Object Oriented Development with Java

(CT038-3-2 and Version VC1)

Concept of Object Orientation

Object Oriented Modeling

Topic & Structure of the lesson

- **Overview of**
  - **-Objects**
  - **-Class**
  - **-Principles of Object Orientation**
  - **-Package**

Learning outcome

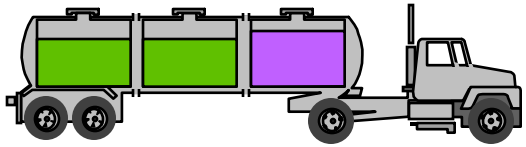
- At the end of this lesson, you will be able to
  - Describe abstraction, encapsulation, modularity, and hierarchy
  - Describe the physical structure of a class
  - Describe the relationship between a class and an object
  - Define polymorphism and generalization

Key Terms you must be able to usse

- **Overview of**
  - **Object**
  - **Class**
  - **Abstraction**
  - **Encapsulation**
  - **Inheritance**
  - **Polymorphism**

What Is an Object?

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

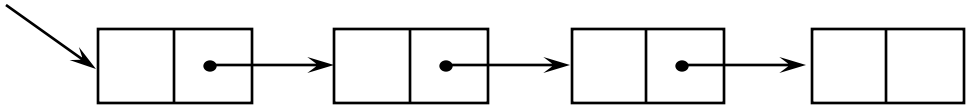


Truck

  - Conceptual entity

Chemical Process

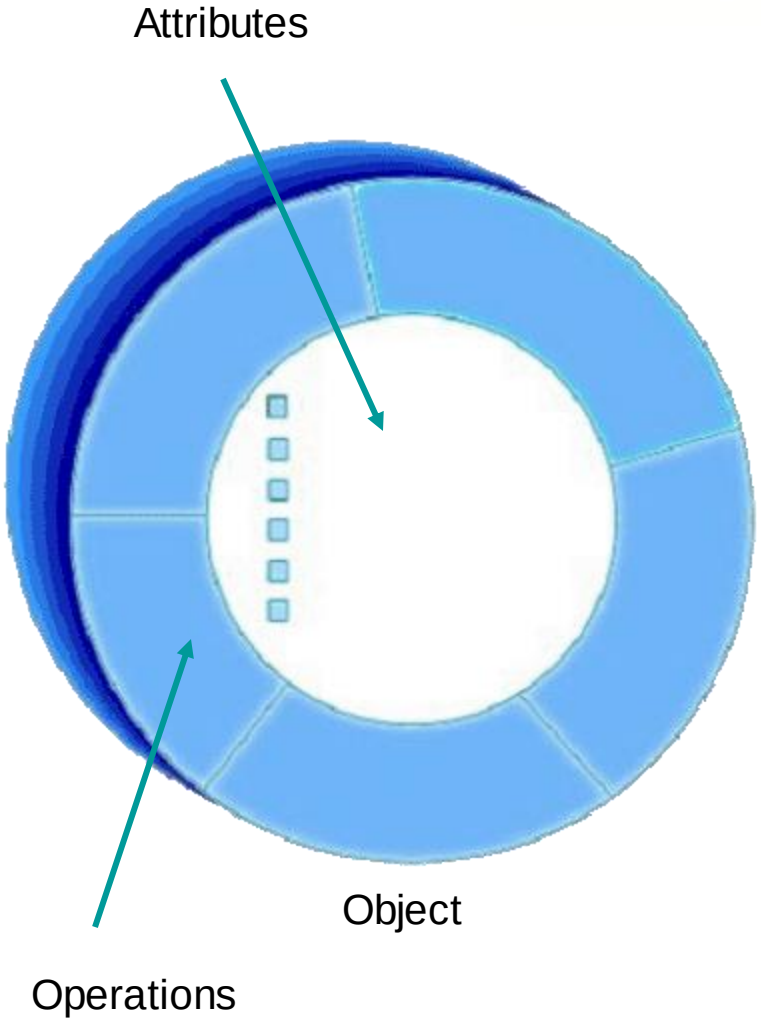
  - Software entity



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.

