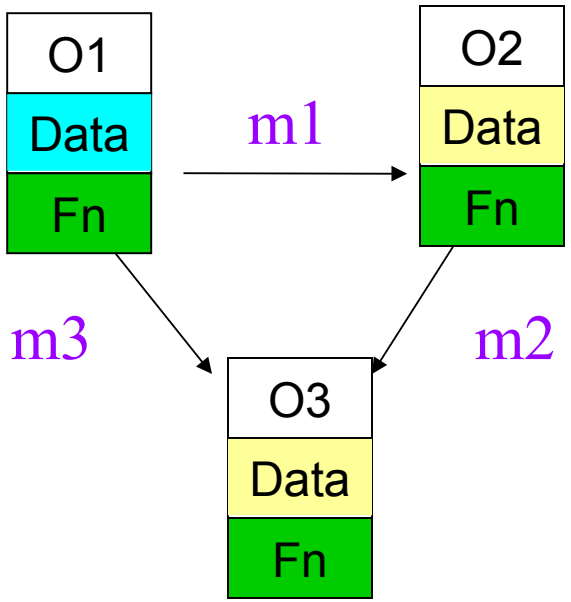
-State transition impossible?



What is Object-Oriented Application?

- Collection of discrete objects, interacting w. each other
- Objects have property and behavior (causing state transition)
- Interactions through message passing

  (A sender object sends a request (message) to another object (receiver) to invoke a method of the receiver object's)



{m in Fn}

What is OOAD?

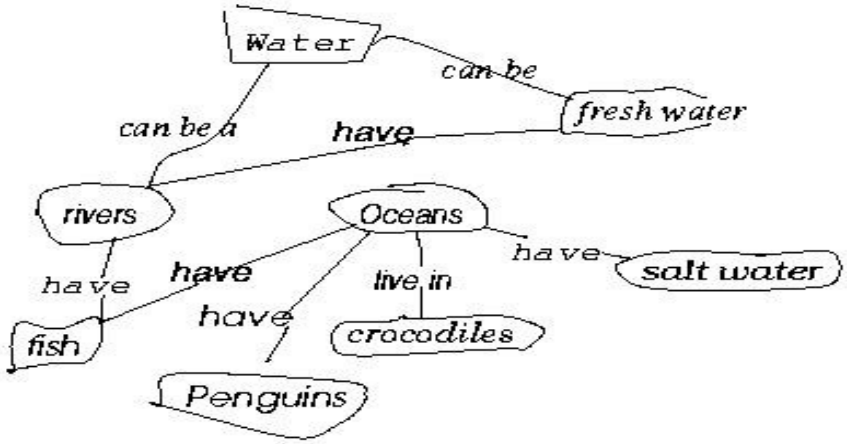
- Analysis — understanding, finding and describing concepts in the problem domain.
- Design — understanding and defining software solution/objects that *represent* the analysis concepts and will eventually be implemented in code.
- OOAD — Analysis is object-oriented and design is object-oriented. A software development approach that emphasizes a logical solution based on objects.

