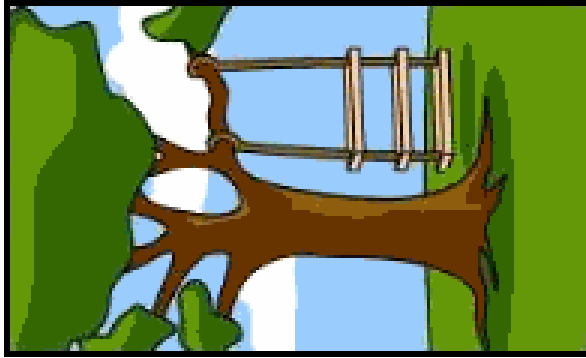


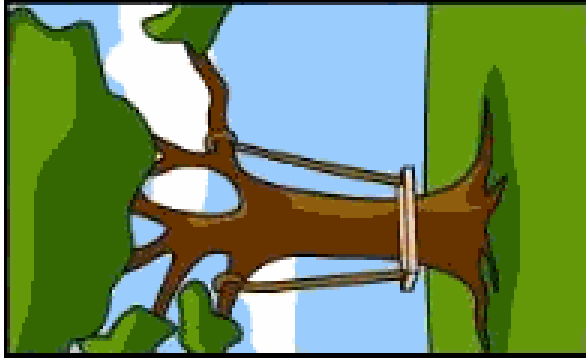
Module 1: Introduction to OOAD

(Object-Oriented Analysis and Design)

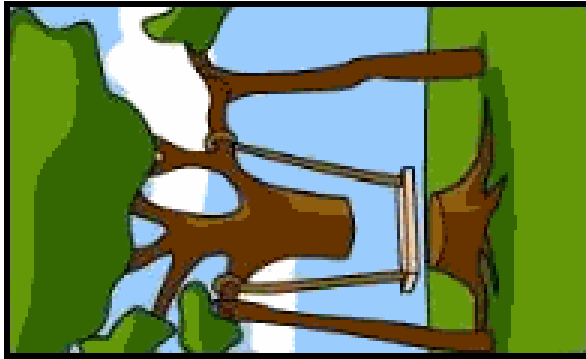
- ❑ Why OO?
- ❑ What is OOAD?
- ❑ How to do OOAD?



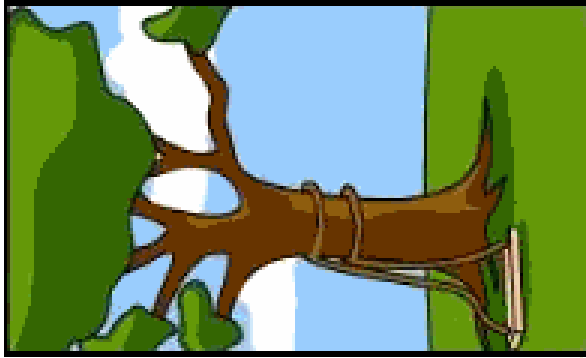
How the customer explained it



How the Project Leader understood it



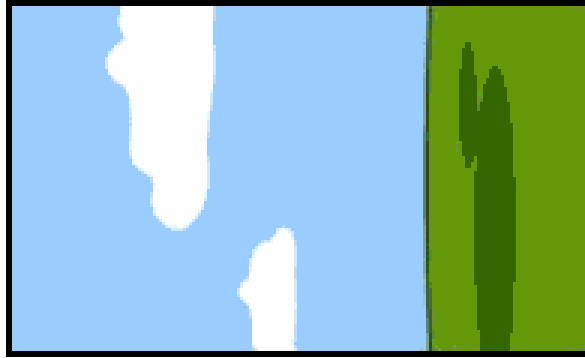
How the Analyst designed it



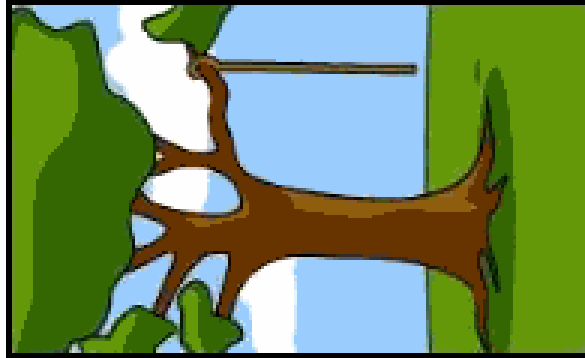
How the Programmer wrote it



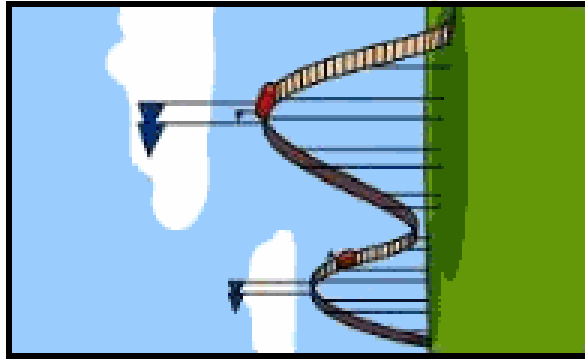
How the Business Consultant described it



