Essentials of Visual Modeling with UML

Module 3: Concepts of Object Orientation

Objectives

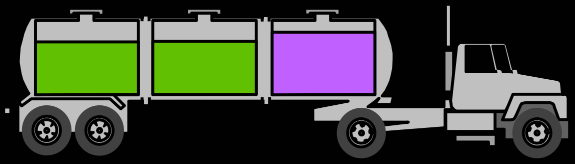
- Describe abstraction, encapsulation, modularity, and hierarchy.
- Describe the physical structure of a class.
- Describe the relationship between a class and an object.
- Define polymorphism and generalization.

Where Are We?

- ★ What is an object?
- Four principles of OO
- What is a class?
- Polymorphism and generalization
- Organizing model elements

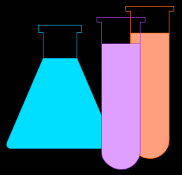
What Is an Object?

- Informally, an object represents an entity, either physical, conceptual, or software.
  - Physical entity



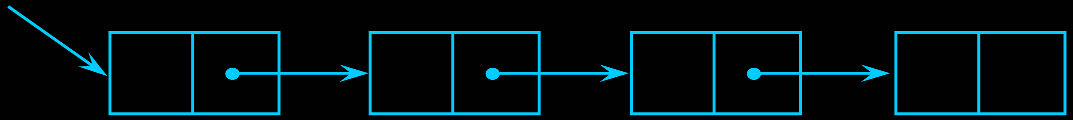
Truck

  - Conceptual entity



Chemical Process

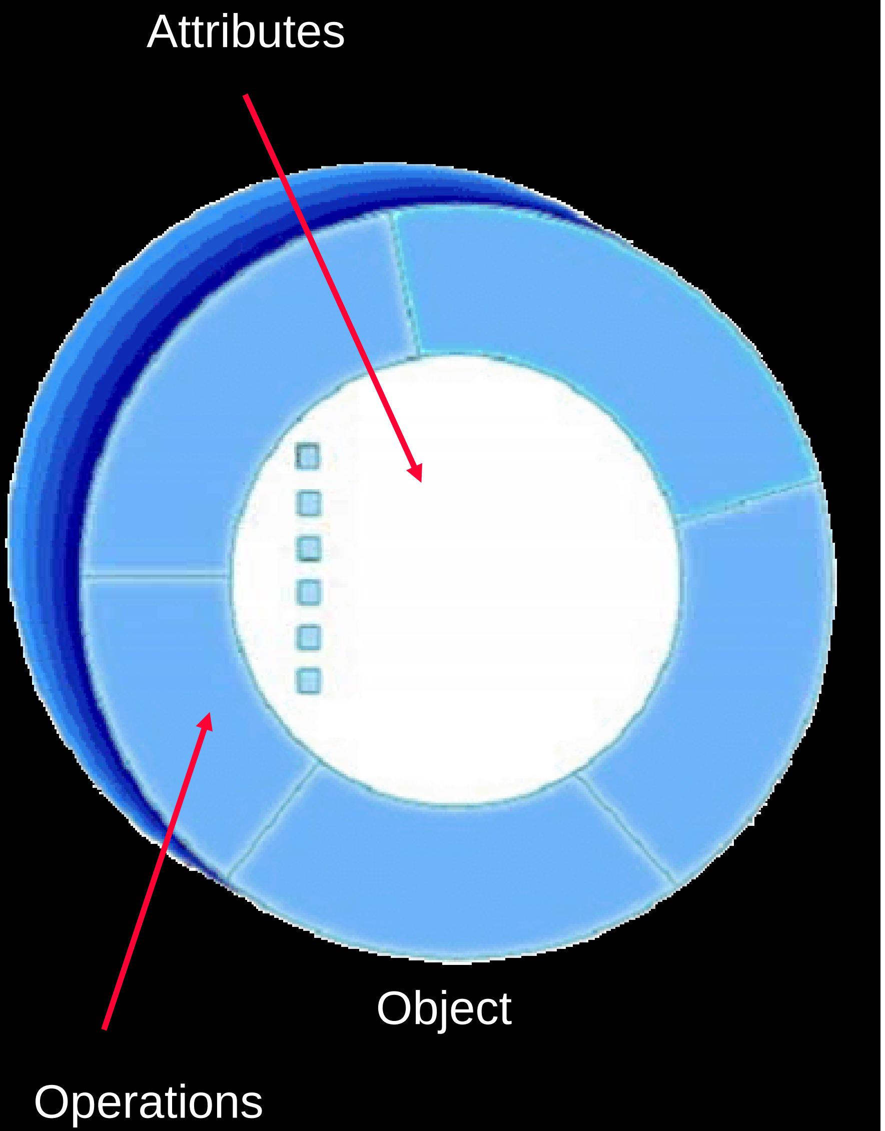
  - Software entity



Linked List

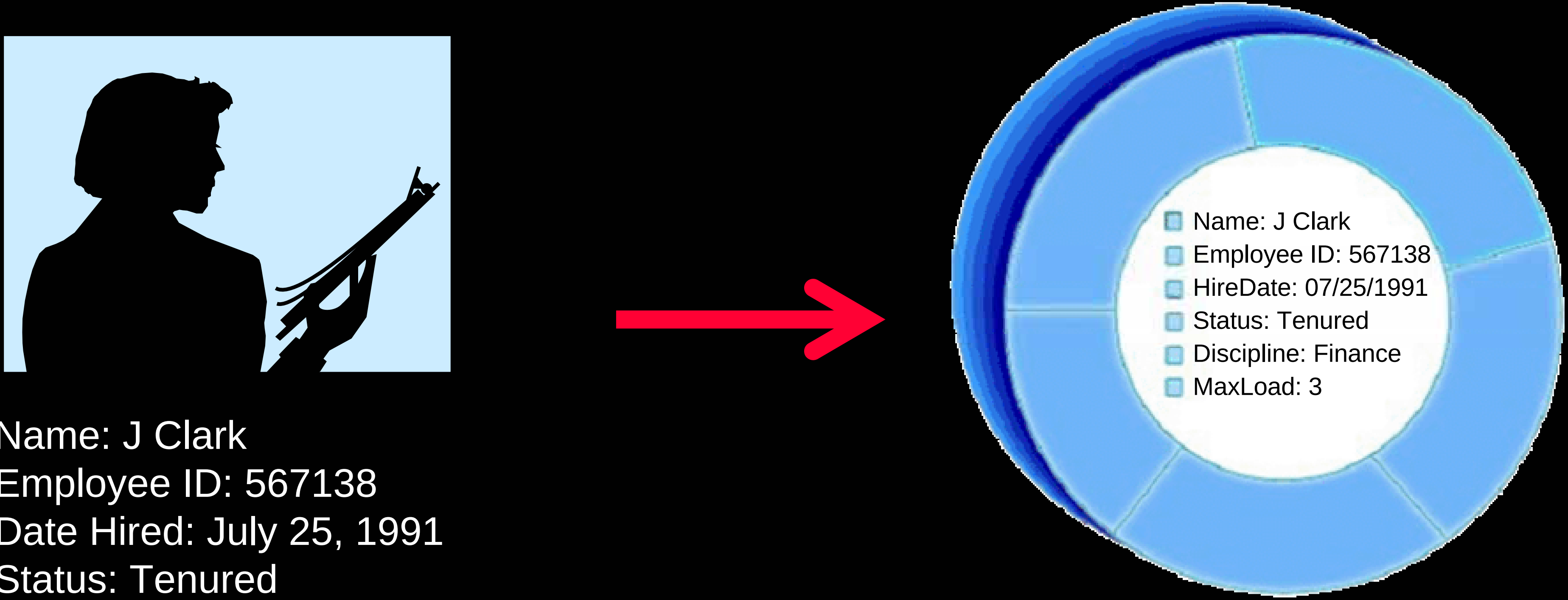
A More Formal Definition

- An object is an entity with a well-defined boundary and *identity* that encapsulates *state* and *behavior.*
  - State is represented by attributes and relationships.
  - Behavior is represented by operations, methods, and state machines.



An Object Has State

- State is a condition or situation during the life of an object, which satisfies some condition, performs some activity, or waits for some event.
- The state of an object normally changes over time.