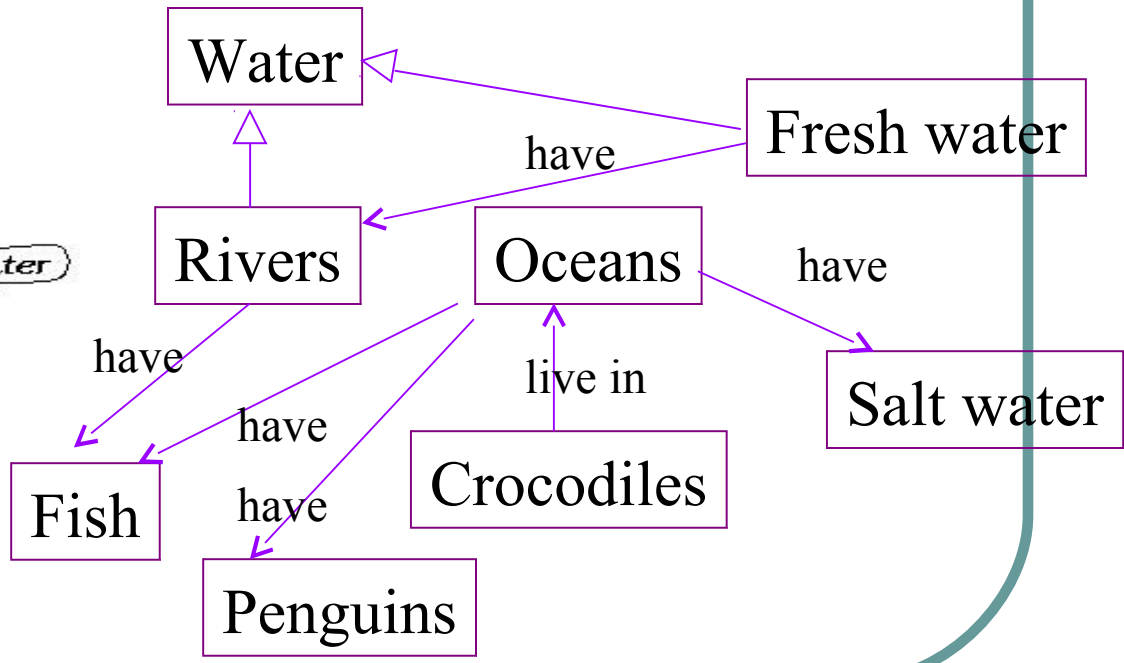
Traceability!

Involves both a notation and a process

Harry again …

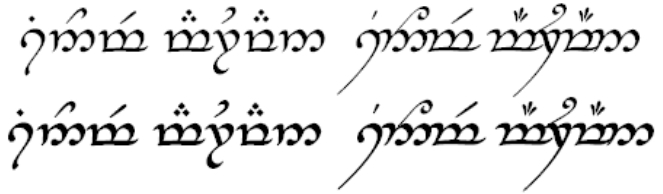
What do we see here?