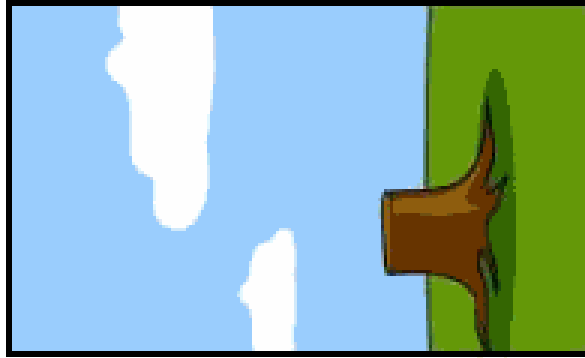
How the project was documented



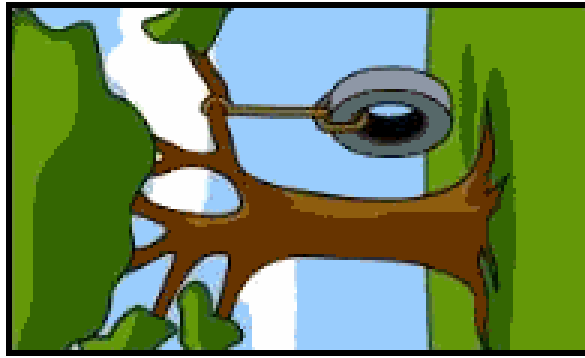
What operations installed



How the customer was billed



How it was supported



What the customer really needed

Why Object-Oriented?

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

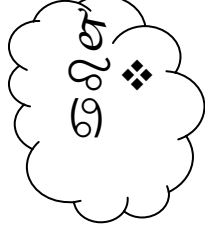
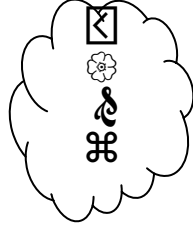
*Let's become famous
by building a city
with a tower that
reaches up to
heaven (verse
four).*

*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).*



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]*



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

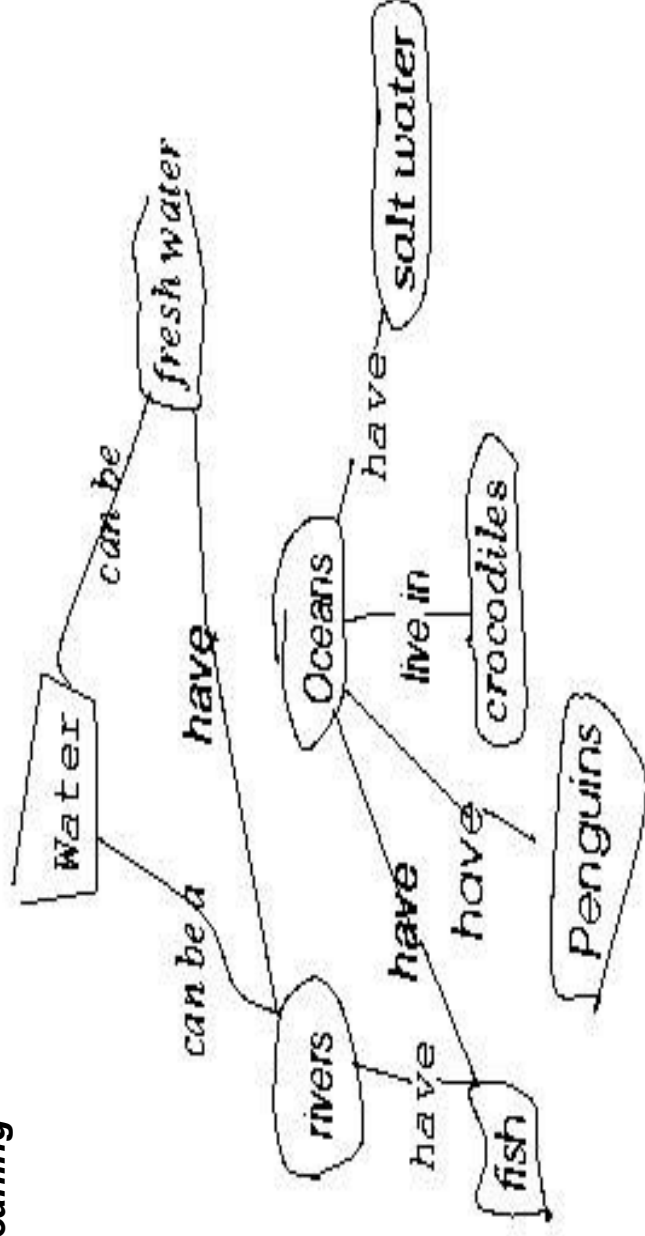
What does the “mythical man-month” mean?

