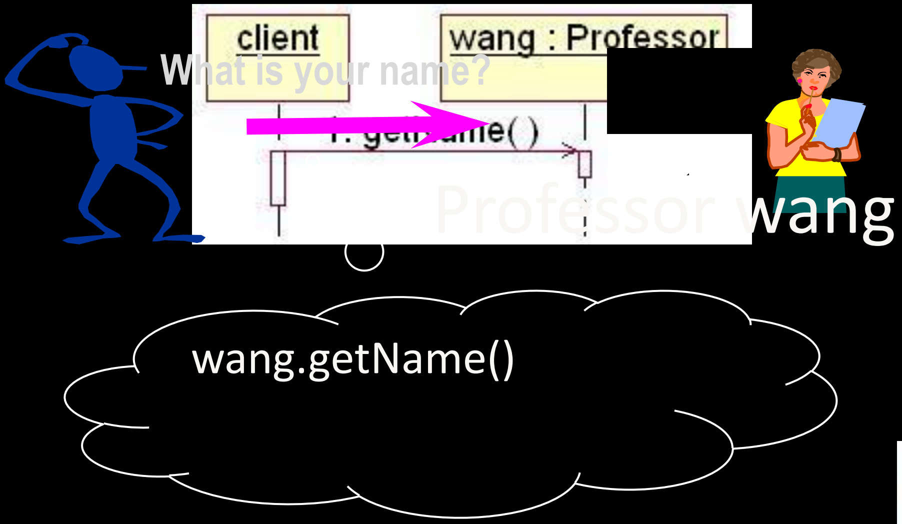
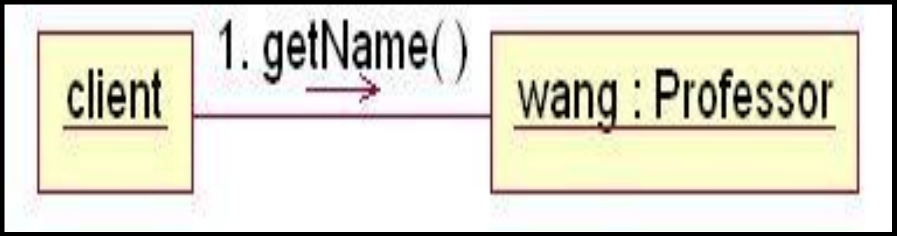
Lecture-3

What is a message?

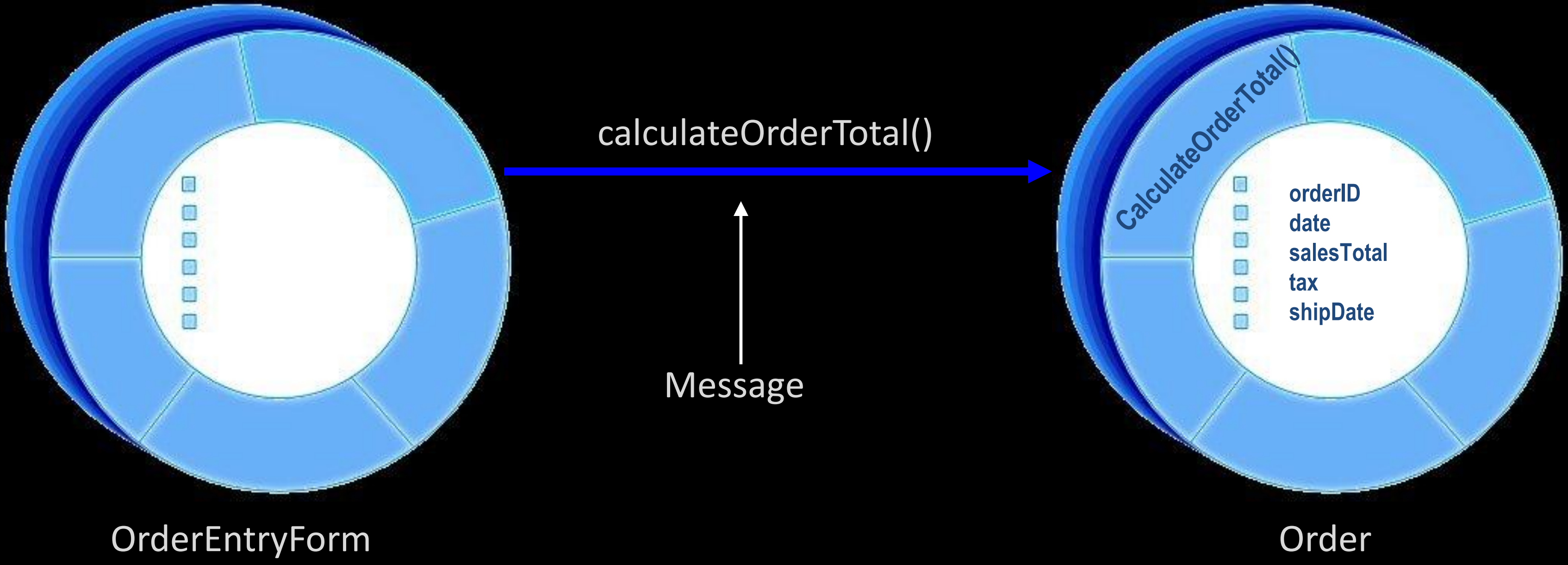
- A specification of a communication between objects that conveys information with the expectation that activity will ensue
  - One object asks another object to perform an operation.