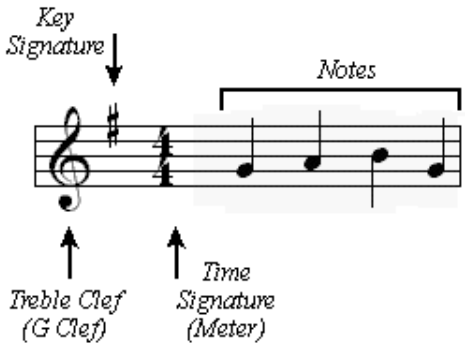




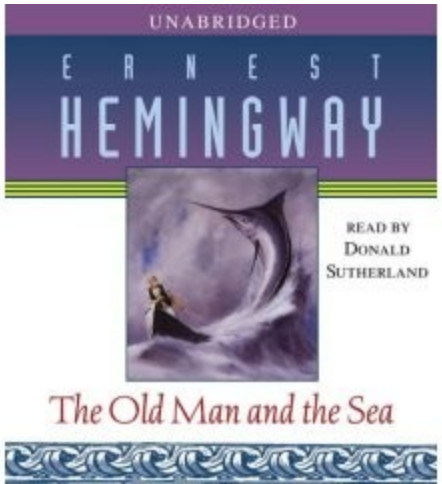
How to do OOAD - notation vs. process

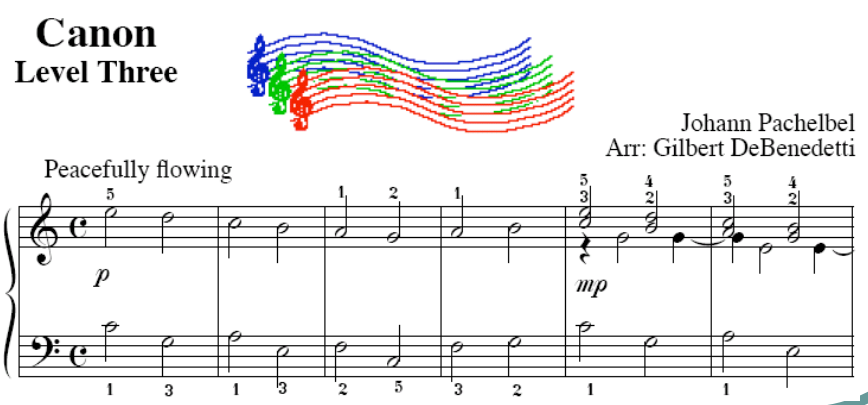
- UML is a notation.
- So are English, Elvish, Ku, …