Why Object-Oriented?

- concepts and objects

Study of a first grade class [Martin & Odell] [Novak, 1984, Cambridge University Press]

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**

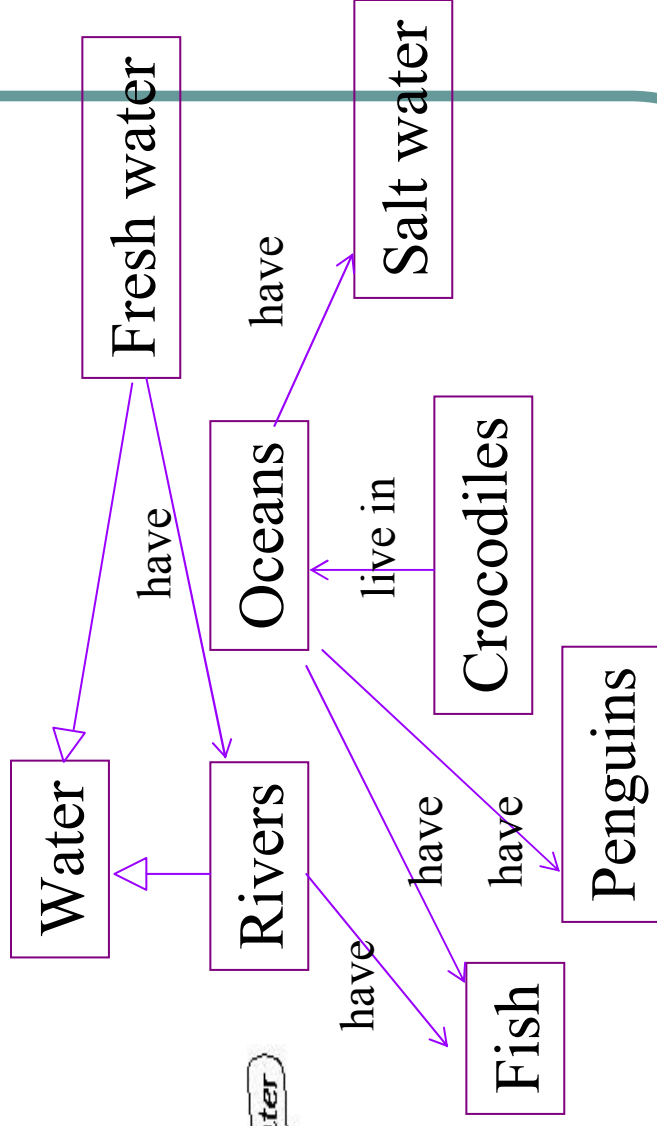
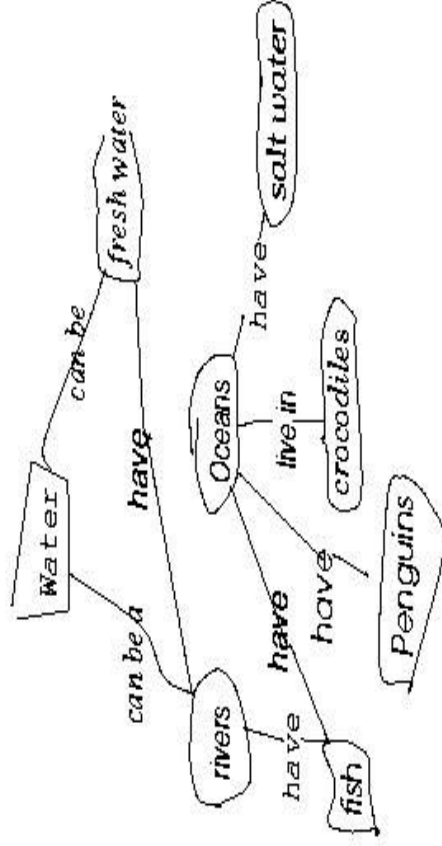


Do you understand what Harry understands?

Why Object-Oriented?

... for Conceptual ... Modeling Reasons

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



Things, Relationships, Diagram

Why Object-Oriented -> for Modeling

- 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

Abstraction

Focus on the essential

Omits tremendous amount of details

...Focus on what an object “is and does”