- ▢ Name: J Clark
- ▢ Employee ID: 567138
- ▢ HireDate: 07/25/1991
- ▢ Status: Tenured
- ▢ Discipline: Finance
- ▢ MaxLoad: 3

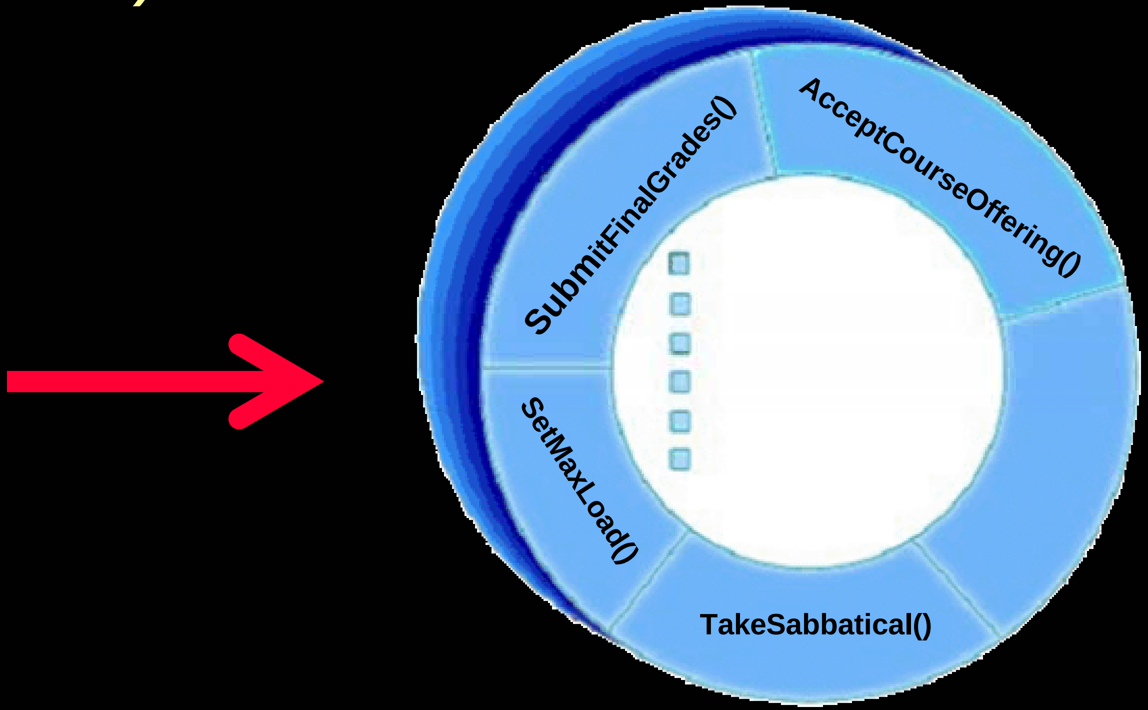
Name: J Clark
Employee ID: 567138
Date Hired: July 25, 1991
Status: Tenured
Discipline: Finance
Maximum Course Load: 3 classes

Professor Clark

An Object Has Behavior

- Behavior determines how an object acts and reacts.
- The visible behavior of an object is modeled by a set of messages it can respond to (operations that the object can perform).

Professor Clark’s behavior
Submit Final Grades
Accept Course Offering
Take Sabbatical
Maximum Course Load: 3 classes



Professor Clark

An Object Has Identity

- Each object has a unique identity, even if the state is identical to that of another object.

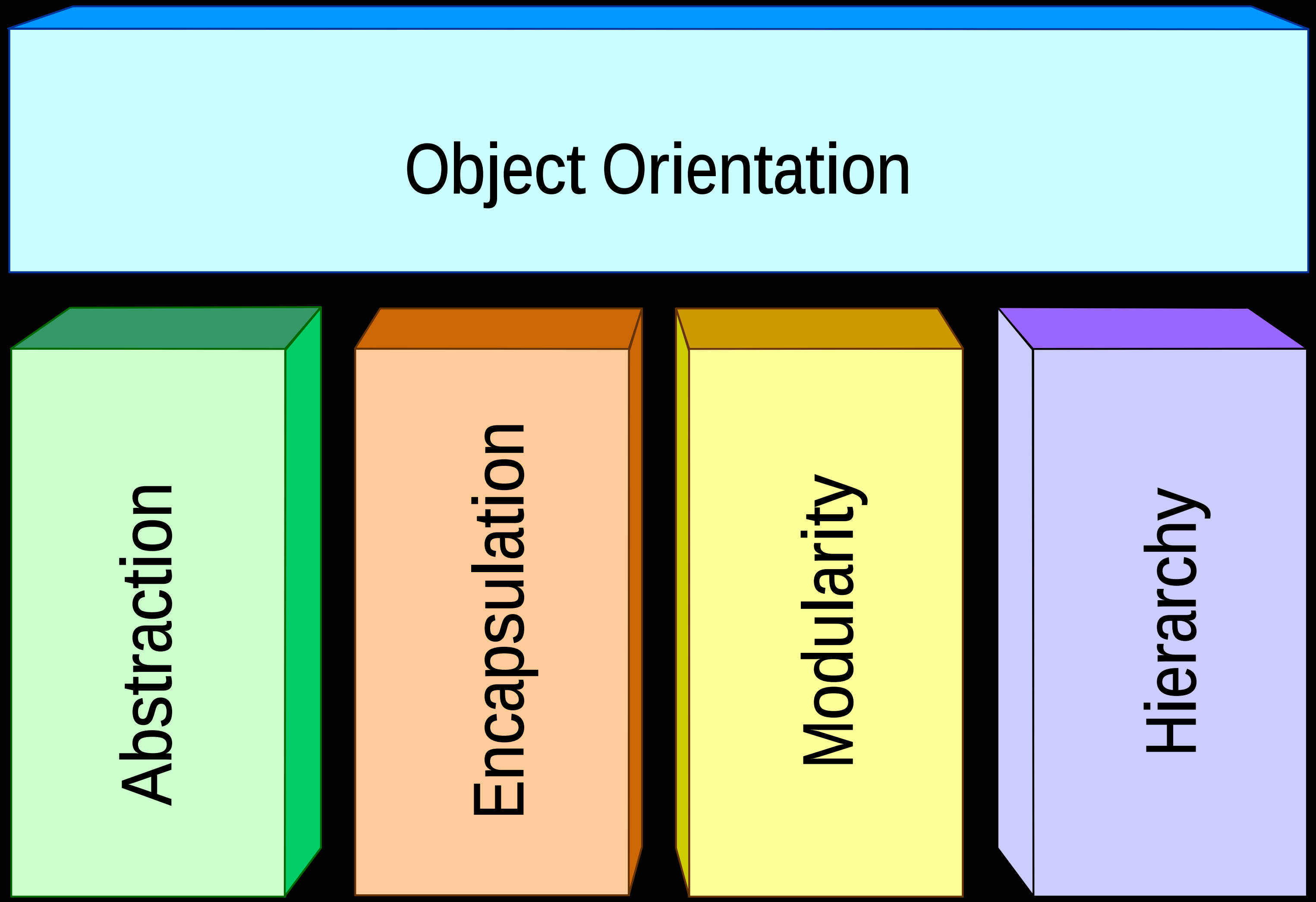
Professor "J Clark" teaches Biology

Professor "J Clark" teaches Biology

Where Are We?

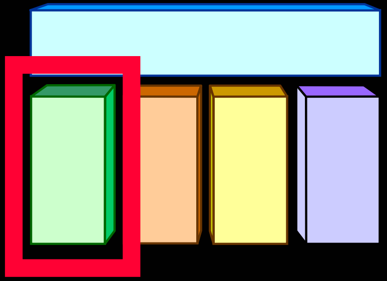
- What is an object?
- ★ Four principles of OO
- What is a class?
- Polymorphism and generalization
- Organizing model elements

Basic Principles of Object Orientation



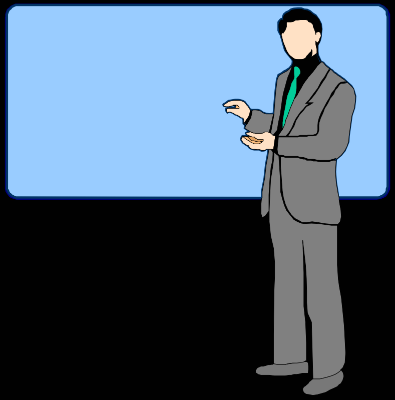
What Is Abstraction?

- The essential characteristics of an entity that distinguishes it from all other kinds of entities.
- Defines a boundary relative to the perspective of the viewer.
- Is not a concrete manifestation, denotes the ideal essence of something.



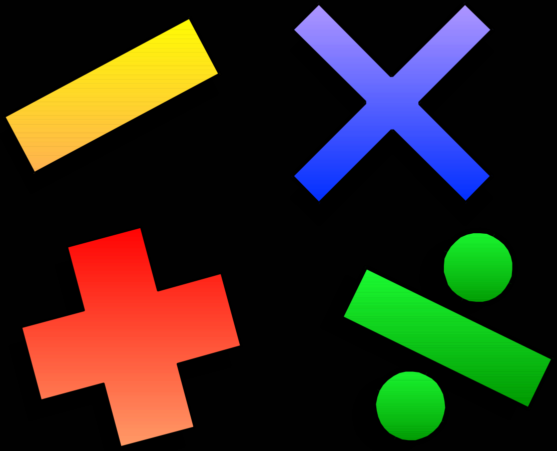
Example: Abstraction

Student



Professor

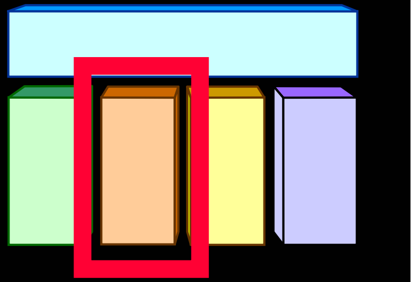
Course Offering (9:00 a.m.,
Monday-Wednesday-Friday)