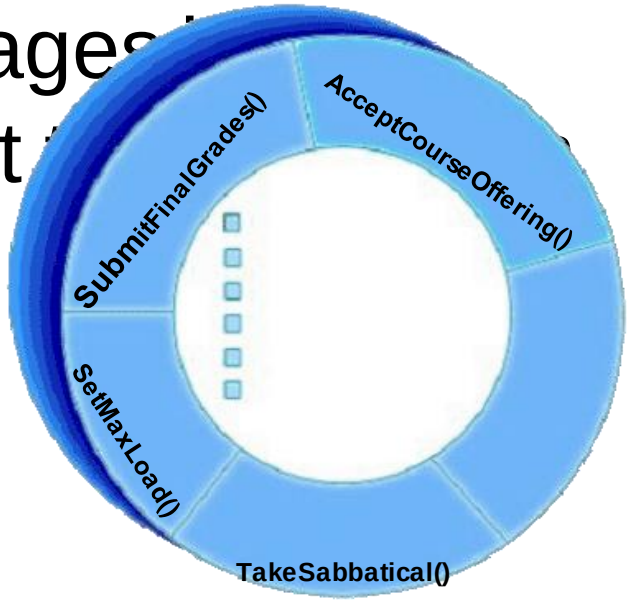
Name: J Clark
Employee ID: 567138
Date Hired: July 25, 1991
Status: Tenured
Discipline: Finance
Maximum Course Load: 3 classes

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 [illegible] respond to (operations that [illegible] perform).

Professor Clark's behavior
- Submit Final Grades
- Accept Course Offering
- Take Sabbatical
- Set Max Load