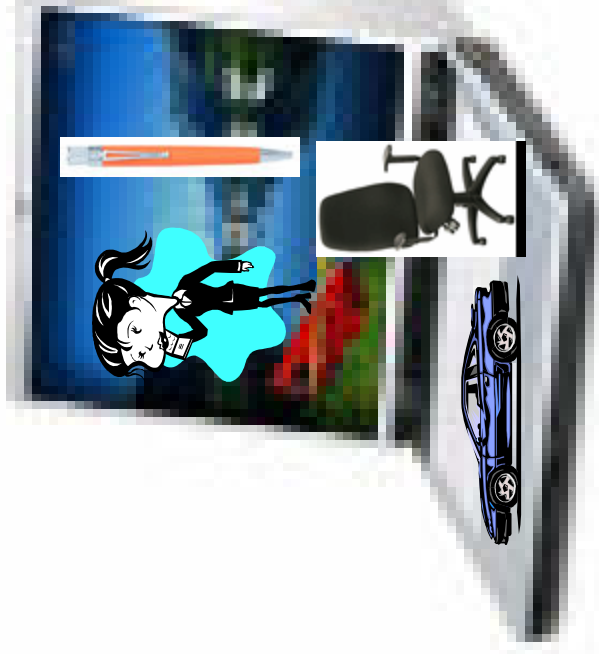


- To understand *why* a software system is needed, *what* it should do, and *how* it should do it.
- To communicate our understanding of *why*, *what* and *how*.
- To detect commonalities and differences in your perception, my perception, his perception and her perception of reality.
- To detect misunderstandings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

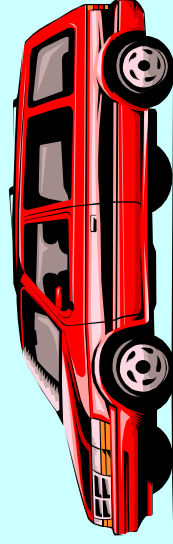


What is Object-Orientation

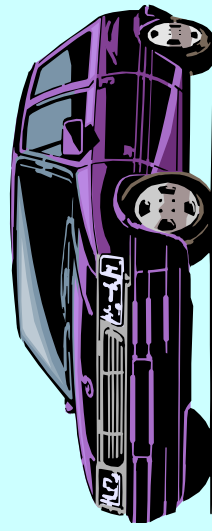
- Classification -> Classes -> Instantiation



<<instanceOf>>



<<instanceOf>>



<<instanceOf>>

Encapsulation

<u>Class</u> Car
<u>Attributes</u> <ul style="list-style-type: none"><input type="checkbox"/> Model<input type="checkbox"/> Location<input type="checkbox"/> #Wheels = 4
<u>Operations</u> <ul style="list-style-type: none"><input type="checkbox"/> Start<input type="checkbox"/> Accelerate

CLASS

- a collection of objects that share common properties, attributes, behavior and semantics, in general.
- A collection of objects with the same data structure (attributes, state variables) and behavior (function/code/operations) in the solution space.

Classification

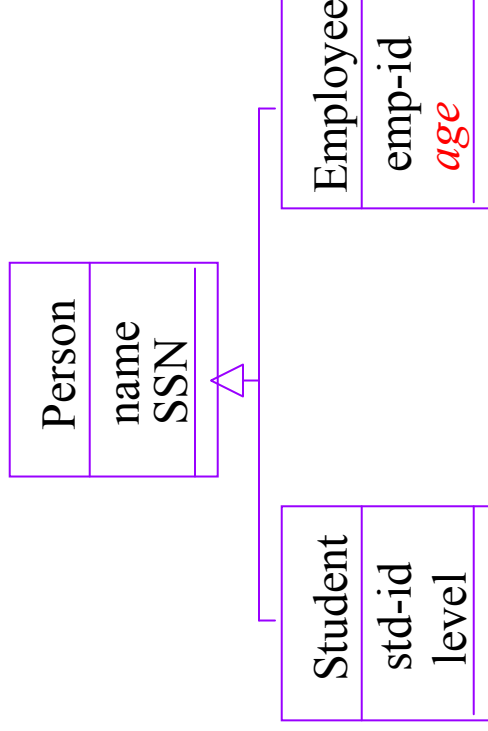
- Grouping of common objects into a class

Instantiation.

- The act of creating an instance.