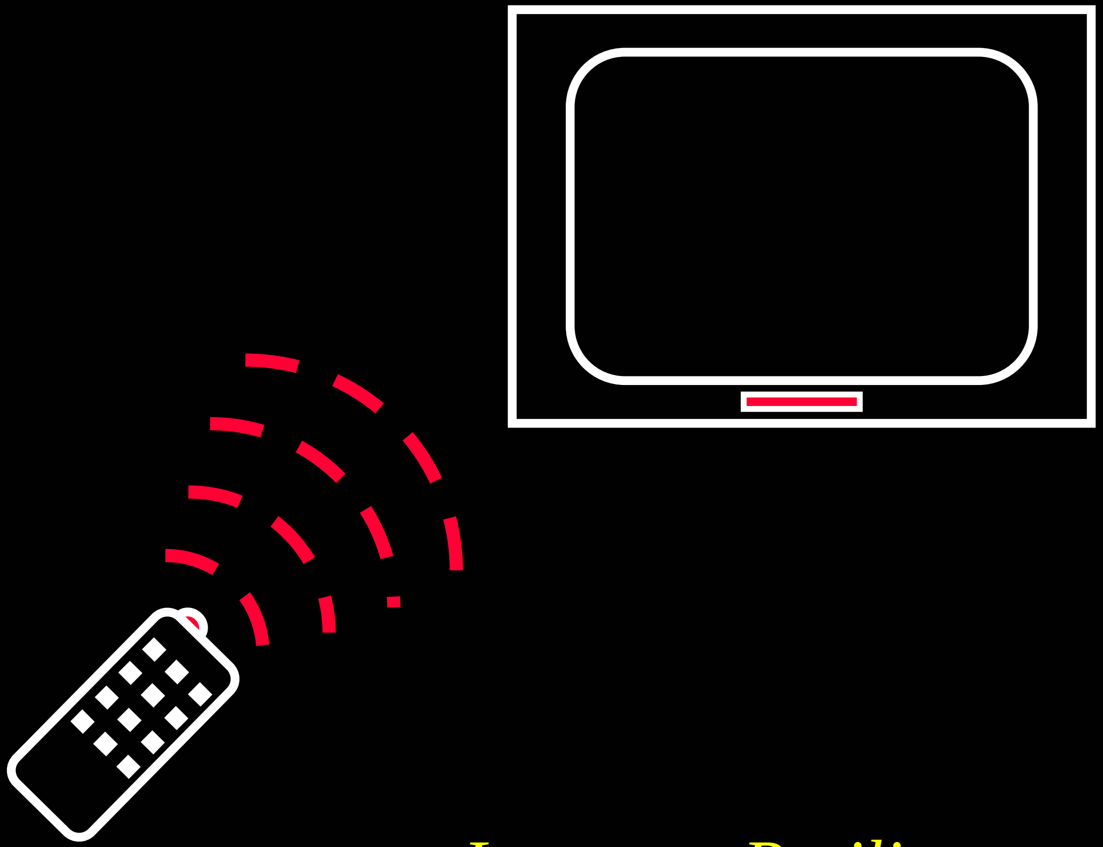


Course (e.g. Algebra)

What Is Encapsulation?

- Hides implementation from clients.
  - Clients depend on interface.

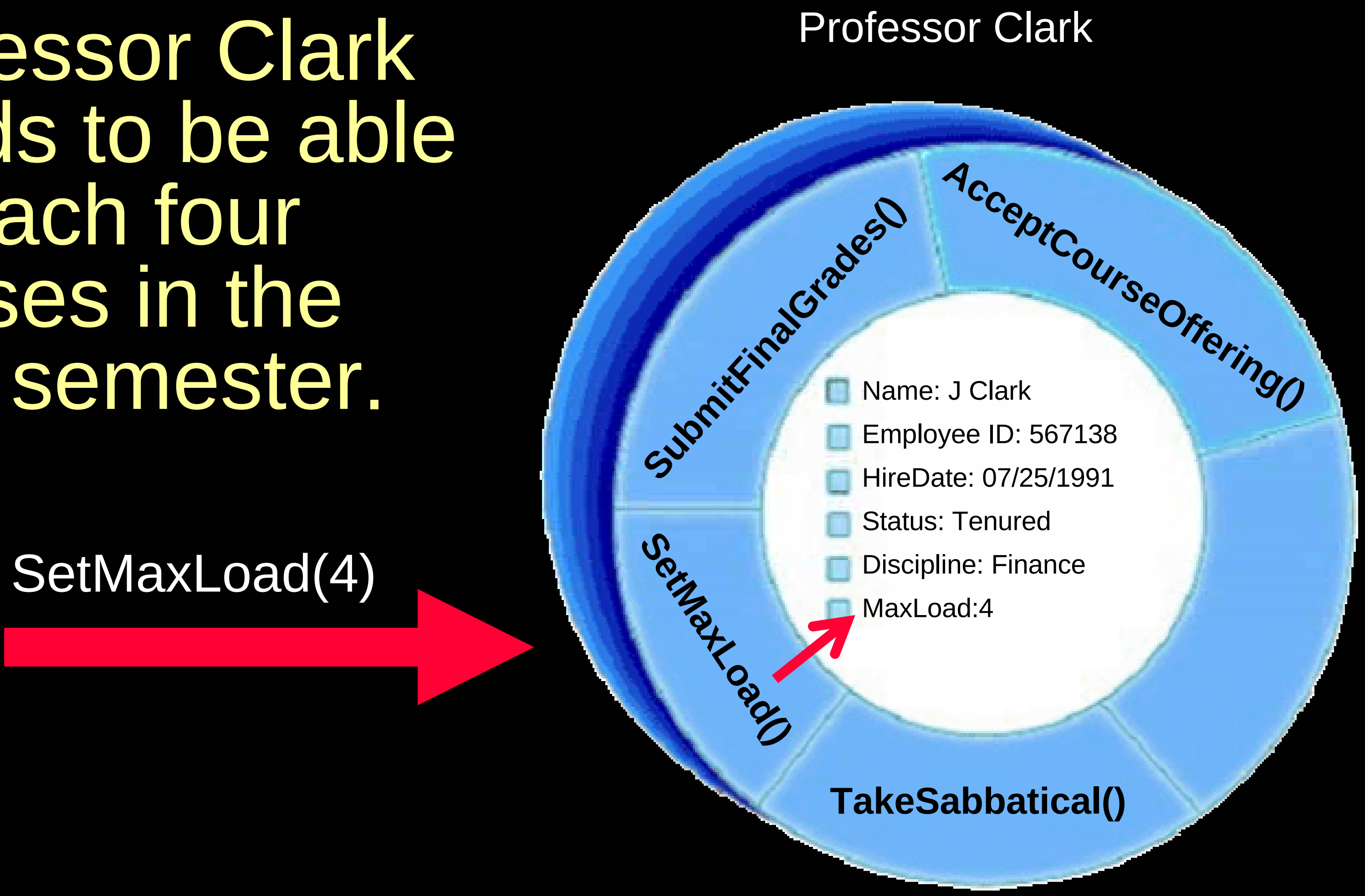




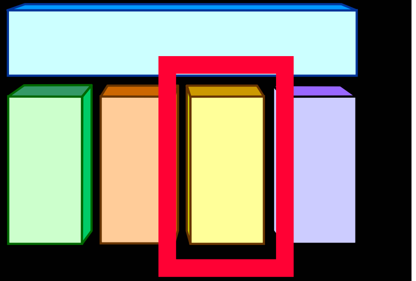
Improves Resiliency

Encapsulation Illustrated

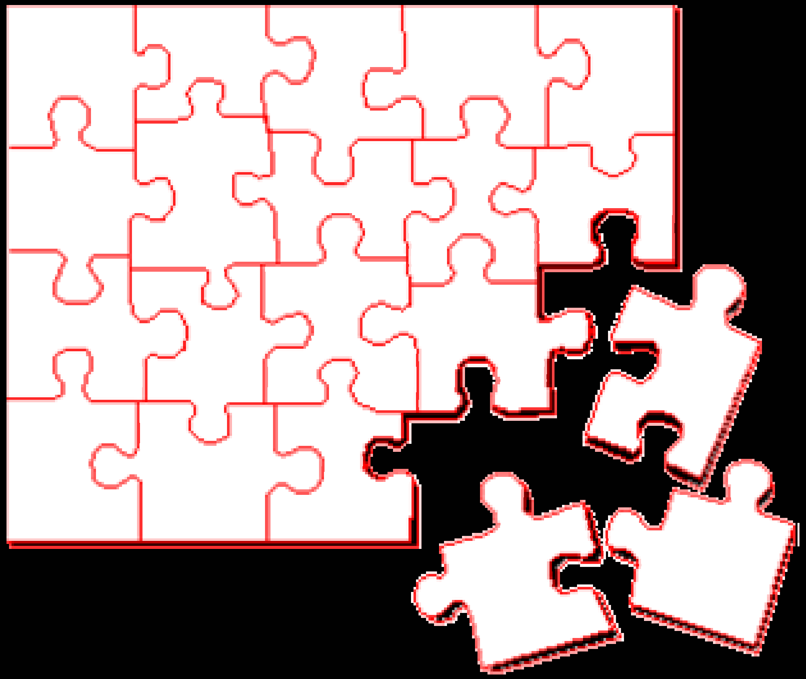
- Professor Clark needs to be able to teach four classes in the next semester.



What Is Modularity?