Professor Clark

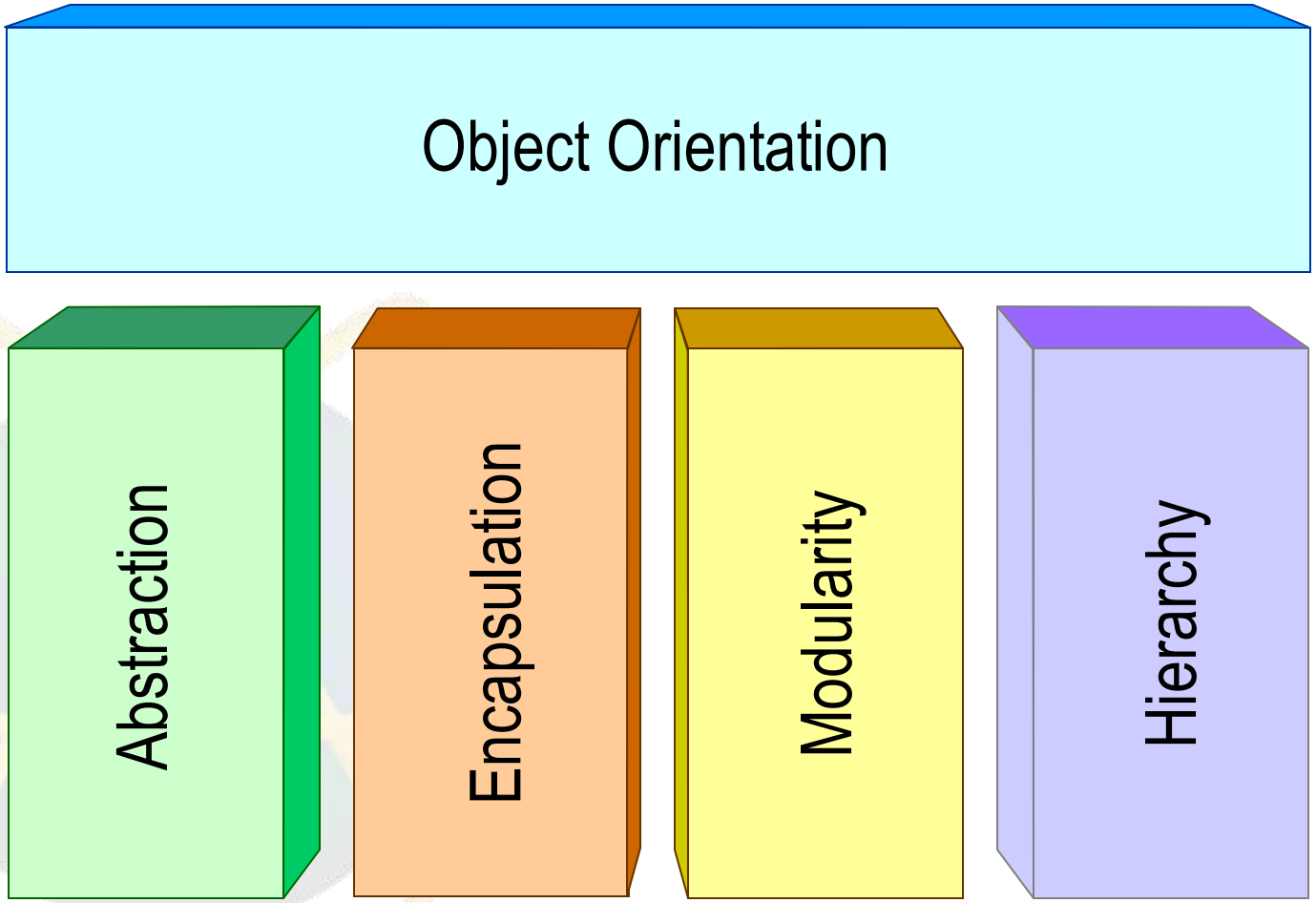
An Object Has Identity

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

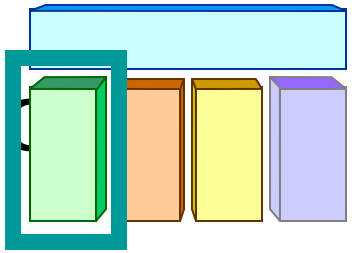
Professor “J Clark” teaches Biology

Professor “J Clark” teaches Biology

Basic Principles of Object Orientation



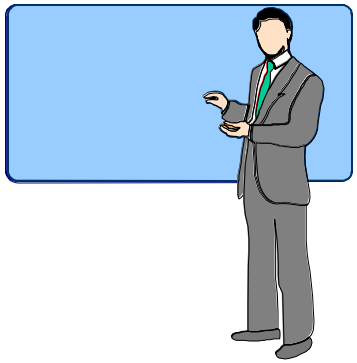
What Is Abstracti



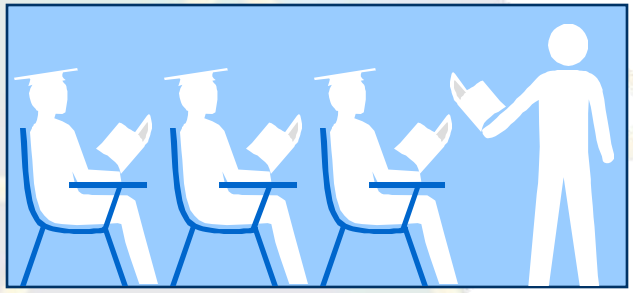
- The essential characteristics of an entity that distinguishes it from all other kinds of entities.