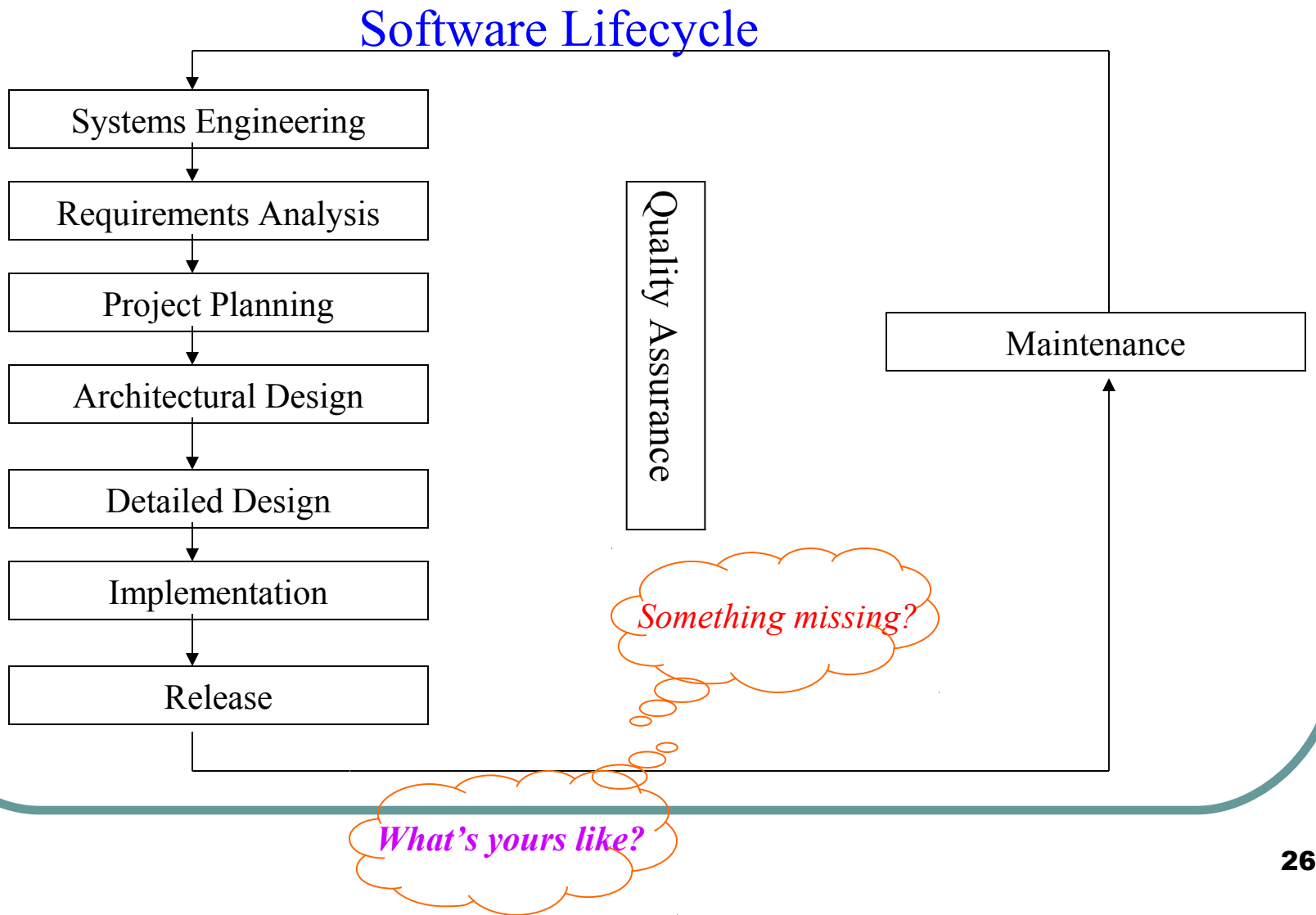
- But as yet I can't



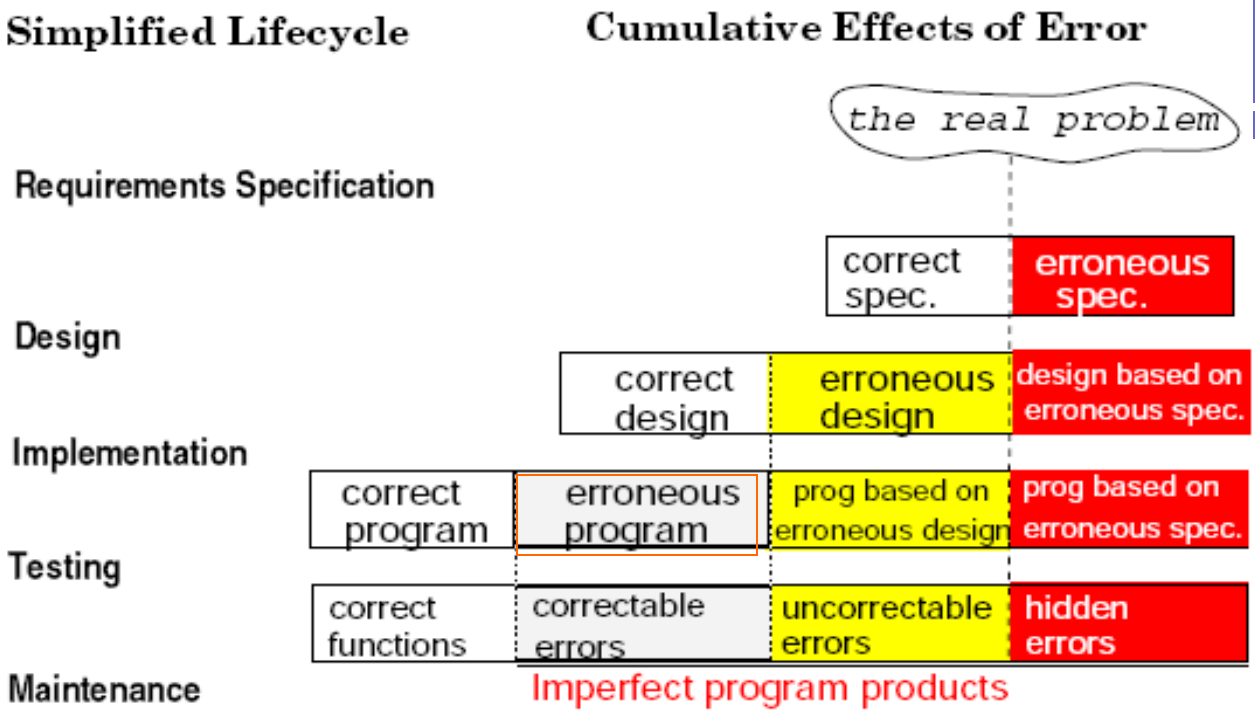


How to Do OOAD
– Where to Use OO?

Traceability!



Error Propagation in Lifecycle [Mizuno82]



Artificial problem

Accidental design

How big is the erroneous spec.?
How costly is it?

Traceability!

Specifications are important too

Why RE?

How big is the "erroneous specification"?

? **Bell Labs and IBM studies**

80% of all defects are inserted in the requirements phase.
Improving the requirements definition process reduces
the amount of testing and rework required.