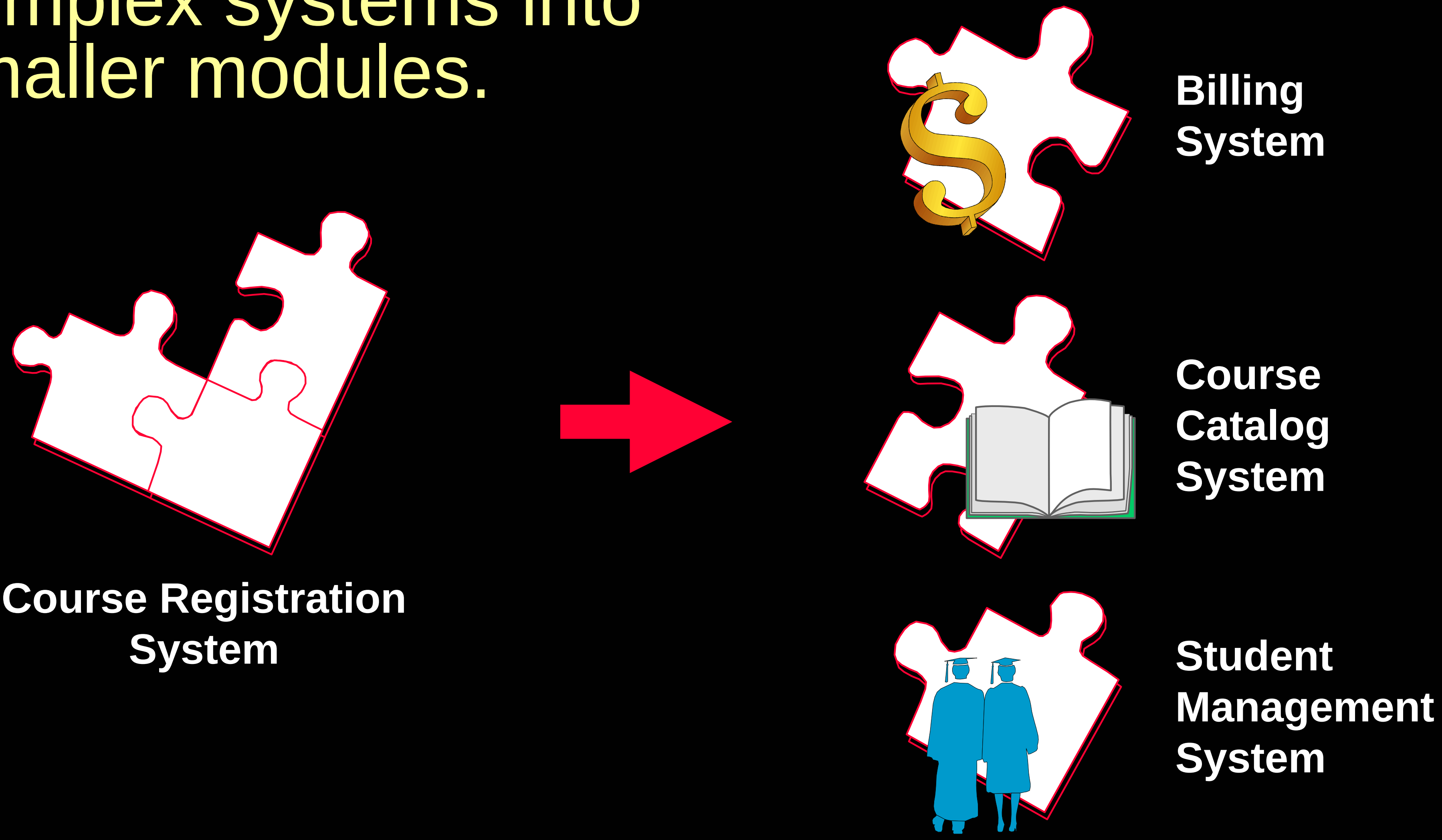


- Breaks up something complex into manageable pieces.
- Helps people understand complex systems.



Example: Modularity

- For example, break complex systems into smaller modules.



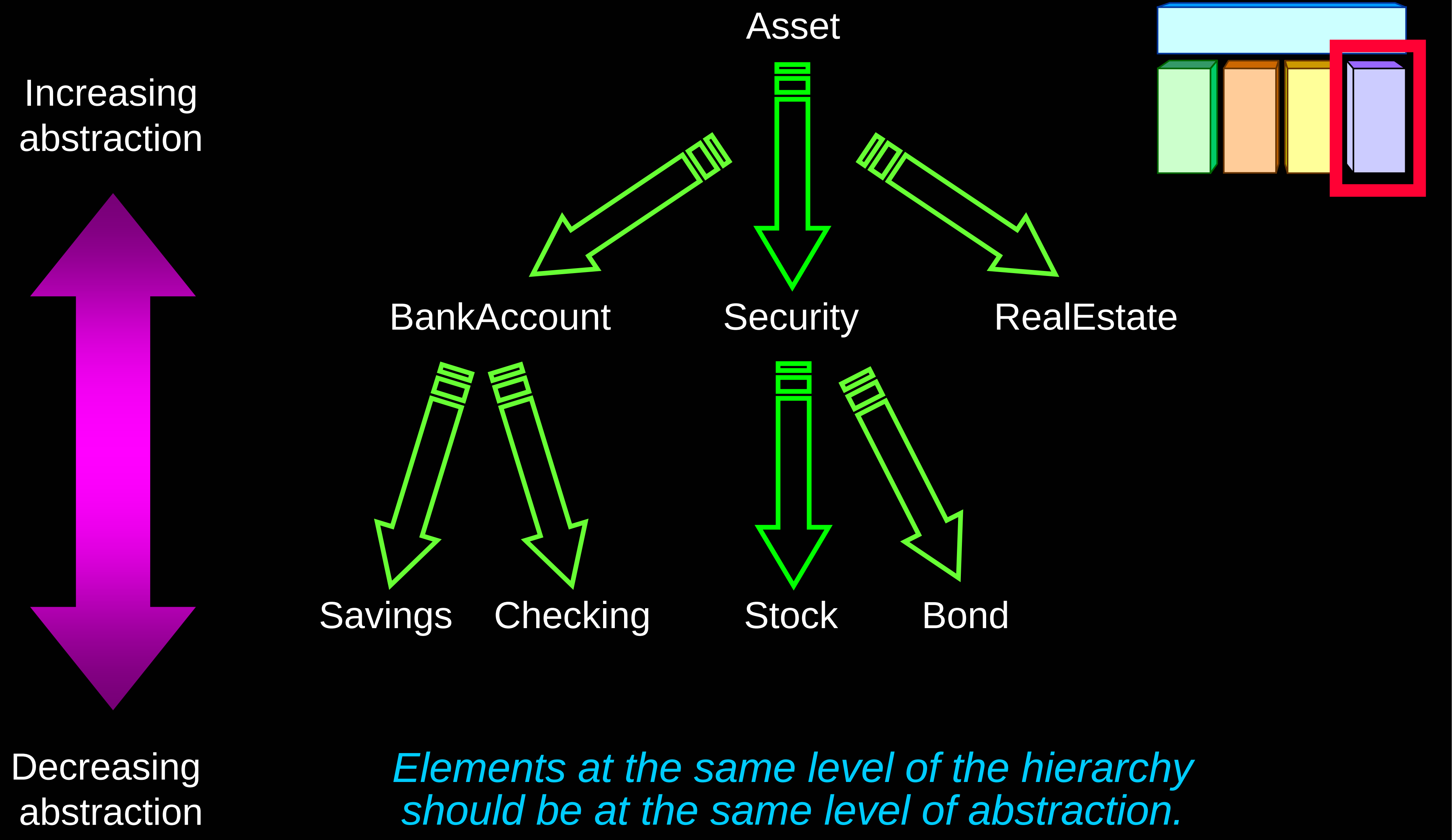
Course Registration System

Billing System

Course Catalog System

Student Management System

What Is Hierarchy?

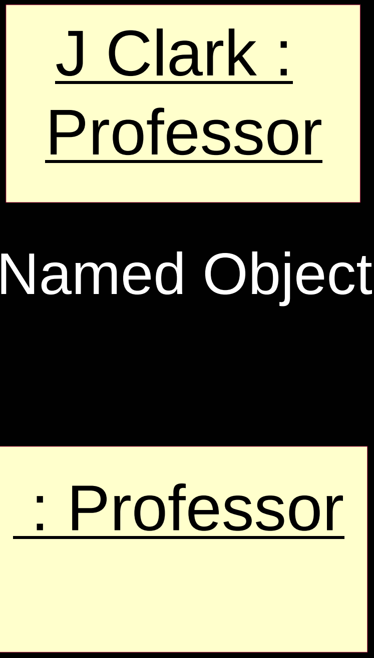


Elements at the same level of the hierarchy should be at the same level of abstraction.

Representing Objects in the UML

- An object is represented as a rectangle with an underlined name.

Professor J Clark



Anonymous Object

Where Are We?