What is Object-Orientation

- Subclass vs. Superclass

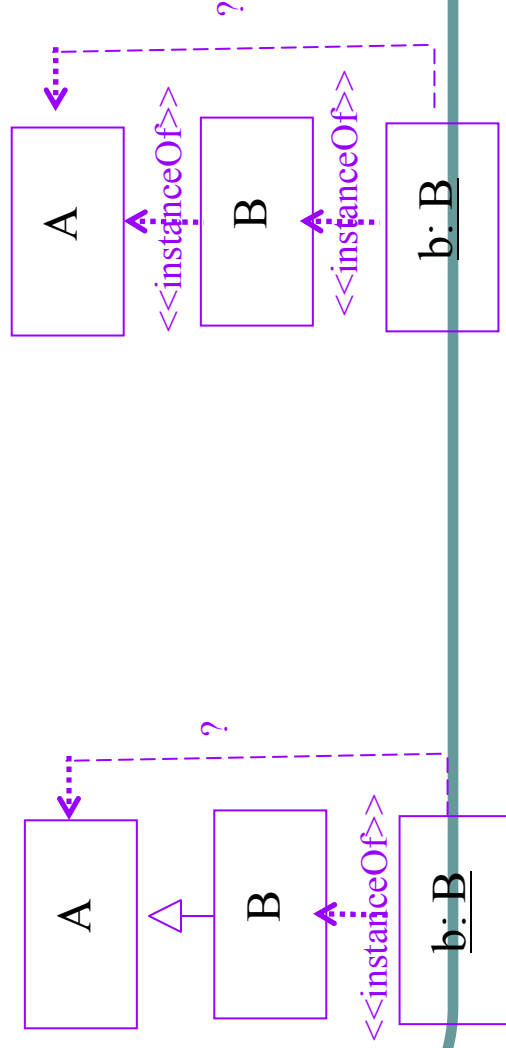
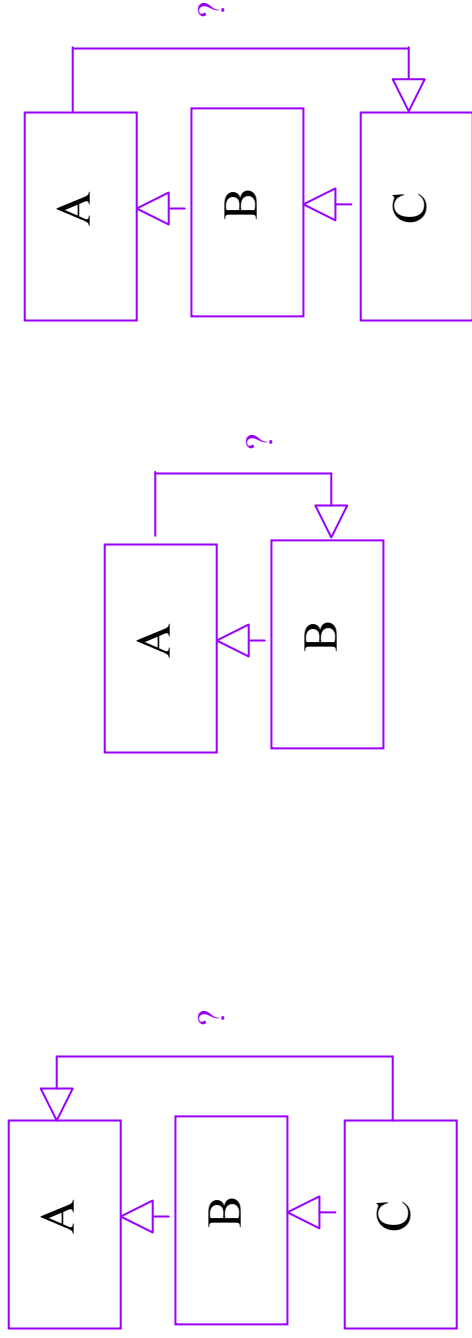


What is generalization?
What is over-generalization?
multiple inheritance?

- **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.