And the above figures do not include the end user losses
who have to live with poor software on a daily basis[Testing Techniques Newslett

? **U.S. Air Force projects**

36% of all defects were due to faulty requirements translation.
Only 9% of these errors were resolved (in the requirements phase)[Sheldon92]

? **Voyager and Galileo spacecraft**

Of the 197 significant software faults found during integration & system testing
only 3 of those errors were programming errors;
the vast majority of the faults were requirements problems. [Lutz93]

? **Application Specific Integrated Circuits [ASICs)**

>1/2 are faulty on first fabrication. A majority of these faults are related to reqs. errors.

? **[UK Health and Safety] Executive**

Specification 44.1%	Operation and Maintenance 14.7%
Design and Implementation 14.7%	Changes after commissioning 20.6%
Installation and Commissioning 5.9%	

[Her Majesty's Stationary Office 1995 ISBN 0 7176 0847 6]

Lawrence Chung

Why RE?

How costly are requirements errors?

\# [Lindstrom93]

Get the requirements wrong, you'll destroy the proj

\# [Boehm87]

COST (correcting design/implementation errors)
= 100 X COST (correcting requirements errors)

\# [Humphrey, Managing the Software Process, Ch1, p11-12]

a useful rule of thumb: It takes about 1 to 4 working hours
find and fix a bug through inspections and about 15 to 20
working hours to find and fix a bug in function or system

\# [Curtis88]

Three most frequent problems plaguing large software systems:

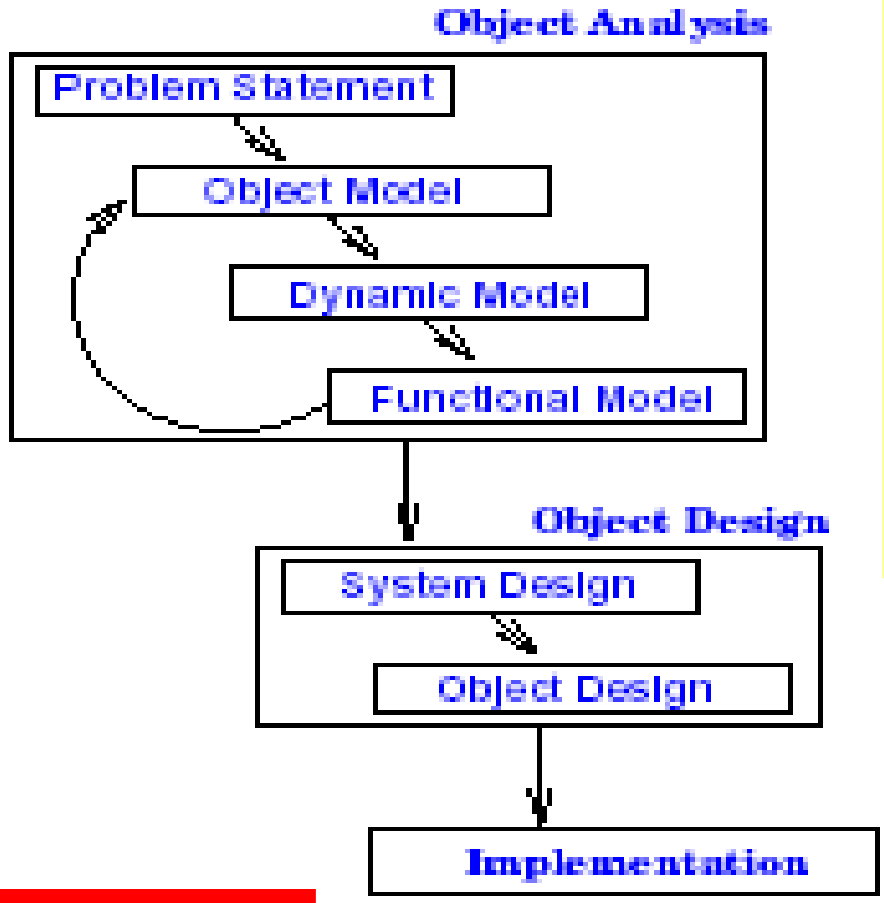
communication and coordination
thin spread of domain application knowledge
changing and conflicting requirements

Defining the problem is The Problem

Lawrence Chung

How to Do OOAD
– OMT as Object-Oriented Methodology

- ***OMT (Object Modeling Technique) by James Rumbaugh***



Object Model: describes the ***static*** structure of the objects in the system and their relationships -> Object Diagrams.