- What is an object?
- Four principles of OO
- ★ What is a class?
- Polymorphism and generalization
- Organizing model elements

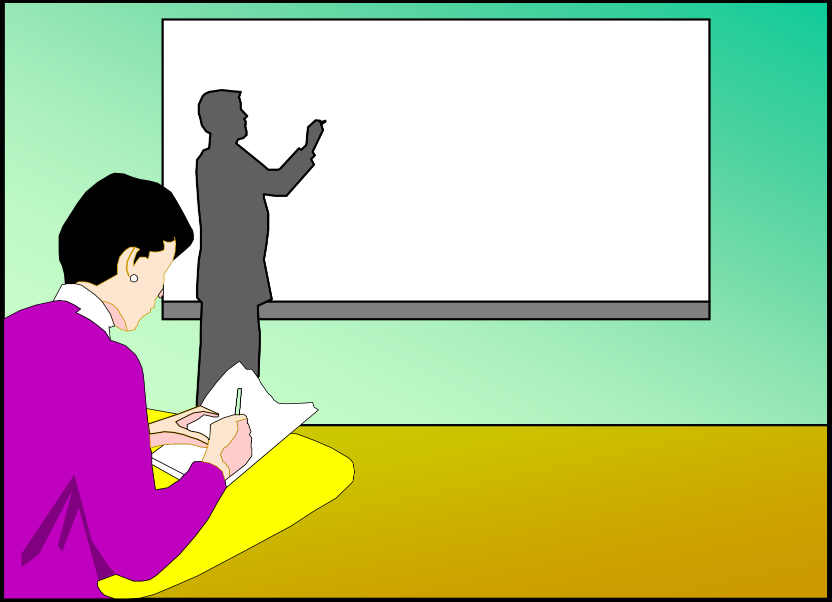
What Is a Class?

- A class is a description of a set of objects that share the same *attributes*, *operations*, *relationships*, and *semantics*.
  - An object is an instance of a class.
- A class is an abstraction in that it
  - Emphasizes relevant characteristics.
  - Suppresses other characteristics.

A Sample Class

Class
Course

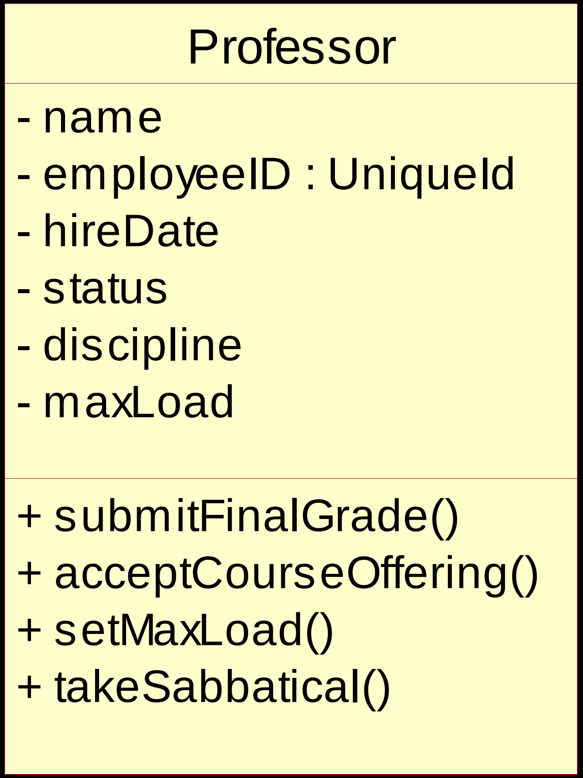
Properties
Name
Location
Days offered
Credit hours
Start time
End time



Behavior
Add a student
Delete a student
Get course roster
Determine if it is full

Representing Classes in the UML

- A class is represented using a rectangle with compartments.



Professor

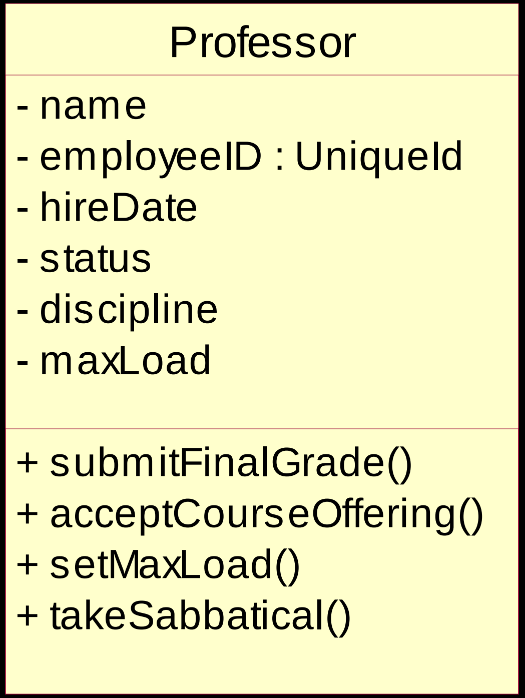
- name
- employeeID : UniqueId
- hireDate
- status
- discipline
- maxLoad

+ submitFinalGrade()
+ acceptCourseOffering()
+ setMaxLoad()
+ takeSabbatical()

Professor J Clark

Class Compartments

- A class has three sections:
  - The class name
  - The structure (attributes)
  - The behavior (operations)



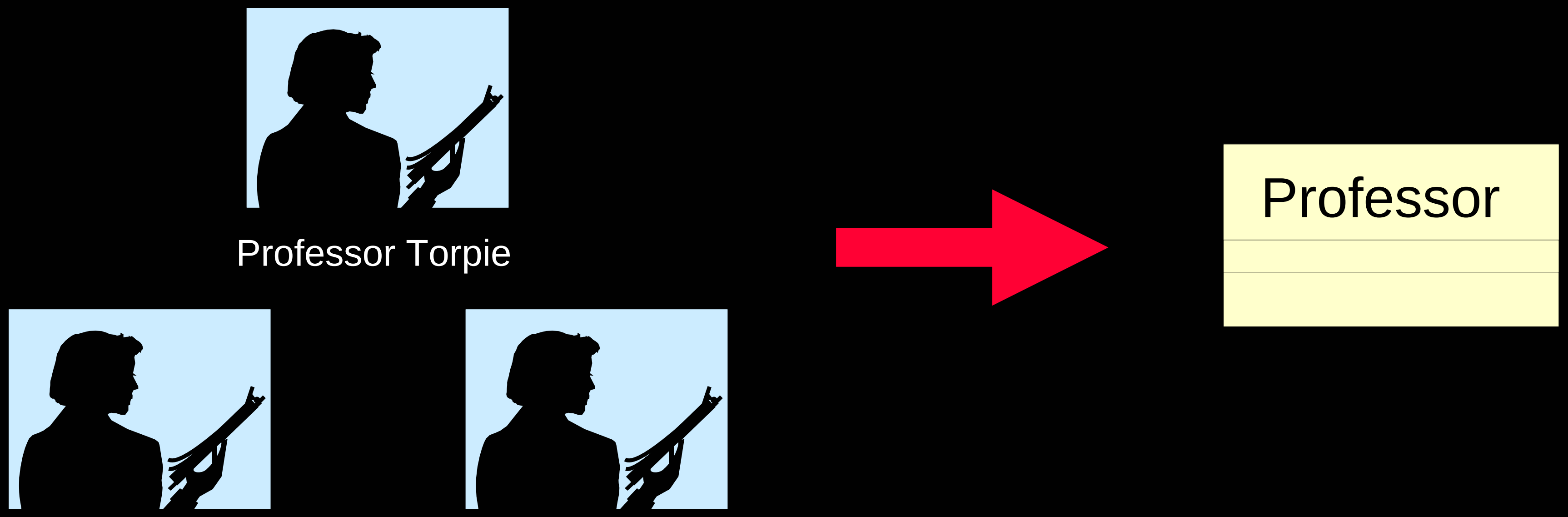
Professor

- name
- employeeID : UniqueId
- hireDate
- status
- discipline
- maxLoad

+ submitFinalGrade()
+ acceptCourseOffering()
+ setMaxLoad()
+ takeSabbatical()

The Relationship between Classes and Objects

- A class is an abstract definition of an object.
  - It defines the structure and behavior of each object in the class.
  - It serves as a template for creating objects.
- Classes are not collections of objects.



Professor Torpie

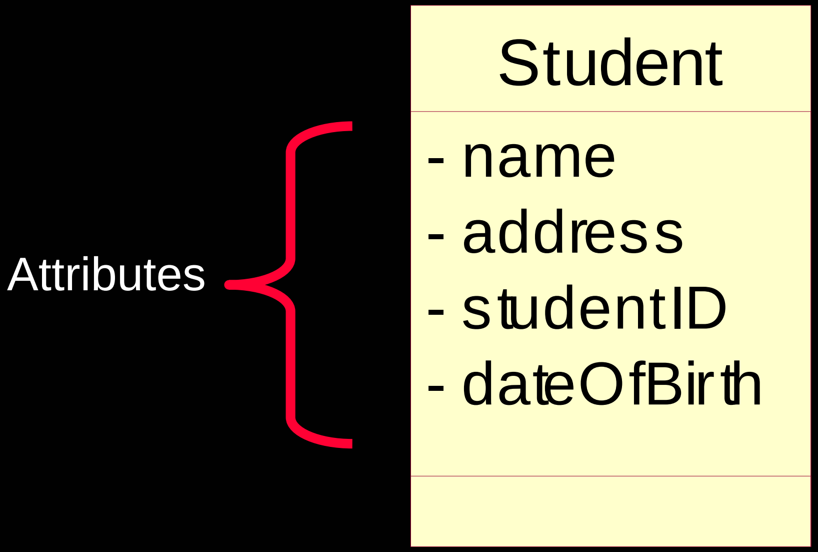
Professor Meijer

Professor Allen

Professor

What Is an Attribute?