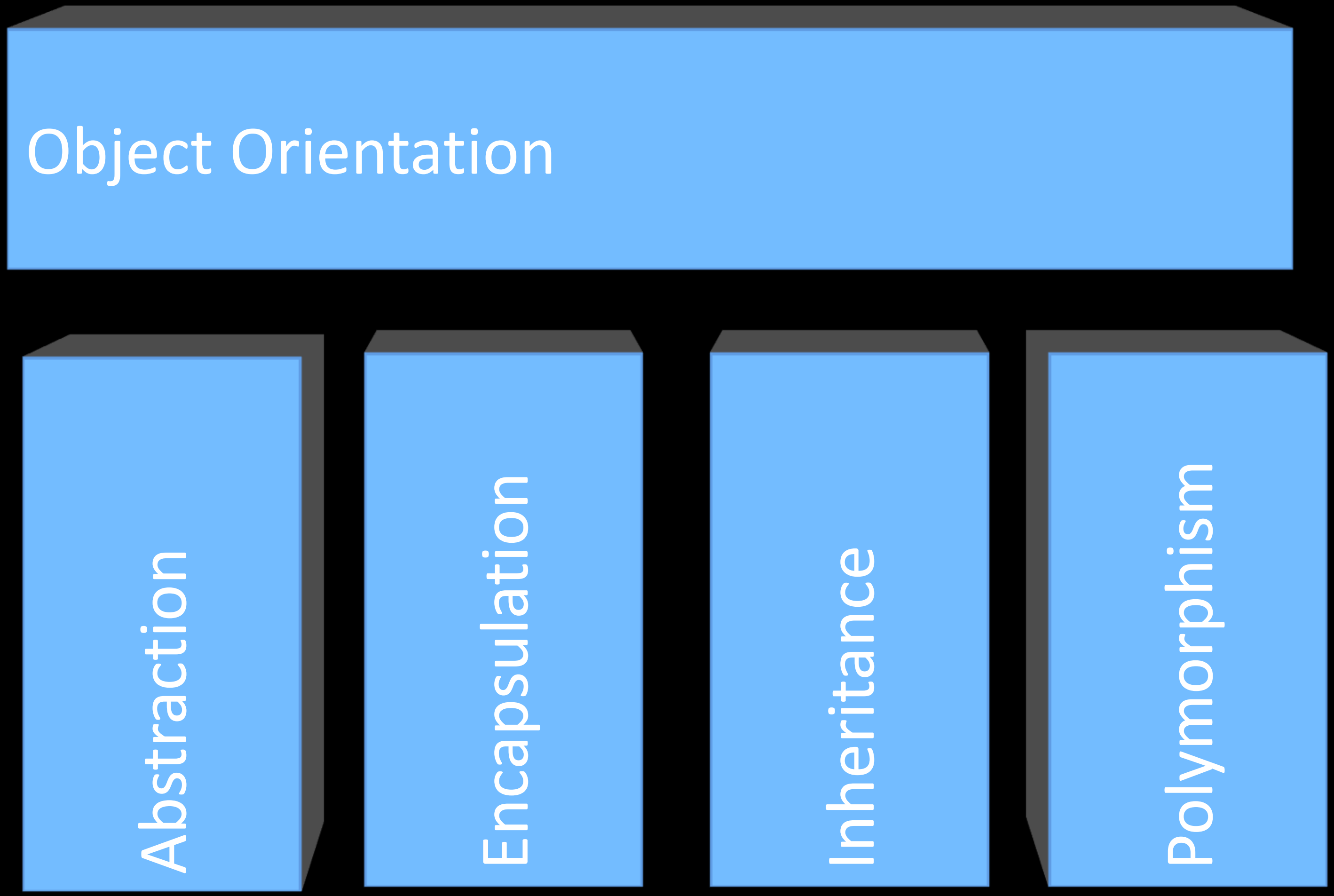
Example: Object Interaction

- The OrderEntryForm wants Order to calculate the total dollar value for the order.



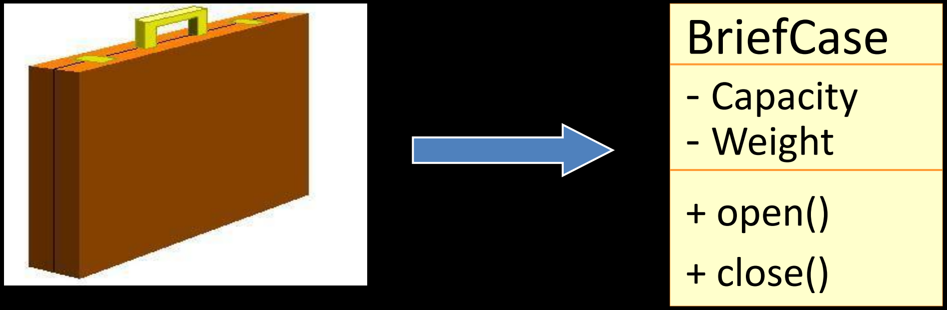
The class Order has the *responsibility* to calculate the total dollar value.

Basic Principles of Object Orientation



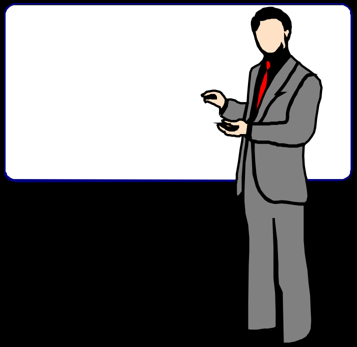
What Is Abstraction?

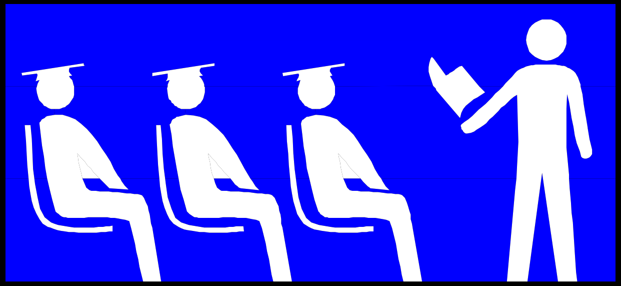
- Abstraction can be defined as:
  - Any model that includes the most important, essential, or distinguishing aspects of something while suppressing or ignoring less important, immaterial, or diversionary details. The result of removing distinctions so as to emphasize commonalties.
    (*Dictionary of Object Technology*, Firesmith, Eykholt, 1995)
- Abstraction
  - Emphasizes relevant characteristics.
  - Suppresses other characteristics.

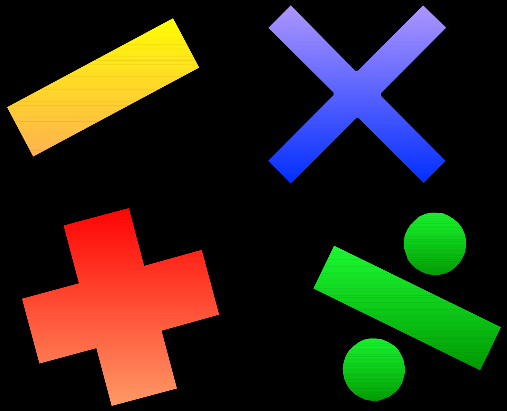


Example: Abstraction

Student


Professor


Course Offering (9:00 AM, Monday-Wednesday-Friday)


Course (e.g. Algebra)

What Is Encapsulation?