What is Object-Orientation

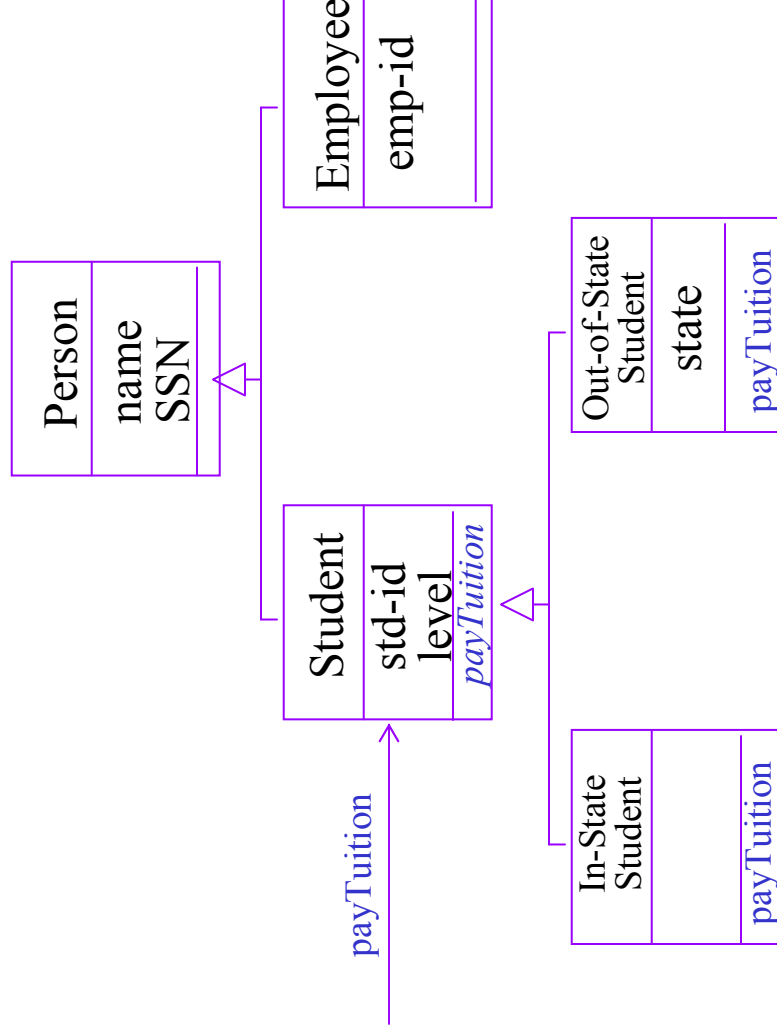
- Subclass vs. Superclass



What is Object-Orientation

- Polymorphism

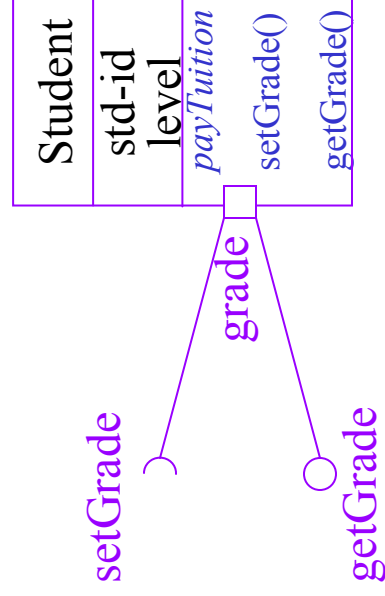
Objects of different classes respond to the same message differently.



What is Object-Orientation

- Interfaces

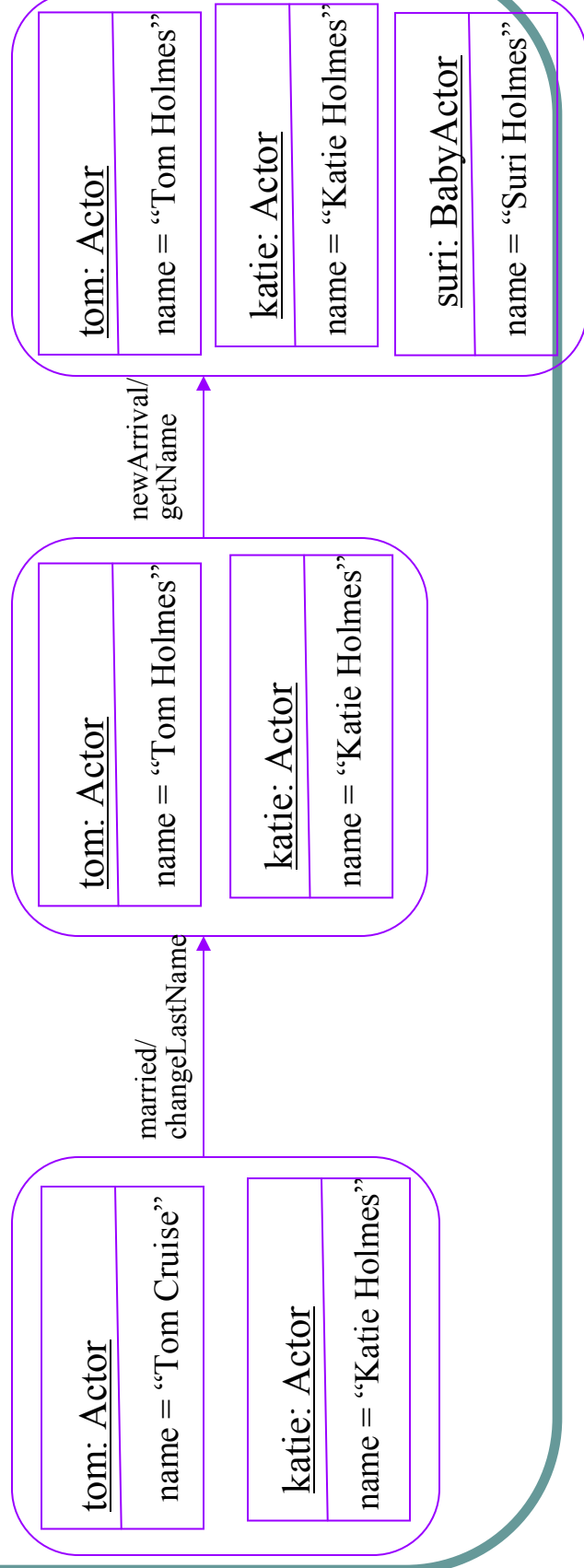
- Information hiding - all data should be hidden within a class, at least in principle.
- make all data attributes private
- provide public methods to get and set the data values (cf. Java design patterns)
- e.g. Grade information is usually confidential, hence it should be kept private to the student.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



What is Object-Orientation?