- An attribute is a named property of a class that describes the range of values that instances of the property may hold.
  - A class may have any number of attributes or no attributes at all.

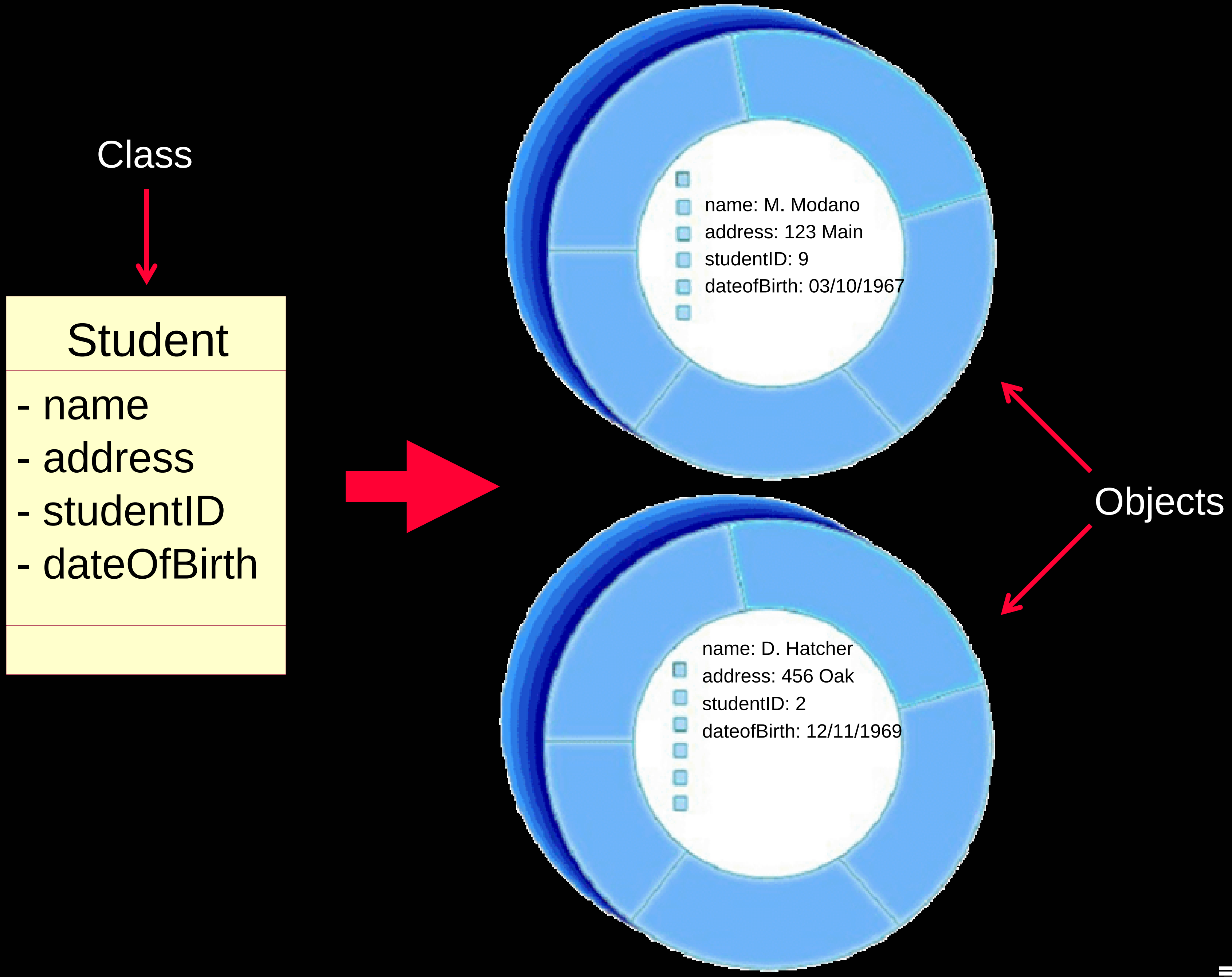


Attributes

Student

- name
- address
- studentID
- dateOfBirth

Attributes in Classes and Objects



Class

Student

- name
- address
- studentID
- dateOfBirth

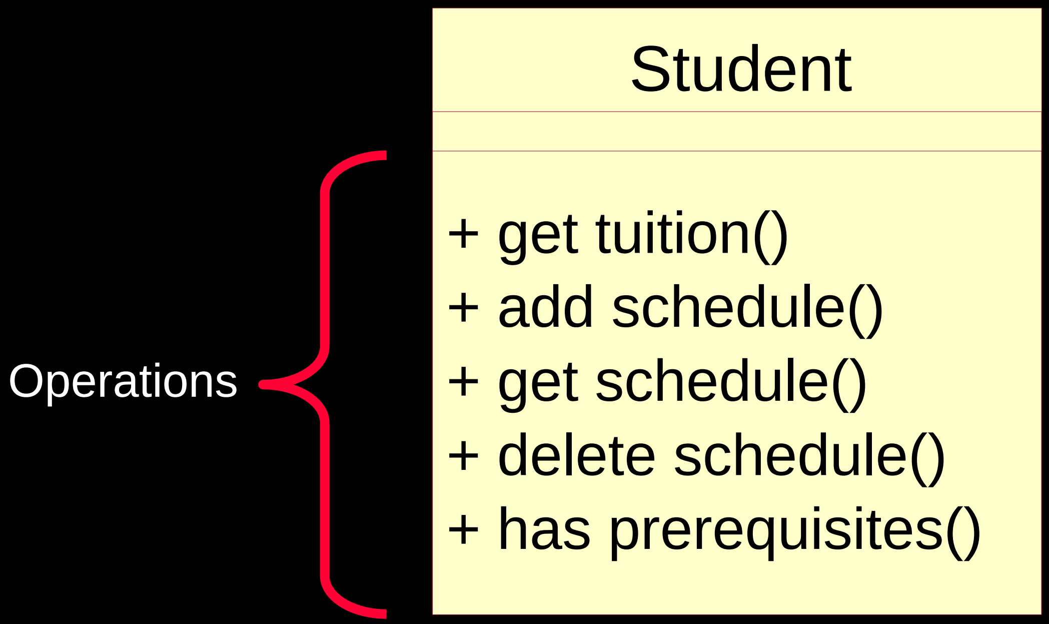
name: M. Modano
address: 123 Main
studentID: 9
dateofBirth: 03/10/1967

name: D. Hatcher
address: 456 Oak
studentID: 2
dateofBirth: 12/11/1969

Objects

What Is an Operation?

- A service that can be requested from an object to effect behavior. An operation has a signature, which may restrict the actual parameters that are possible.
- A class may have any number of operations or none at all.



Student

+ get tuition()
+ add schedule()
+ get schedule()
+ delete schedule()
+ has prerequisites()