- *Encapsulation* means to design, produce, and describe software so that it can be easily used without knowing the details of how it works.
- Also known as *information hiding*

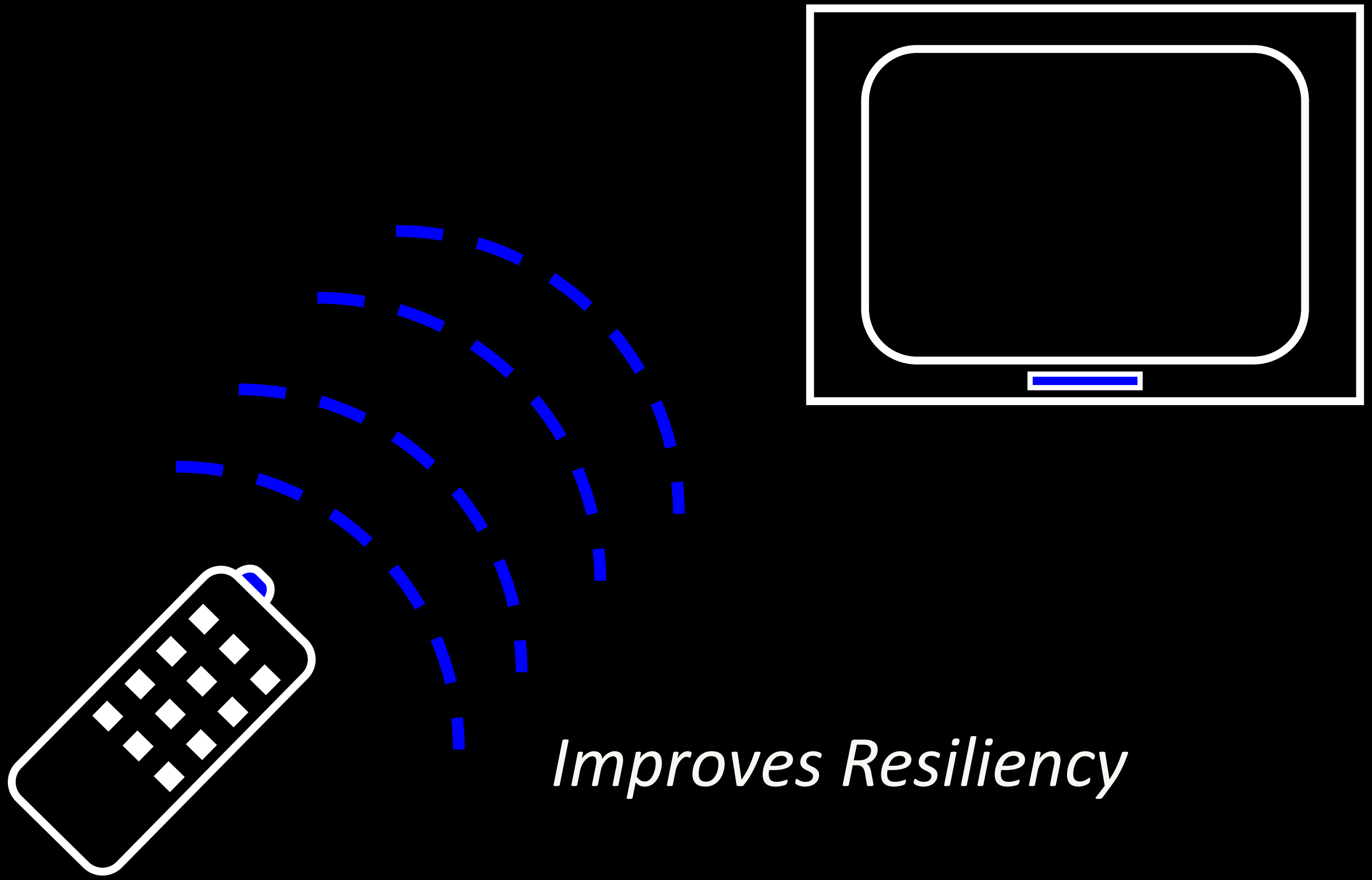
An analogy:

- When you drive a car, you don’t have know the details of how many cylinders the engine has or how the gasoline and air are mixed and ignited.
- Instead you only have to know how to use the controls.

What Is Encapsulation?