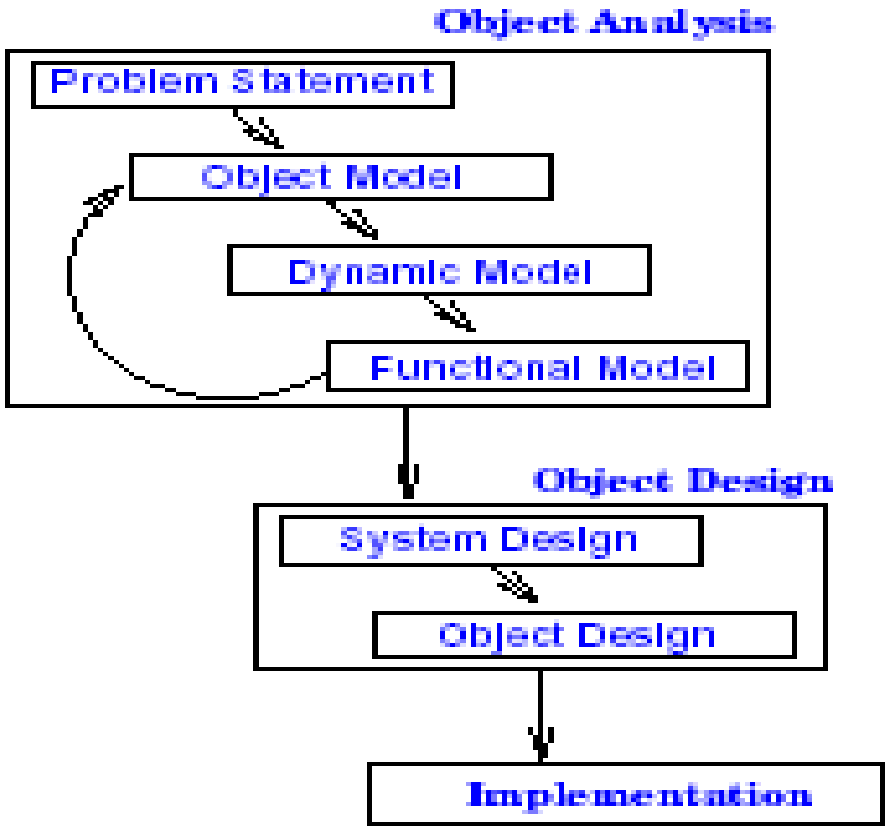
Dynamic Model: describes the ***interactions*** among objects in the system -> State Diagrams.

Functional Model: describes the data ***transformation*** of the system -> DataFlow Diagrams.

Traceability!

How to Do OOAD
– OMT as Object-Oriented Methodology

- ***OMT (Object Modeling Technique) by James Rumbaugh***



Analysis:

i) Model the ***real world*** showing its important properties;

ii) Concise model of what the ***system*** will do

System Design:

Organize into subsystems based on analysis structure and propose ***architecture***

Object Design: Based on analysis model but with implementation details; Focus on ***data structures and algorithms*** to implement each class; Computer and domain objects