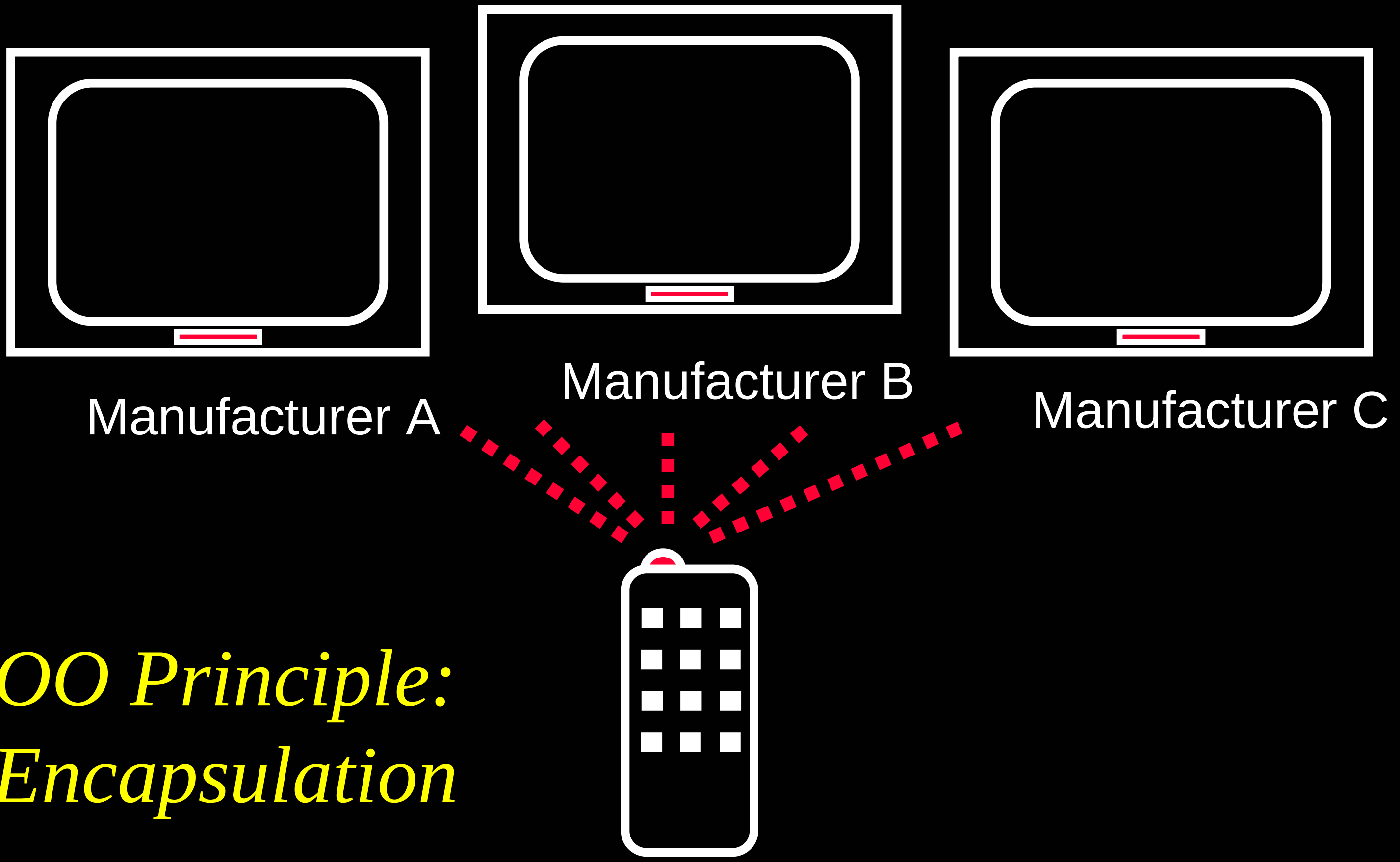
Operations

Where Are We?

- What is an object?
- Four principles of OO
- What is a class?
- ★ Polymorphism and generalization
- Organizing model elements

What Is Polymorphism?

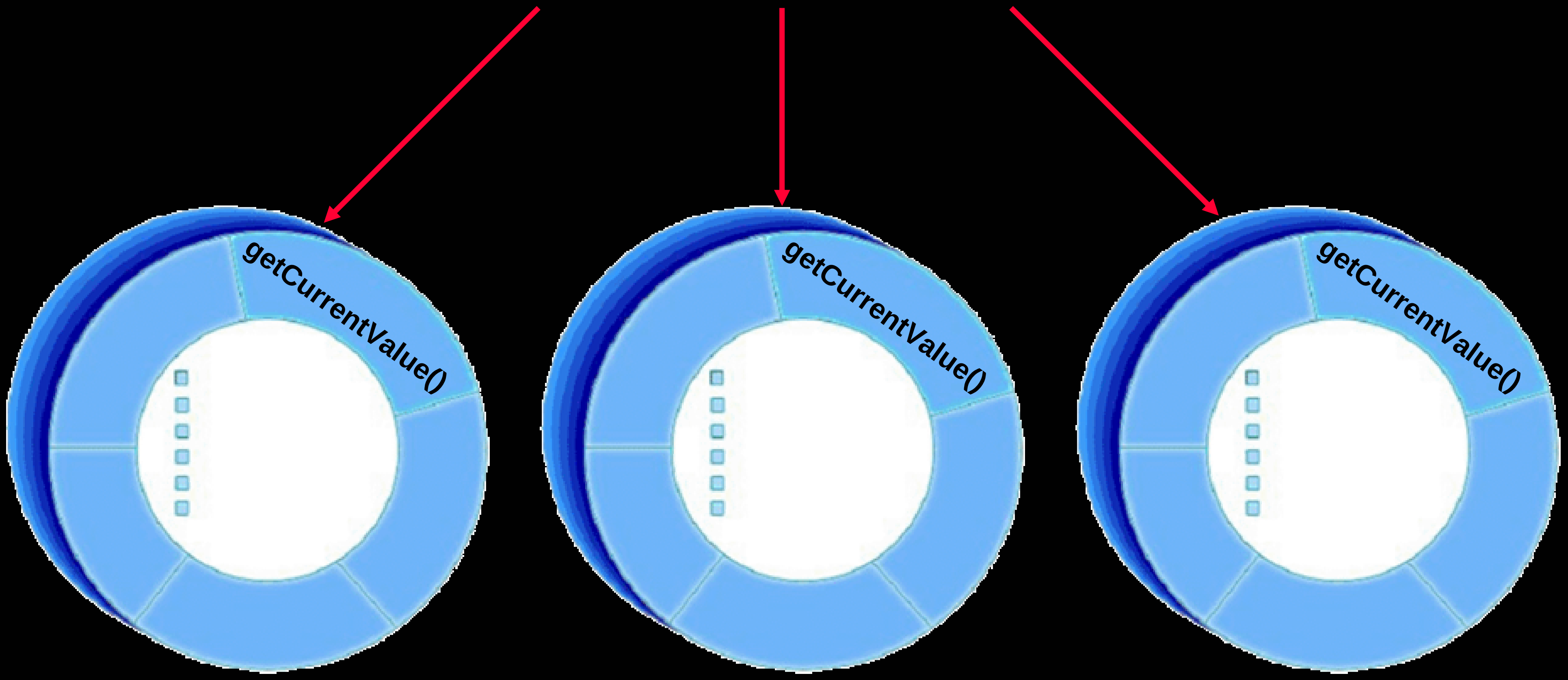
- The ability to hide many different implementations behind a single interface.



OO Principle:
Encapsulation

Example: Polymorphism

financialInstrument.getCurrentValue()



Stock

Bond

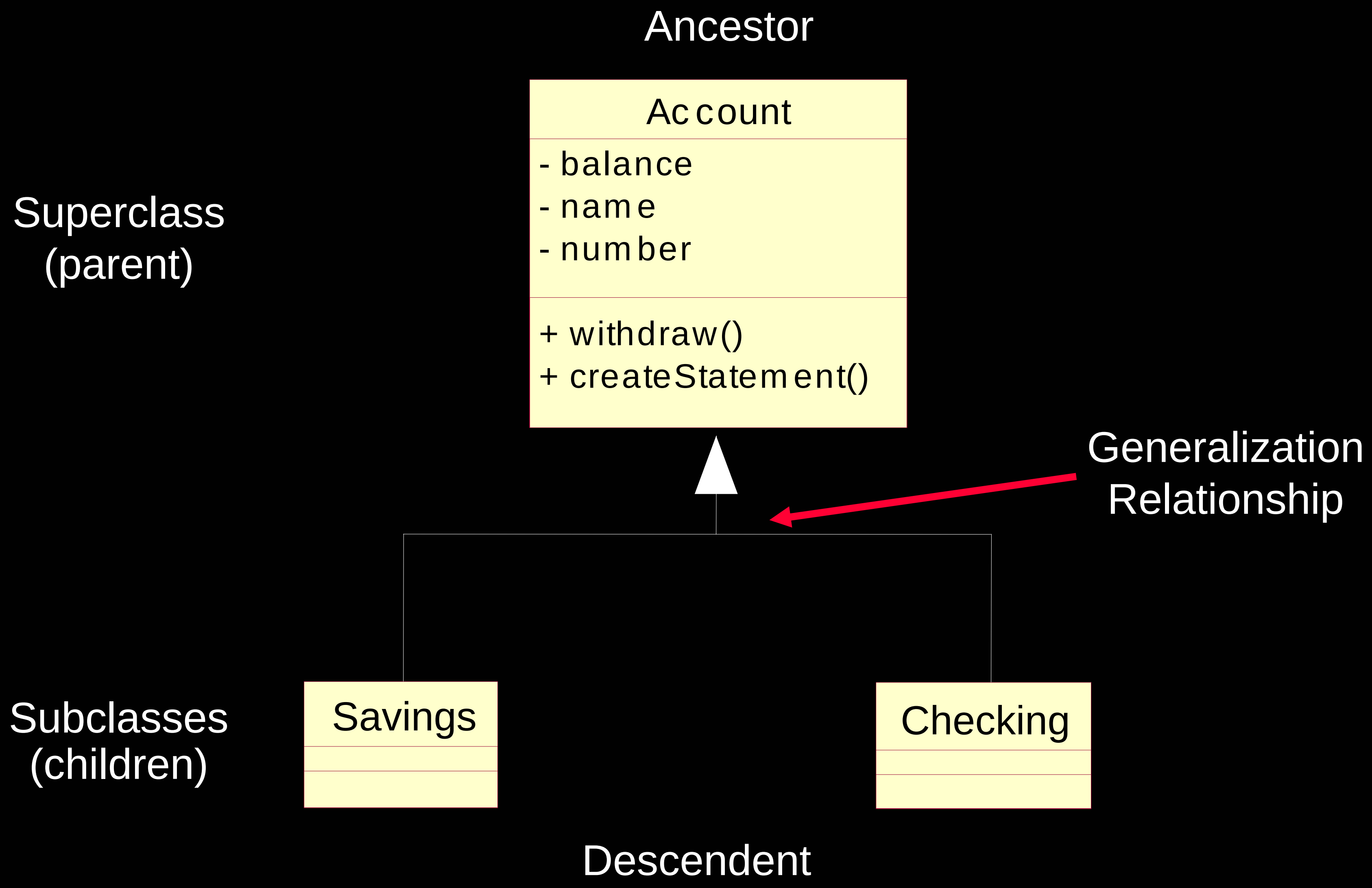
Mutual Fund

What Is Generalization?