Hide implemmentation from clients

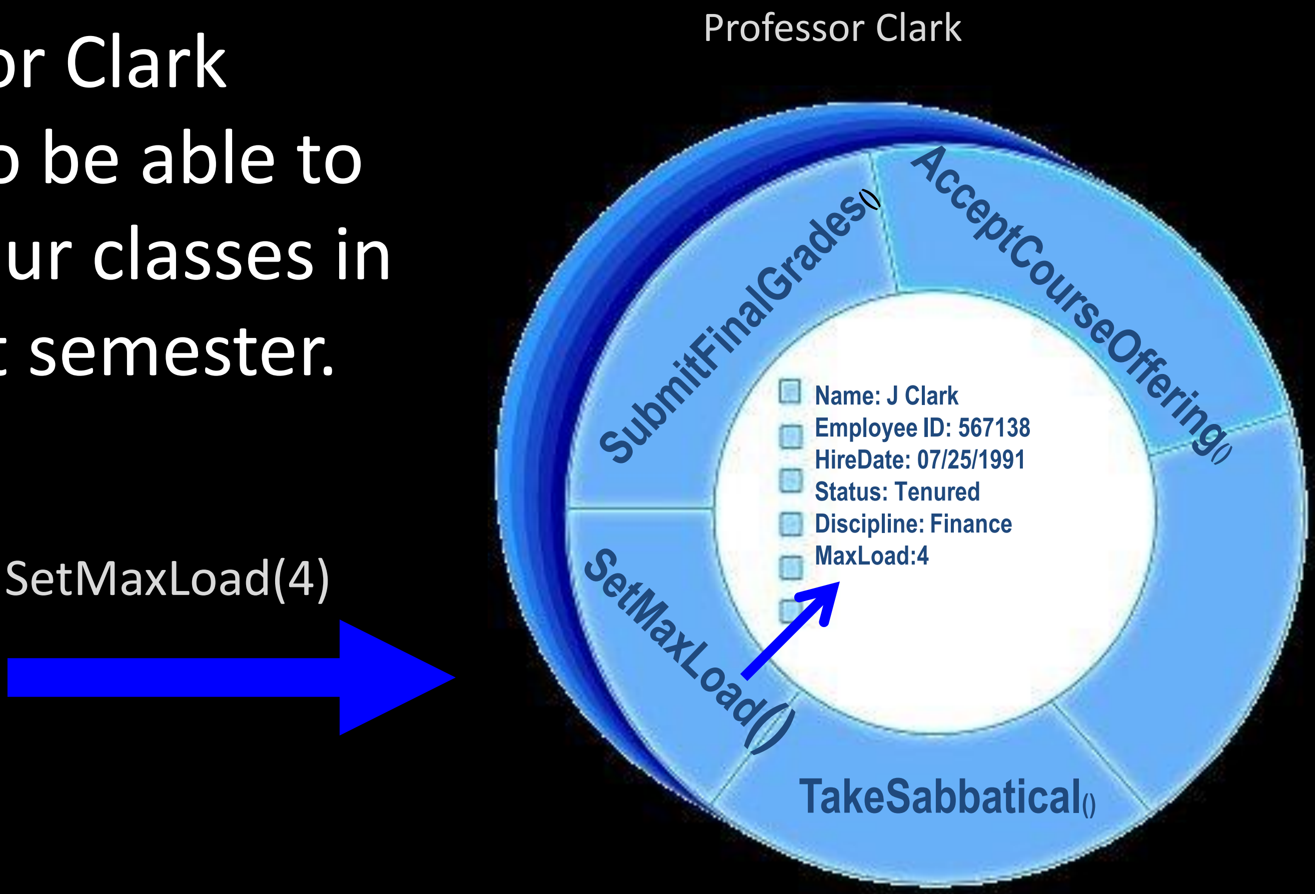
- clients depend on interface



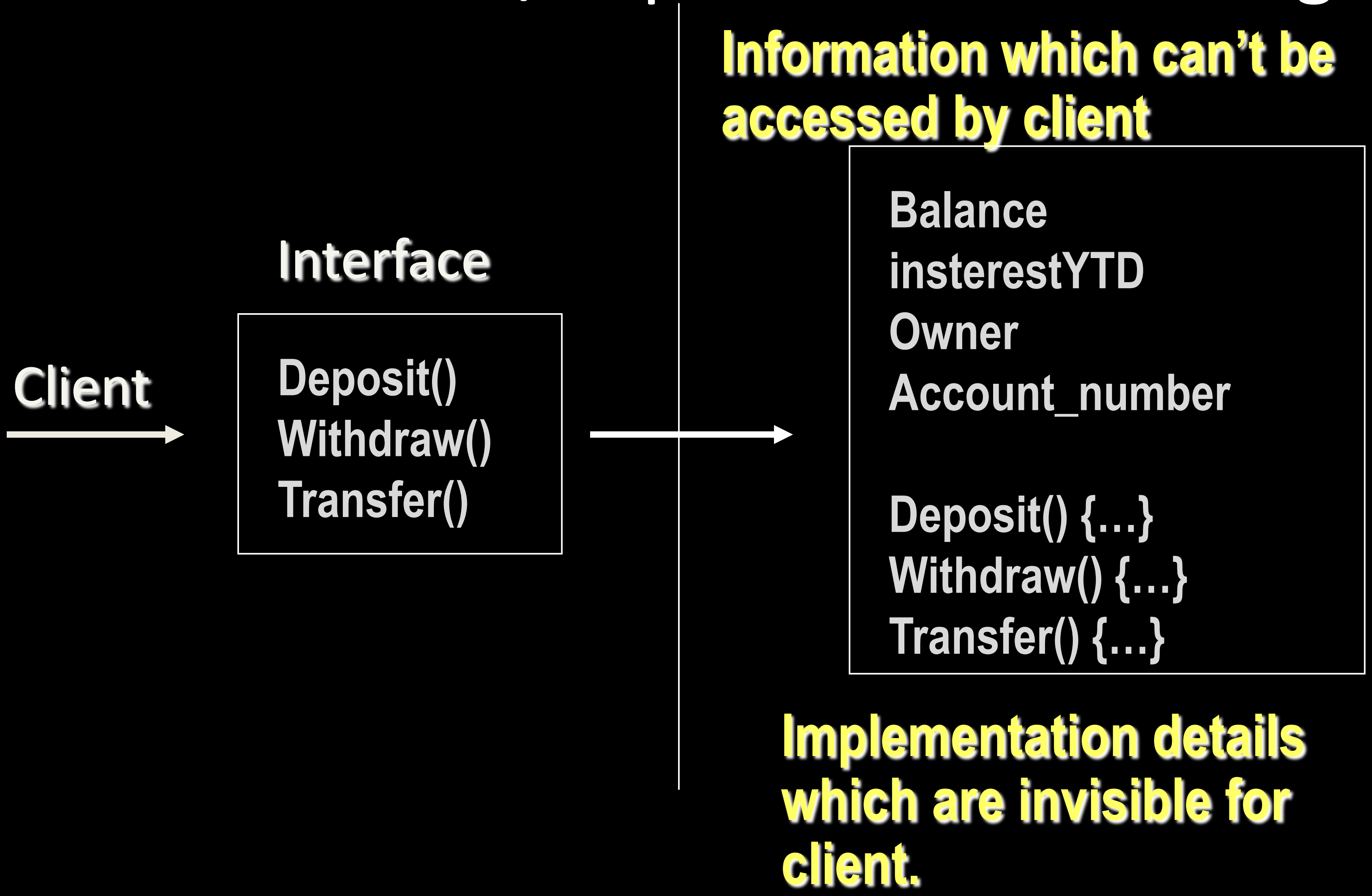
Improves Resiliency

Encapsulation Illustrated

- Professor Clark needs to be able to teach four classes in the next semester.



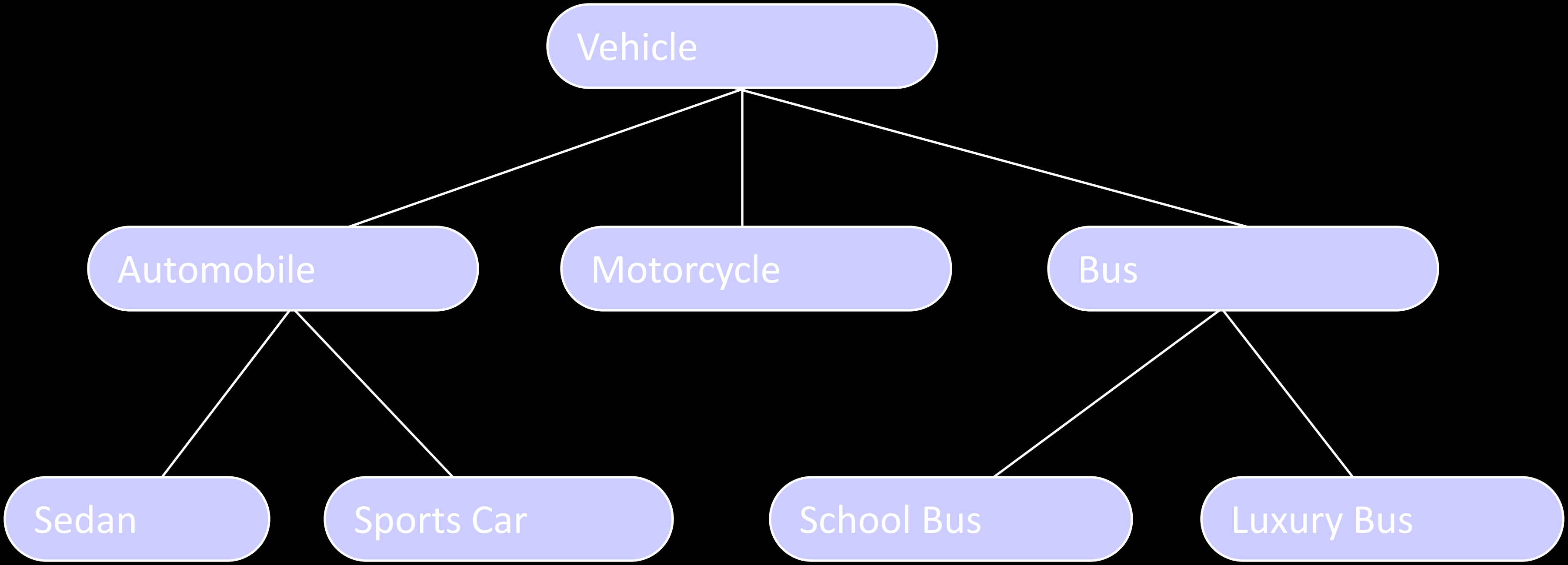
Encapsulation – Information/Implementation hiding