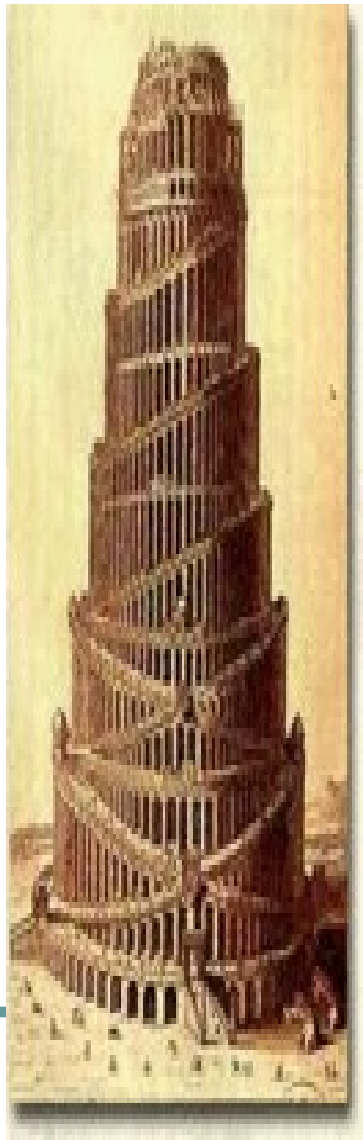
Implementation: Translate the object classes and relationships into a ***programming*** language

Traceability!

A Unified Language + A Good Process

+ *A Good Goal, perhaps*

UNIFIED MODELING LANGUAGE™



Introduction to OOAD - Summary

Why

- Once Software Crisis due to Communication and Complexity
- Languages, Concepts, Models
- OO for Conceptual Modeling

What

- Fundamental OO Concepts
- A little taste of UML

How

- OO development processes & (Design) Patterns

How to Do OOAD
- Historical Perspective

OO Technology	***Process Perspective***
OO Prog. Languages (Smalltalk, C++)	just program!
OO Design (Booch)	design then program
OO Analysis (Rumbaugh, Jacobson)	Analyze (use case) first, then design, then program

Where are we heading?

How to Do OOAD
- OO Development Processes

Some Popular OOAD Processes (for reference only)

- Fusion
  - Hewlett Packard
- Recommended Process and Models
  - ObjectSpace best practices
  - Larman's experiences
  - …
- The Rational Unified Process (RUP)
  - Rational; Booch, Jacobson, and Rumbaugh