- A relationship among classes where one class shares the structure and/or behavior of one or more classes.
- Defines a hierarchy of abstractions in which a subclass inherits from one or more superclasses.
  - Single inheritance.
  - Multiple inheritance.
- Is an “is a kind of” relationship.

Example: Single Inheritance

- One class inherits from another.



Ancestor

Account

- balance
- name
- number

+ withdraw()
+ createStatement()

Superclass
(parent)

Generalization
Relationship

Subclasses
(children)

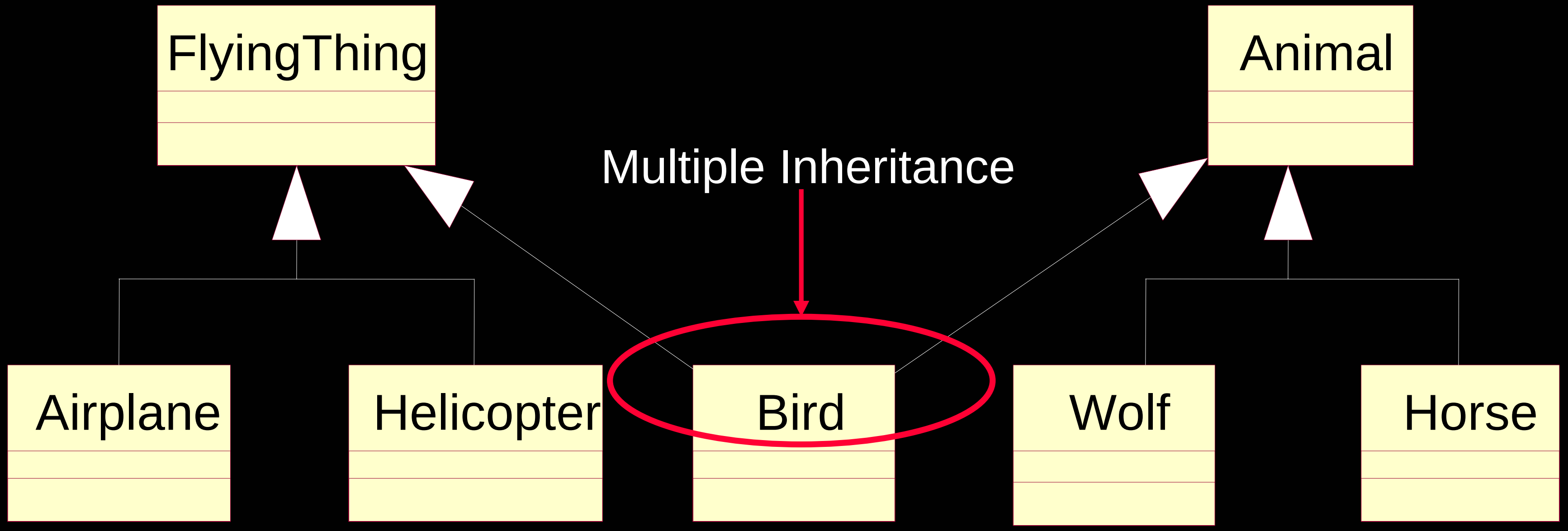
Savings

Checking

Descendent

Example: Multiple Inheritance

- A class can inherit from several other classes.



Use multiple inheritance only when needed and always with caution!

What Is Inherited?

- A subclass inherits its parent’s attributes, operations, and relationships.
- A subclass may:
  - Add additional attributes, operations, relationships.
  - Redefine inherited operations. (Use caution!)
- Common attributes, operations, and/or relationships are shown at the highest applicable level in the hierarchy.

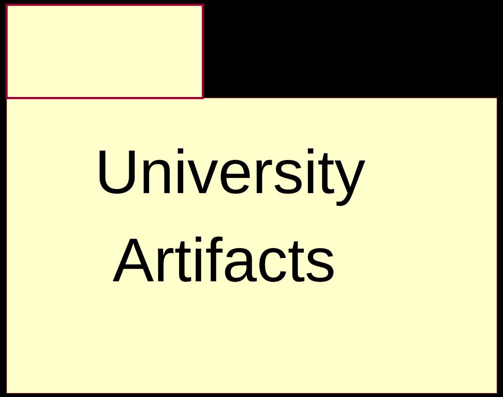
Inheritance leverages the similarities among classes.

Where Are We?

- What is an object?
- Four principles of OO
- What is a class?
- Polymorphism and generalization
- ★ Organizing model elements

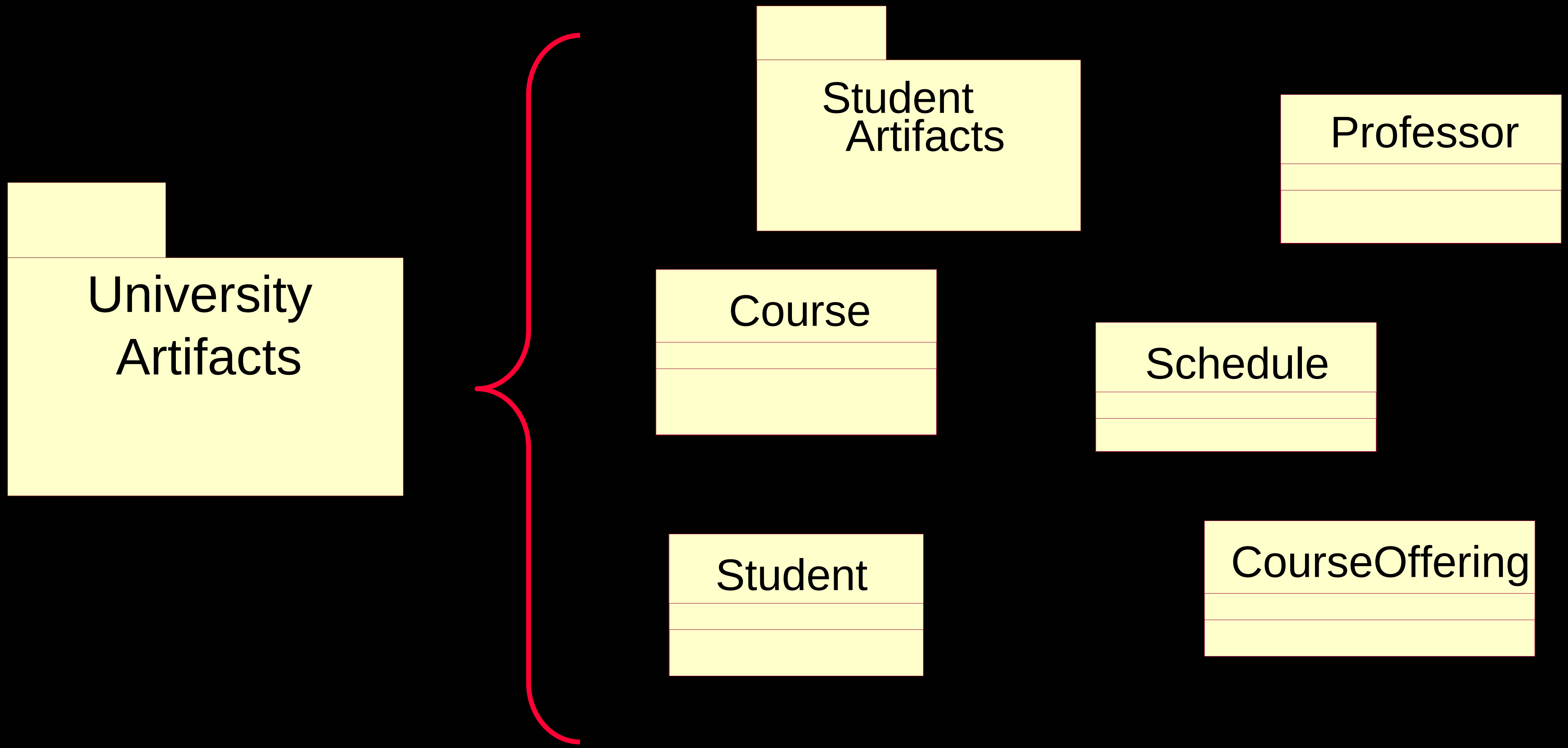
What Is a Package?

- A general purpose mechanism for organizing elements into groups.
- A model element that can contain other model elements.
- A package can be used:
  - To organize the model under development.
  - As a unit of configuration management.



A Package Can Contain Classes

- The package, University Artifacts, contains one package and five classes.



Review

- What is an object?
- What are the four principles of object orientation? Describe each.
- What is a class? How are classes and objects related?
- What is an attribute? An operation?
- Define polymorphism. Provide an example of polymorphism.
- What is generalization?
- Why use packages?

Exercise: Principles of Object Orientation

- The “OO Quiz Show” Rules
  - Everyone in the class is assigned a number.
  - The instructor displays a question.
  - The instructor calls out a number.
  - If the student answers the question correctly, the class continues to the next question.
  - If the student does not answer the question correctly, the class goes back to the beginning.
  - The game is over when all questions have been answered correctly.