What Is Inheritance ?

- *Inheritance* —a way of organizing classes
- Term comes from inheritance of traits like eye color, hair color, and so on.
- Classes with properties in common can be grouped so that their common properties are only defined once.
- Is an “is a kind of” relationship

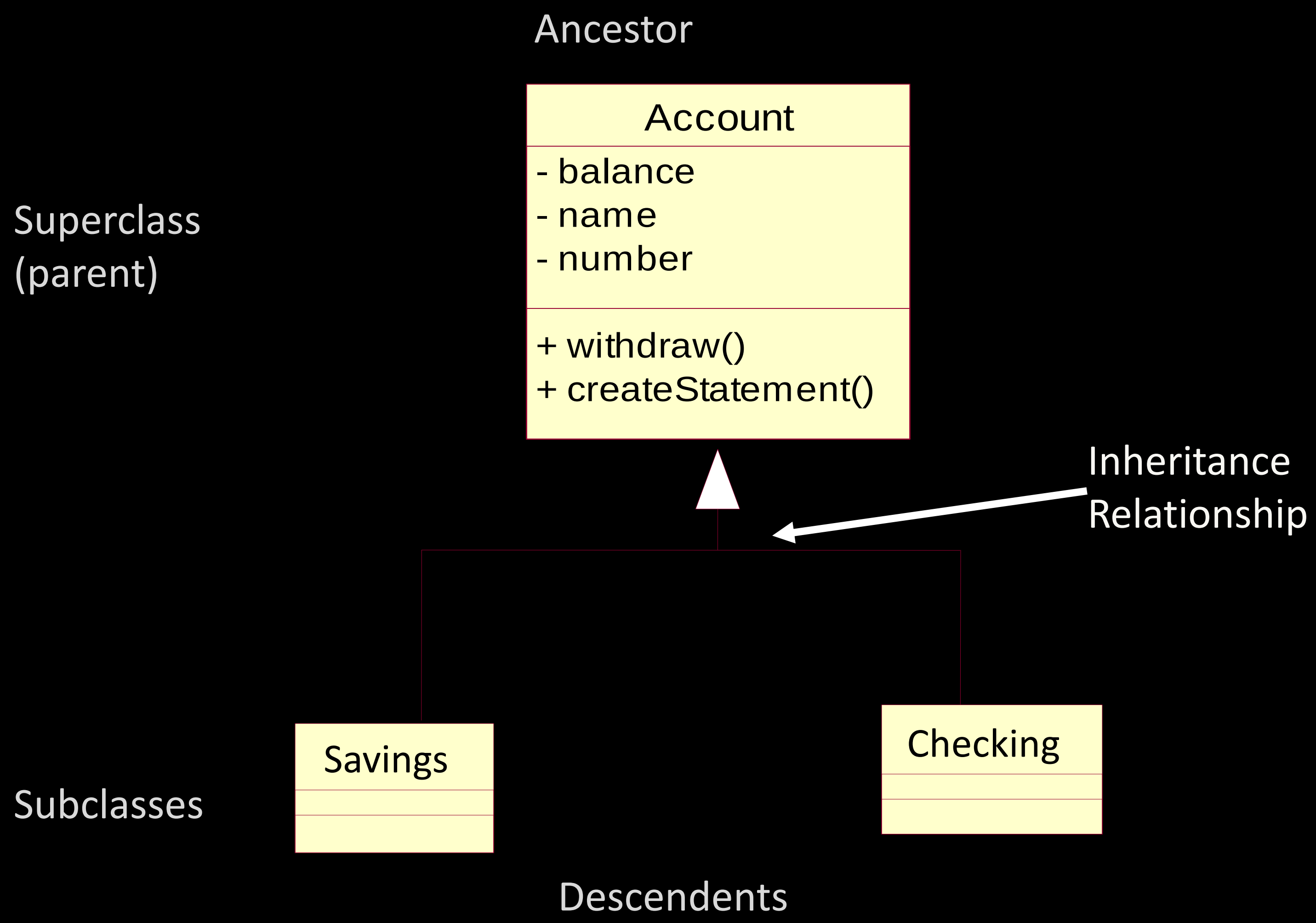
An Inheritance Hierarchy



What properties does each vehicle inherit from the types of vehicles above it in the diagram?