- Used to hide certain details and only show the essential features of the objects
- Defines a boundary relative to the perspective of the viewer.
- Is not a concrete manifestation, denotes the ideal essence of something.

Example: Abstraction

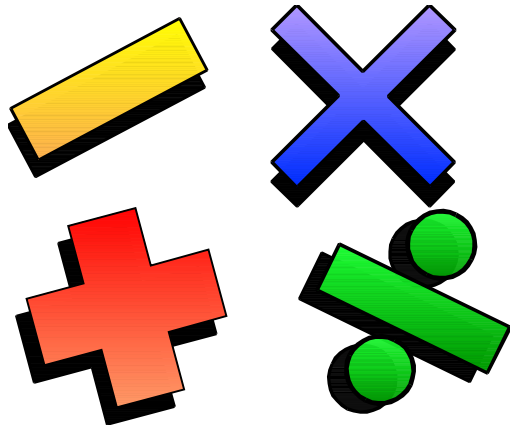
Student



Professor

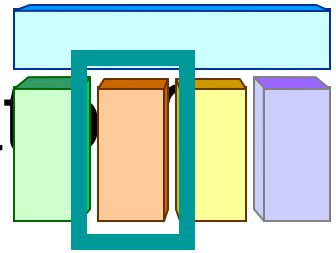


Course Offering (9:00 a.m., Monday-Wednesday-Friday)

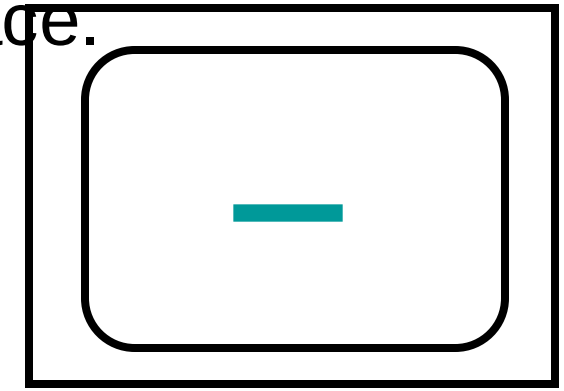


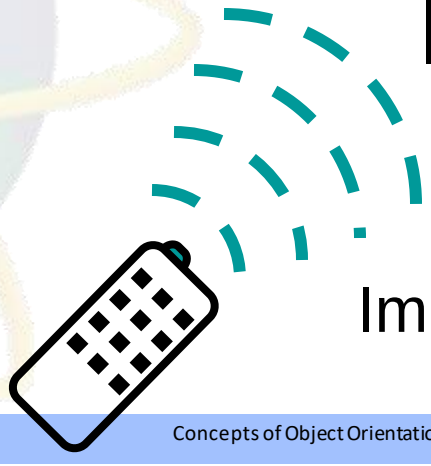
Course (e.g. Algebra)