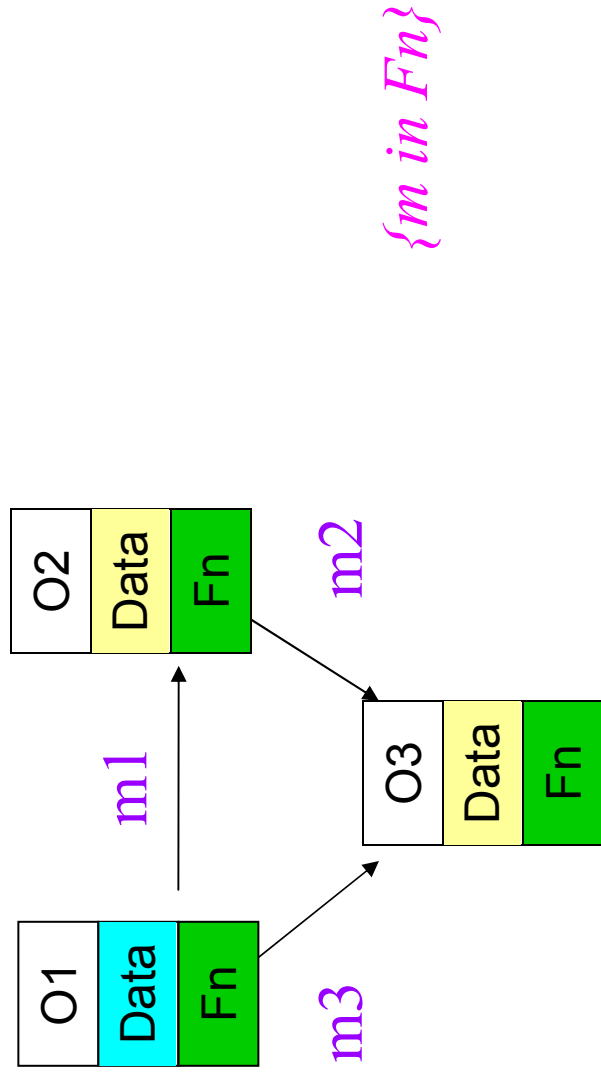
-States & State transitions

- **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]



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)



What is OOAD?

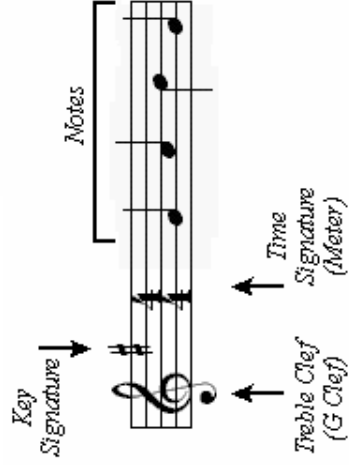
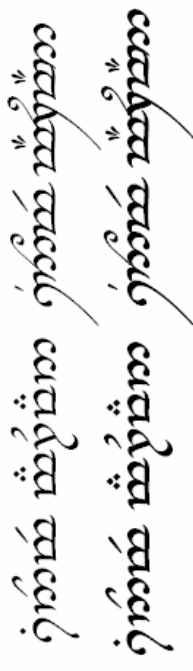
- **Analysis** — understanding, finding and describing concepts in the problem domain.
- **Design** — understanding and defining software solution/objects that *represent* the analysis concepts and will eventually be implemented in code.
- **OOAD** — Analysis is object-oriented and design is object-oriented.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

Traceability!

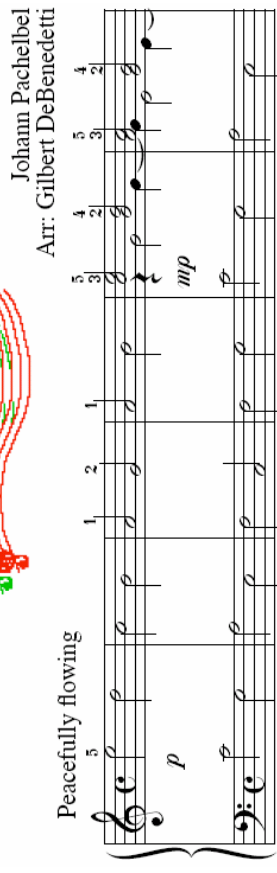
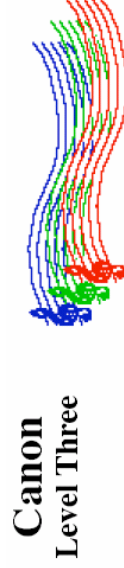
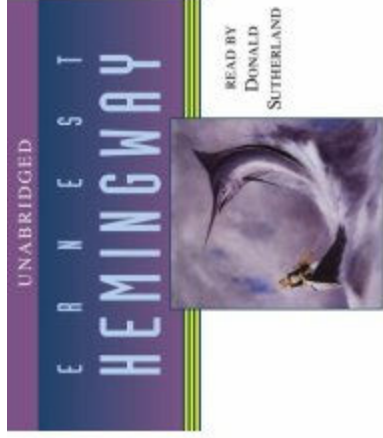
How to do OOAD