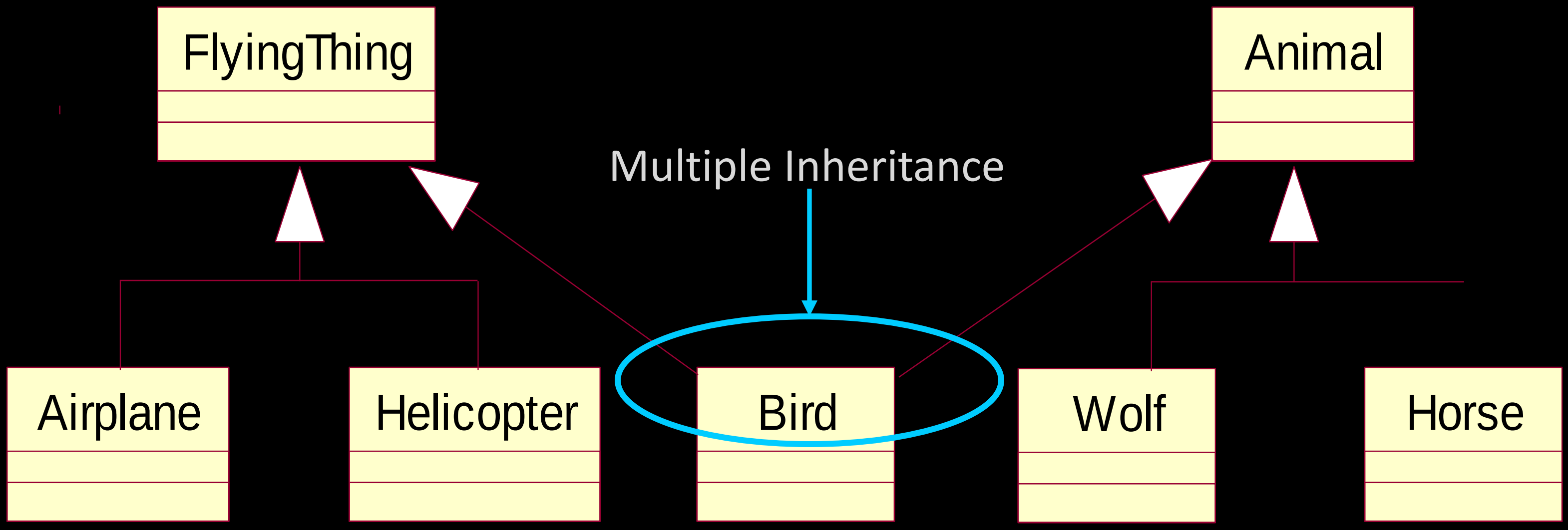
Example: Single Inheritance

- One class inherits from another.



Example: Multiple Inheritance

- A class can inherit from several other classes.



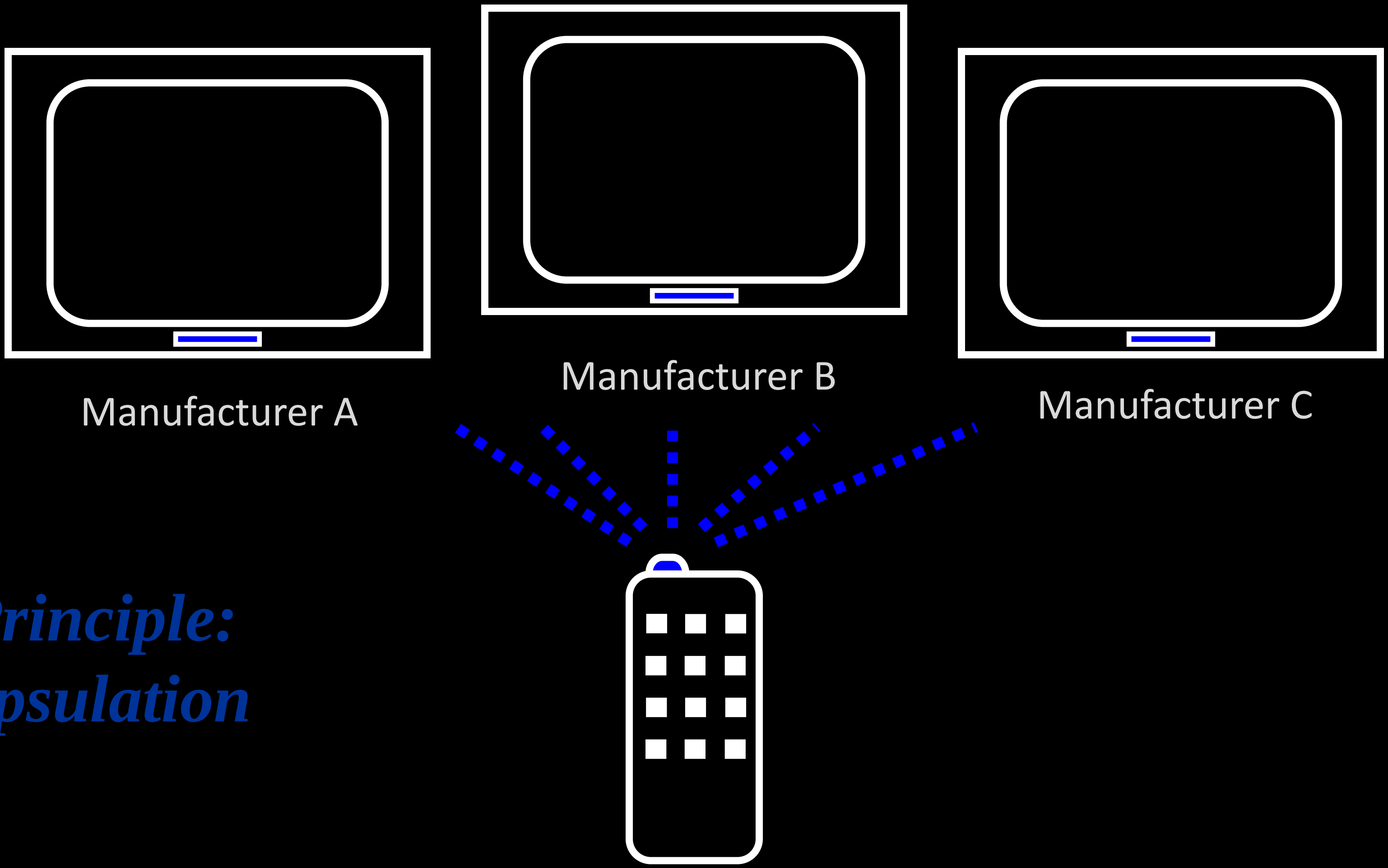
Use multiple inheritance only when needed and always with caution!

Polymorphism

- *Polymorphism*—the same word or phrase can be mean different things in different contexts
- Analogy: in English, **bank** can mean side of a river or a place to put money
- In Java, two or more classes could each have a method called **`output`**
- Each **`output`** method would do the right thing for the class that it was in.
- One **`output`** might display a number whereas a different one might display a name.