What Is Encapsulat



- A mechanism of wrapping the data(variables) and code acting on the data(methods) together as a single unit
  - Clients depend on interface.

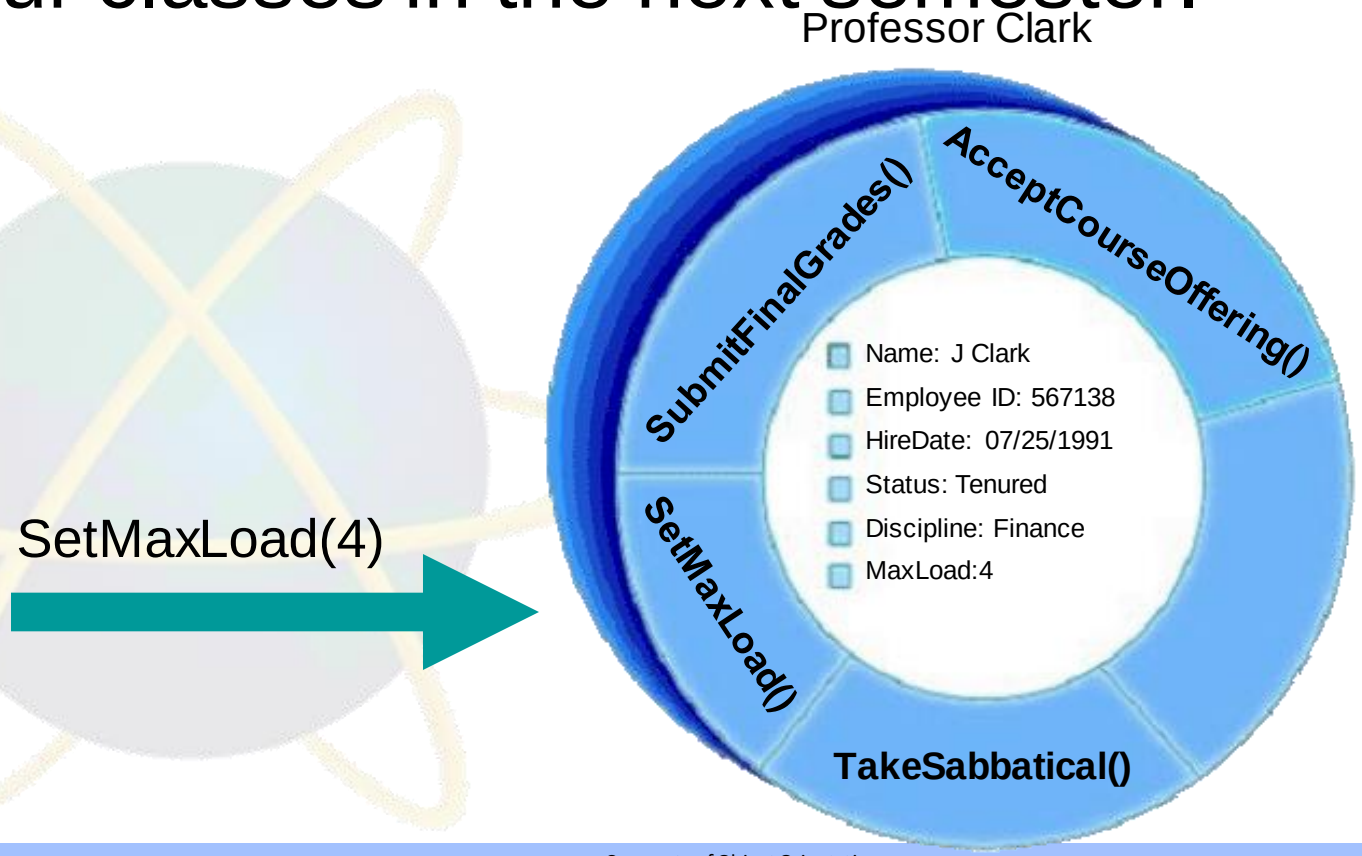




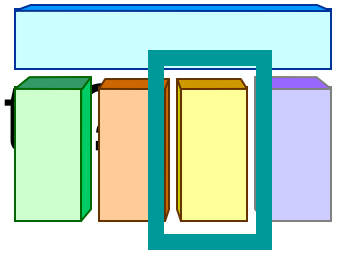
Improves Resiliency

Encapsulation Illustrated

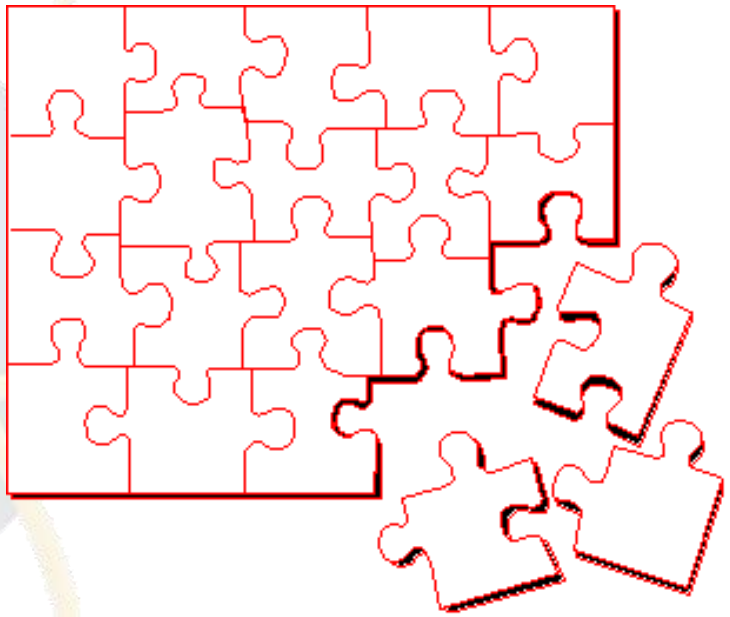
- Professor Clark needs to be able to teach four classes in the next semester.



What Is Modularity?

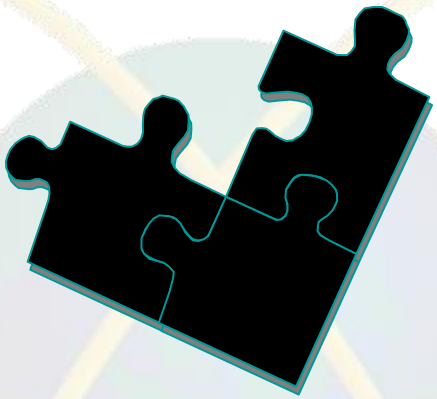


- Breaks up something complex into manageable pieces.
- Helps people understand complex systems.



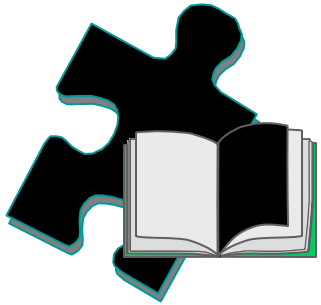
Example: Modularity

- For example, break complex systems into smaller modules.