How to Do OOAD
– One Good Way: Use (OO) Design Patterns

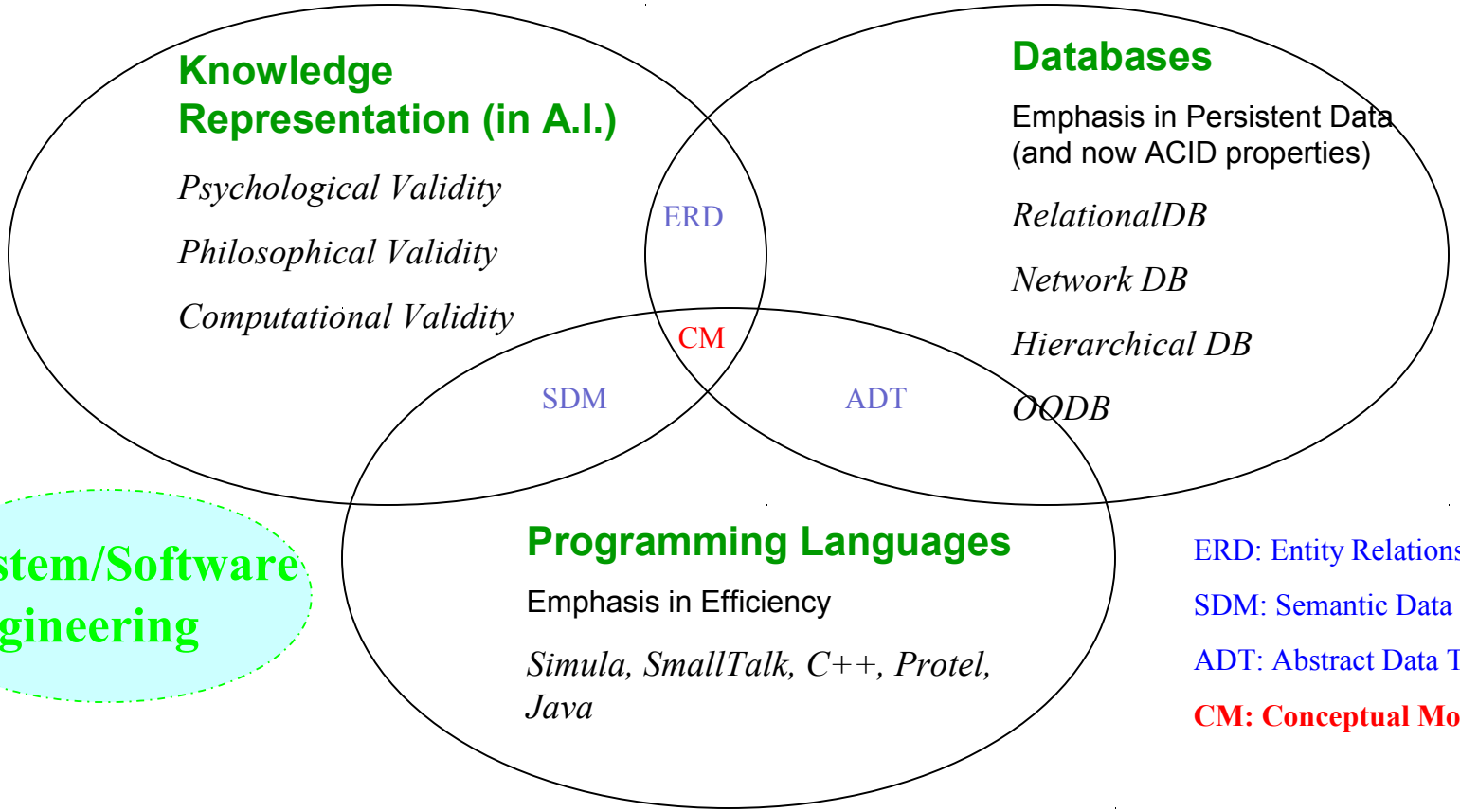
Reusable solutions to typical problems.

"Each design pattern systematically names, explains, and evaluates an important and recurring design in object-oriented systems." [Gamma]

- Name — identifies a particular pattern, creating a vocabulary.
- Problem — identifies context when pattern should be applied.
- Solution — an abstract description of a design problem along with a template object design that solves the problem.
- Consequences — results and trade-offs of applying the pattern.

Why Object-Oriented

- Who's Behind Object-Orientation *w. Diff. Concerns*



Where does ***Unified*** Modeling Language come into this?

Why Object-Oriented
– A New Paradigm with Evolving Object Orientation

- OOP: Object-Oriented Programming
  - Simula (1967), Smalltalk (70's), C++ (mid 80's), Eiffel, Ada95, Turing, …
- OOD: Object-Oriented Design
  - Taxis (1976), Adaplex, …, Grady Booch (1980)
- OOA: Object-Oriented Requirements
  - RML (1981), James Rumbaugh (late 80's)
- OO-Databases (OODBs): 1980-90's
- OLE/DCOM, VisualBasic, CORBA, Java: mid 90's
- .Net, C#, (eb/voice…/-)XML, J2EE: into 2000+
- UML: mid 90's and still evolving

Introduction to OOAD - Points to Ponder

1. How do you think your mental image is represented?
2. What kinds of languages are used for what purpose in our daily life?
3. What are the differences among a concept, a model and a language?
4. What are the differences between a language and a methodology?
5. Can we use C# for analysis?
6. If C++ is a language, does it model anything? If so, what?
7. What does a concept in C++ refer to (i.e., semantics)?
8. What does a concept in a (OO) design refer to?
9. What does a concept in an (OO requirements) analysis refer to?
10. Is the current OOAD for Functional Analysis and Design, or Non-Functional Analysis and Design?
11. What is the relationship between OO (Object-Orientation) and GO (Goal-Orientation), between OO and AO (Agent-Orientation), and between GO and AO?
12. Can you prove you and I communicate with each other perfectly?