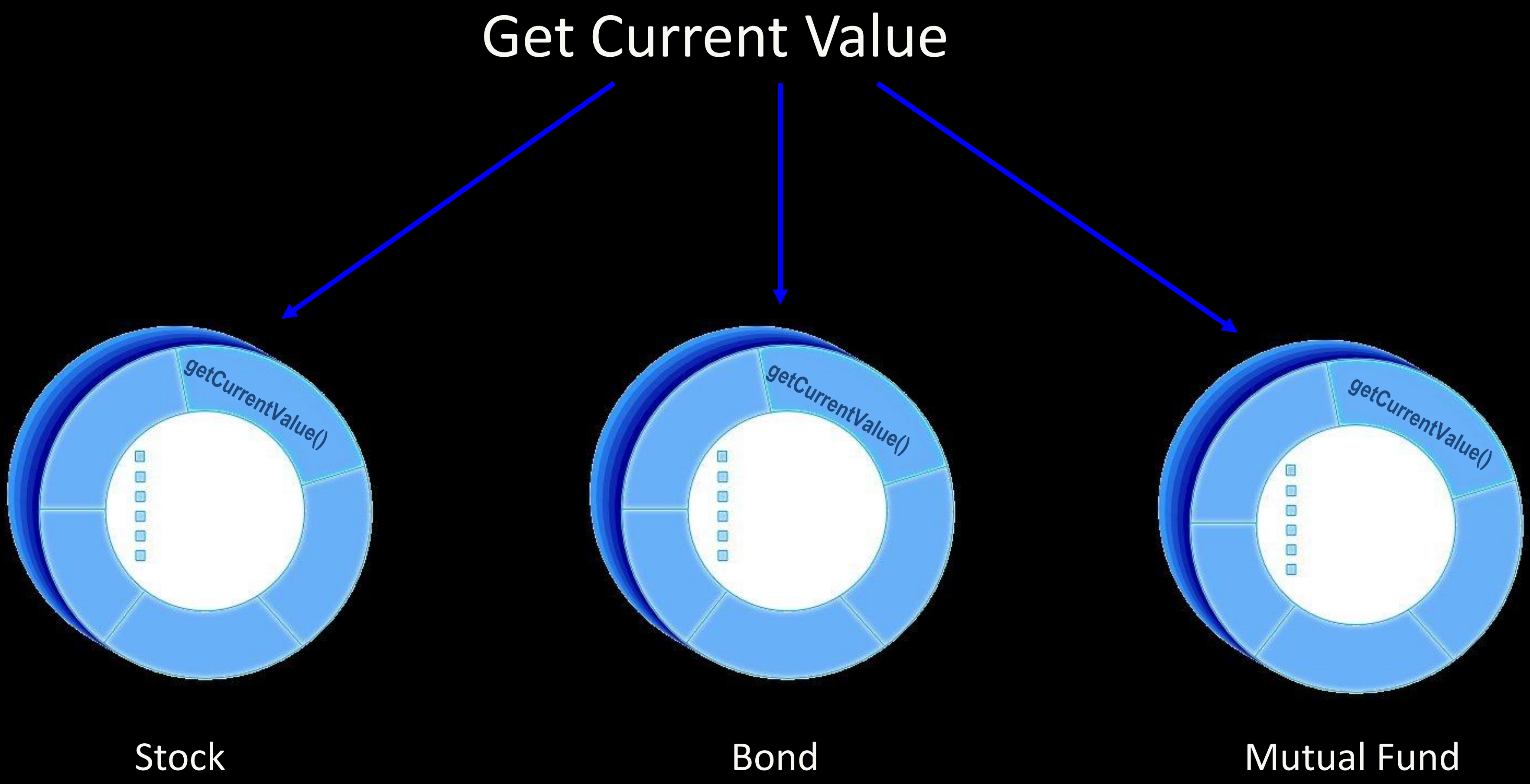
What Is Polymorphism?

- The ability to hide many different implementation behind a single interface.



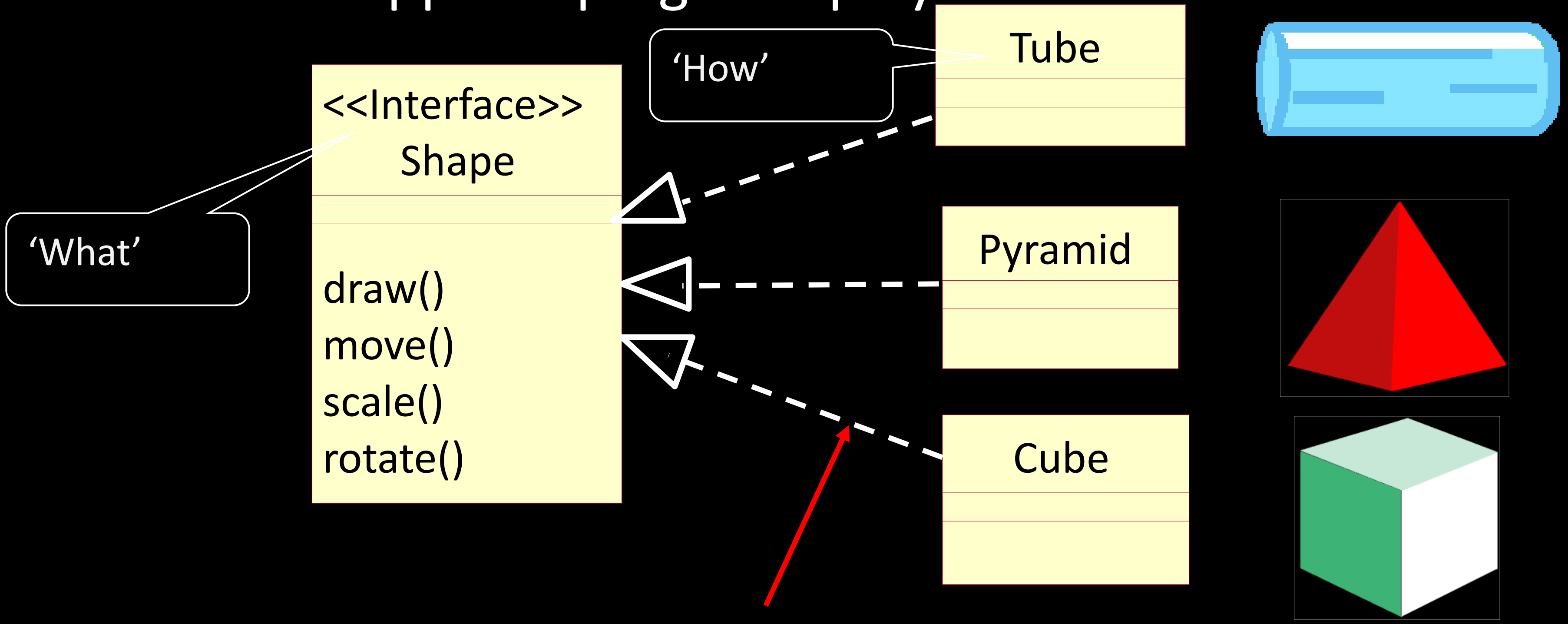
OO Principle: Encapsulation

Example: Polymorphism



What is an Interface?