- notation vs. process

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



- But as yet I can't

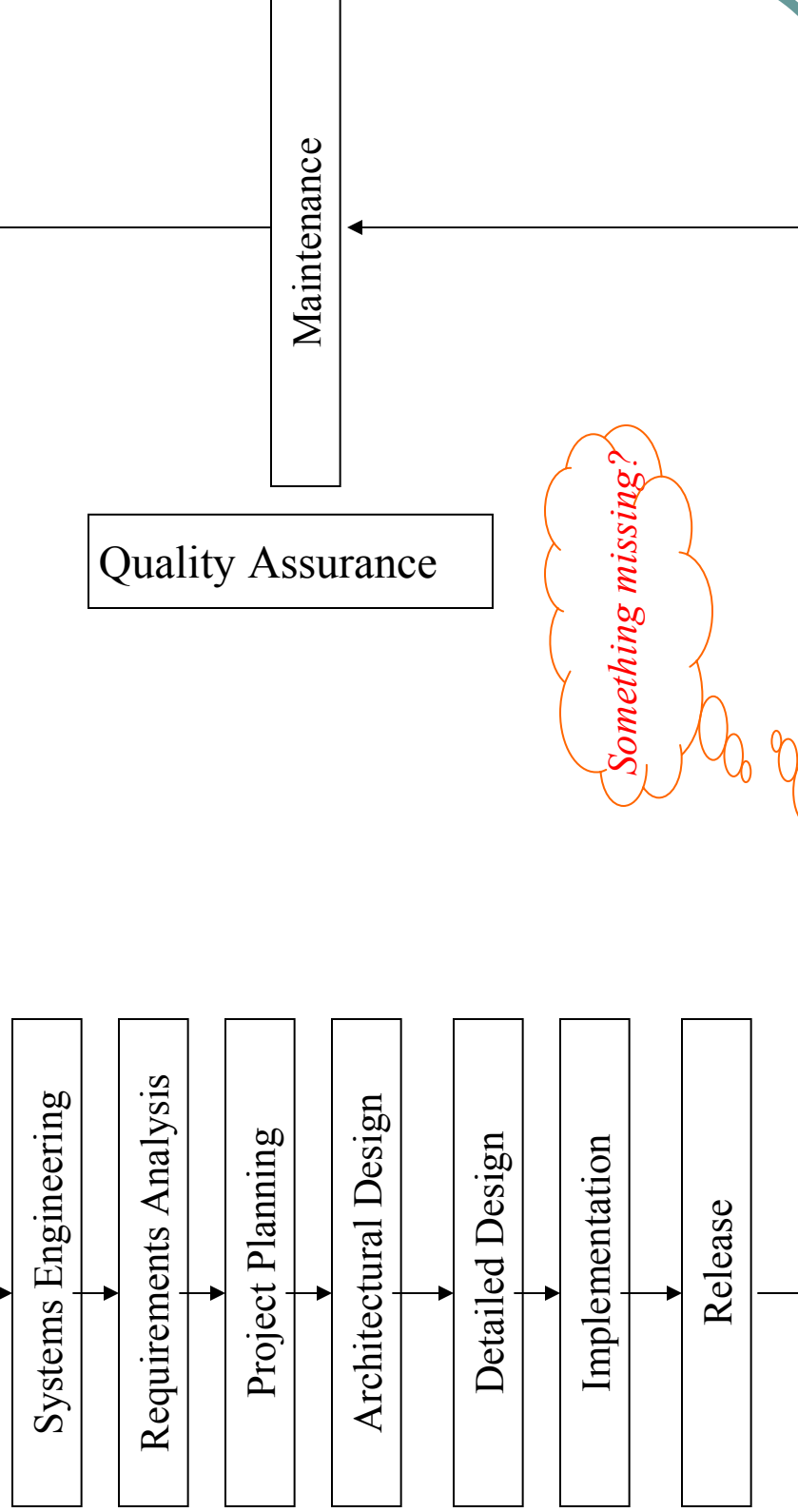


The Old Man and the Sea

How to Do OOAD

– Where to Use OO?

Software Lifecycle



Traceability!

Something missing?

What's yours like?

Introduction to OOAD - Summary

Why

- Once Software Crisis due to Communication and Complexity
- OO for Conceptual Modeling

What

- Fundamental OO Concepts
- A little taste of UML

How

- OO development processes & (Design) Patterns

Points to Ponder

- What are the most important concepts in your project?
- What are the most important concepts in your program?
- Can you reverse engineer a design, DD and HLD?
- What are the most important concepts in your design?
- What are the most important concepts in your PRS, PAS, PDS?

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.

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