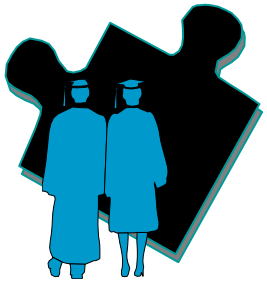


Course Registration System

Billing System

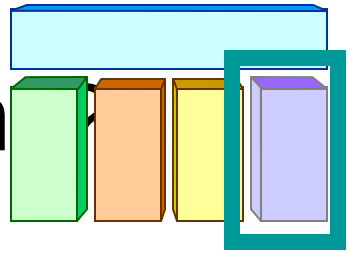


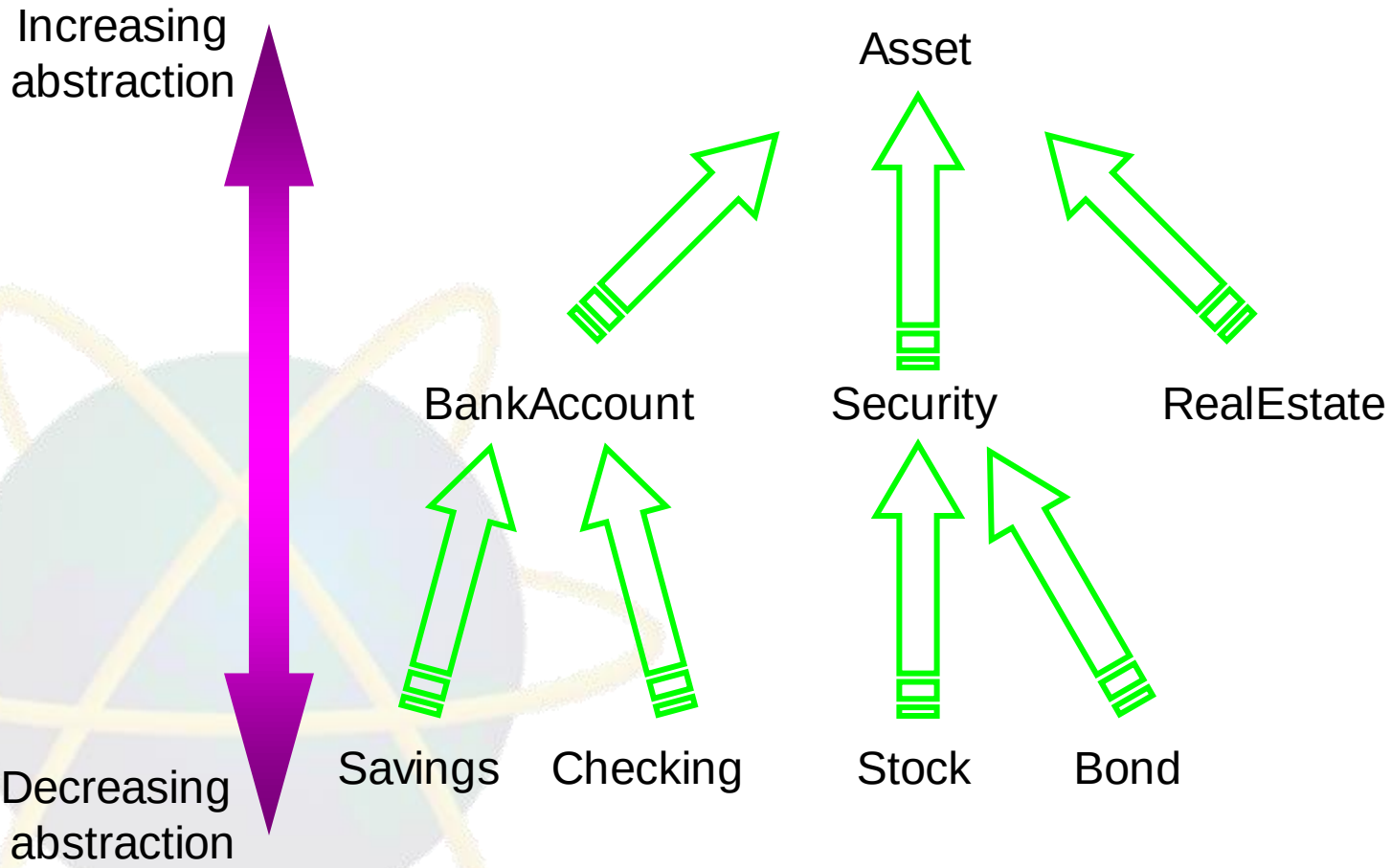
Course Catalog System



Student Management System

What Is Hierarch





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

Professor

Named Object

: Professor

Anonymous Object

What Is a Class?

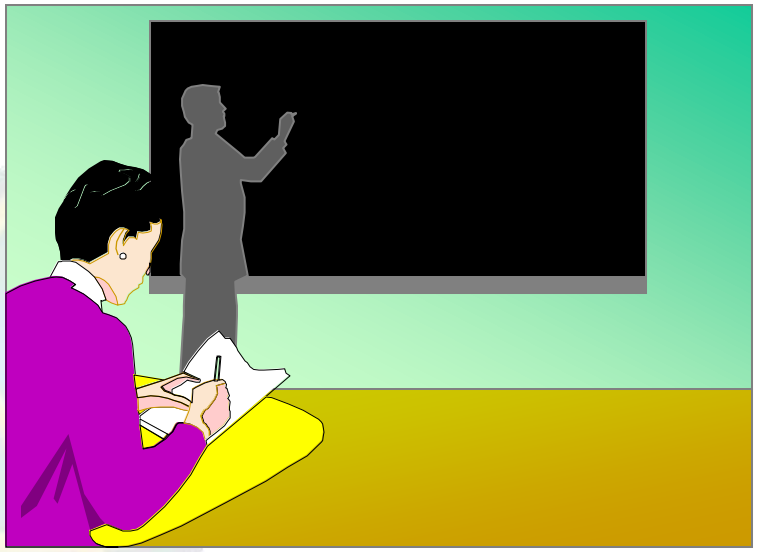
- A class is a description of a set of objects that share the same *attributes*, *operations*, *relationships*, and *semantics*.
  - An object is an instance of a class.
- A class is an abstraction in that it
  - Emphasizes relevant characteristics.
  - Suppresses other characteristics.