- An ***interface*** is a collection of operations that specify a service of a class or component.
- Interfaces formalize polymorphism
- Interfaces support "plug-and-play" architectures

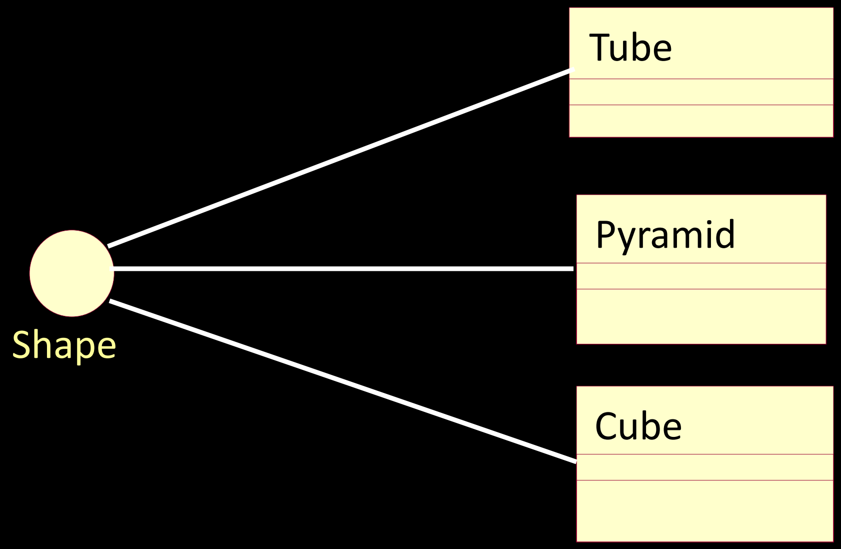


Realization relationship

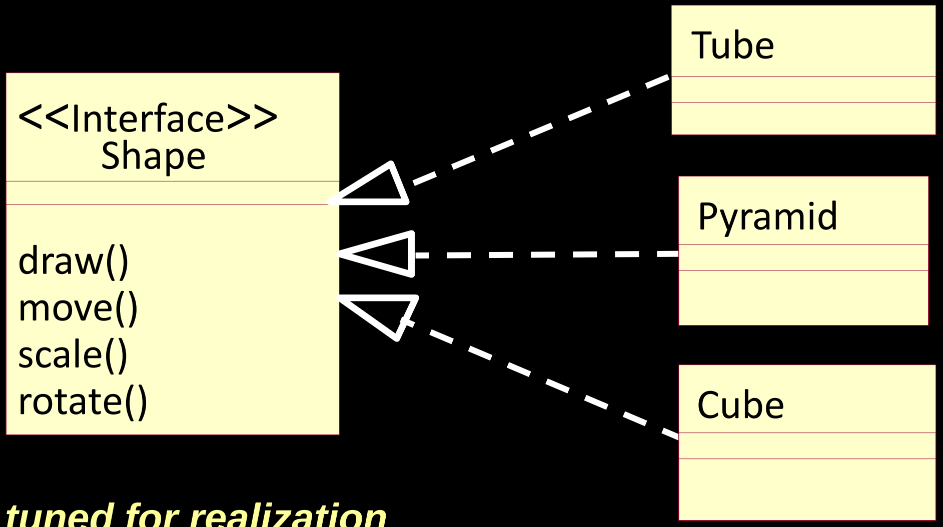
(stay tuned for realization relationships)

How Do You Represent An Interface?

Elided/Iconic Representation ("lollipop")



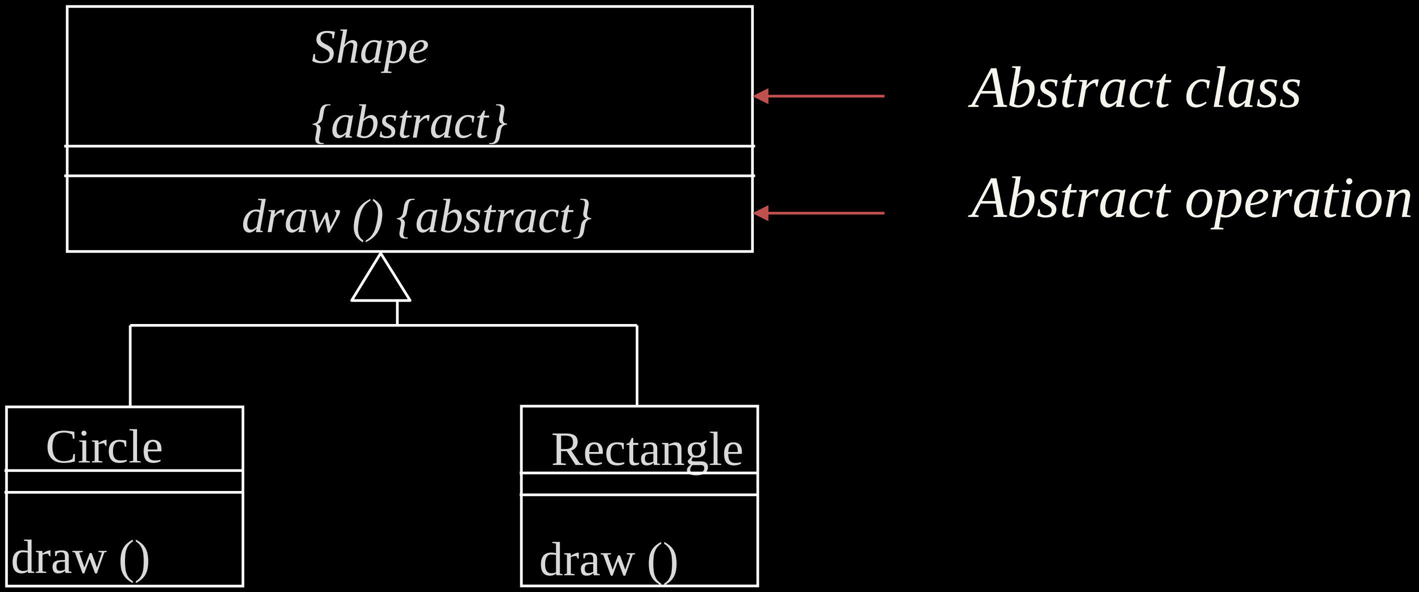
Canonical (Class/Stereotype) Representation



(stay tuned for realization relationships)

What is an Abstract Class?

- An ***abstract class*** is a class that may not has any direct instances.
- In the UML, you specify that a class is abstract by writing its name in italics.
- An ***abstract operation*** is an operation that it is incomplete and requires a child to supply an implementation of the operation.
- In the UML, you specify an abstract operation by writing its name in italics.


Shape
{abstract}
draw () {abstract}
Circle
draw ()
Rectangle
draw ()
Abstract class
Abstract operation