A Sample Class

A·P·U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

Class
Course

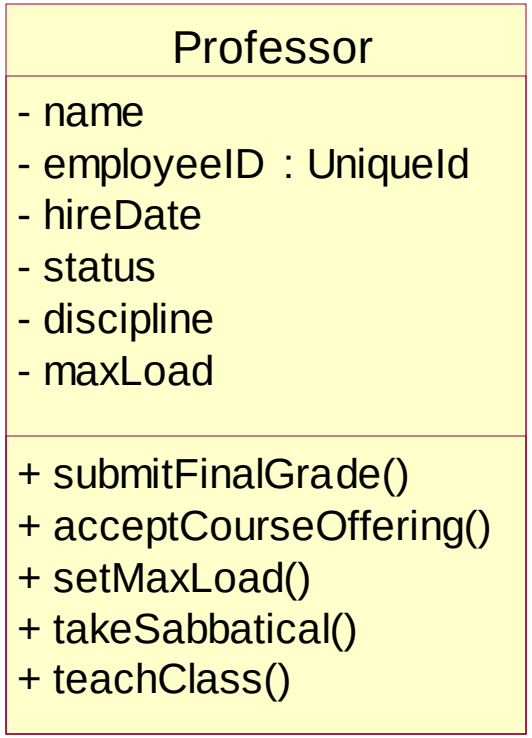
Properties
Name
Location
Days offered
Credit hours
Start time
End time



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

Representing Classes in UML

- A class is represented using a rectangle with three compartments:
  - The class name
  - The structure (attributes)
  - The behavior (operations)



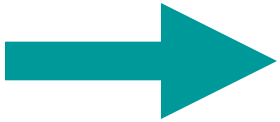
The Relationship between Classes and Objects

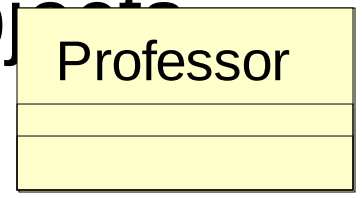
- A class is an abstract definition of an object.
  - It defines the structure and behavior of each object in the class.
  - It serv template for creating objects.
- Classe ot collections of obj

Professor Torpie

Professor Meijer

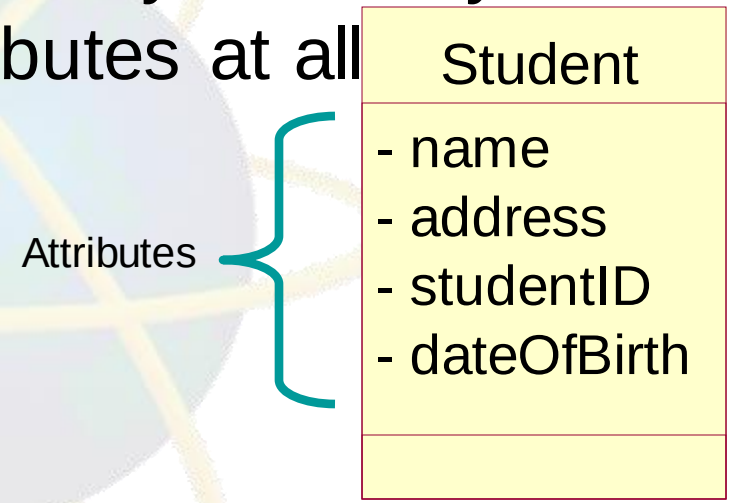
Professor Allen



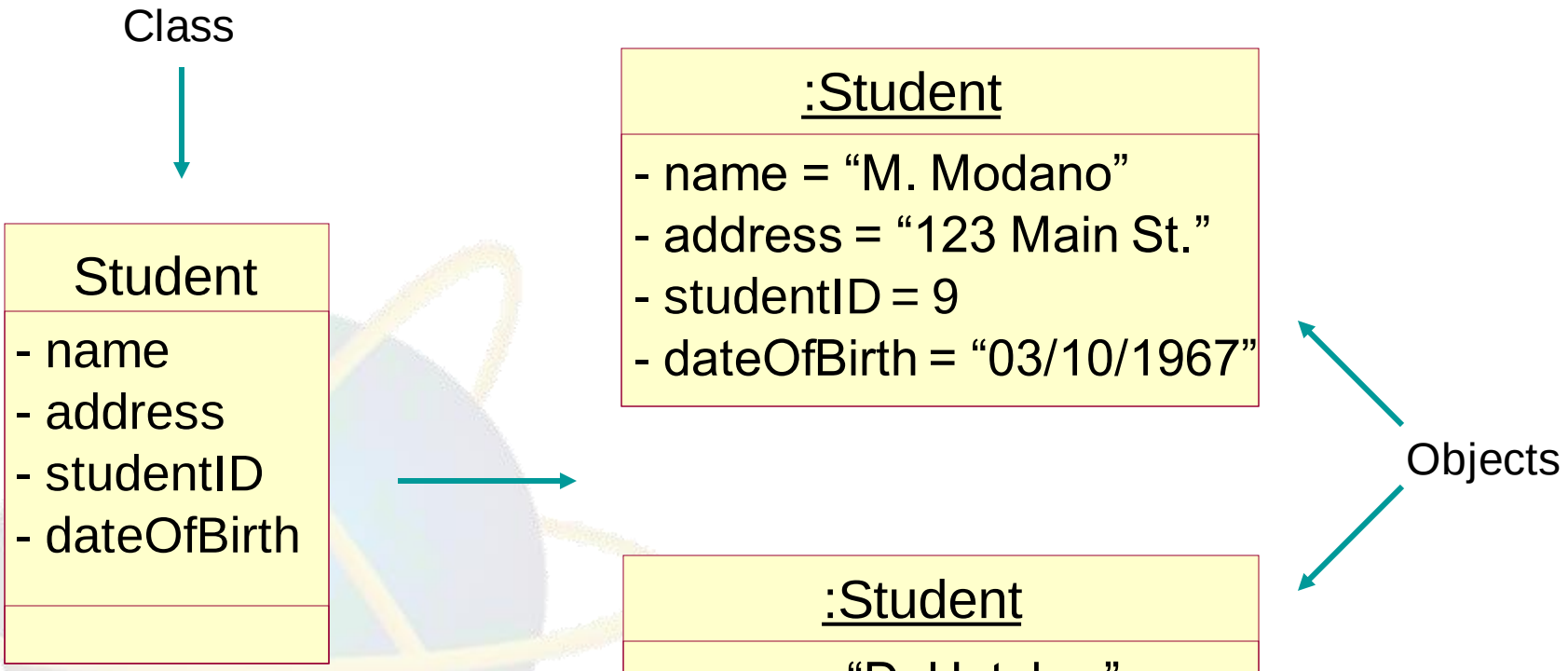


What Is an Attribute?

- 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

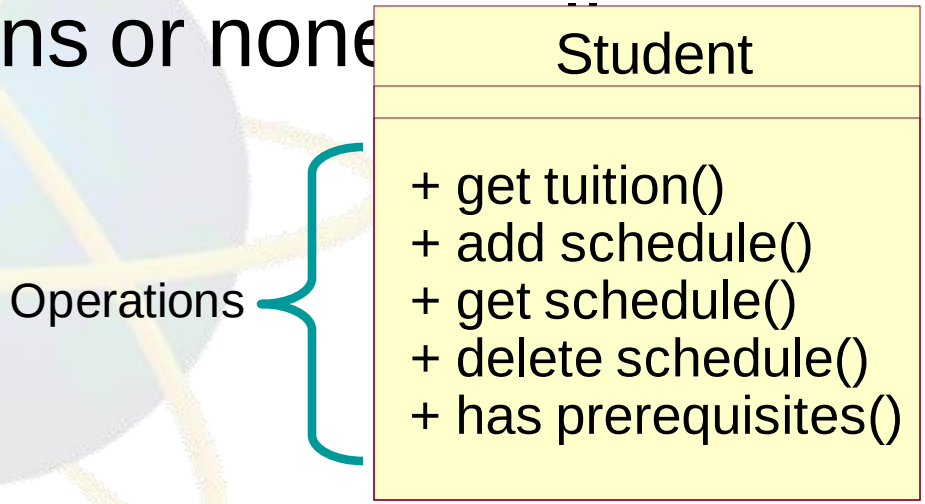


Attributes in Classes and Objects



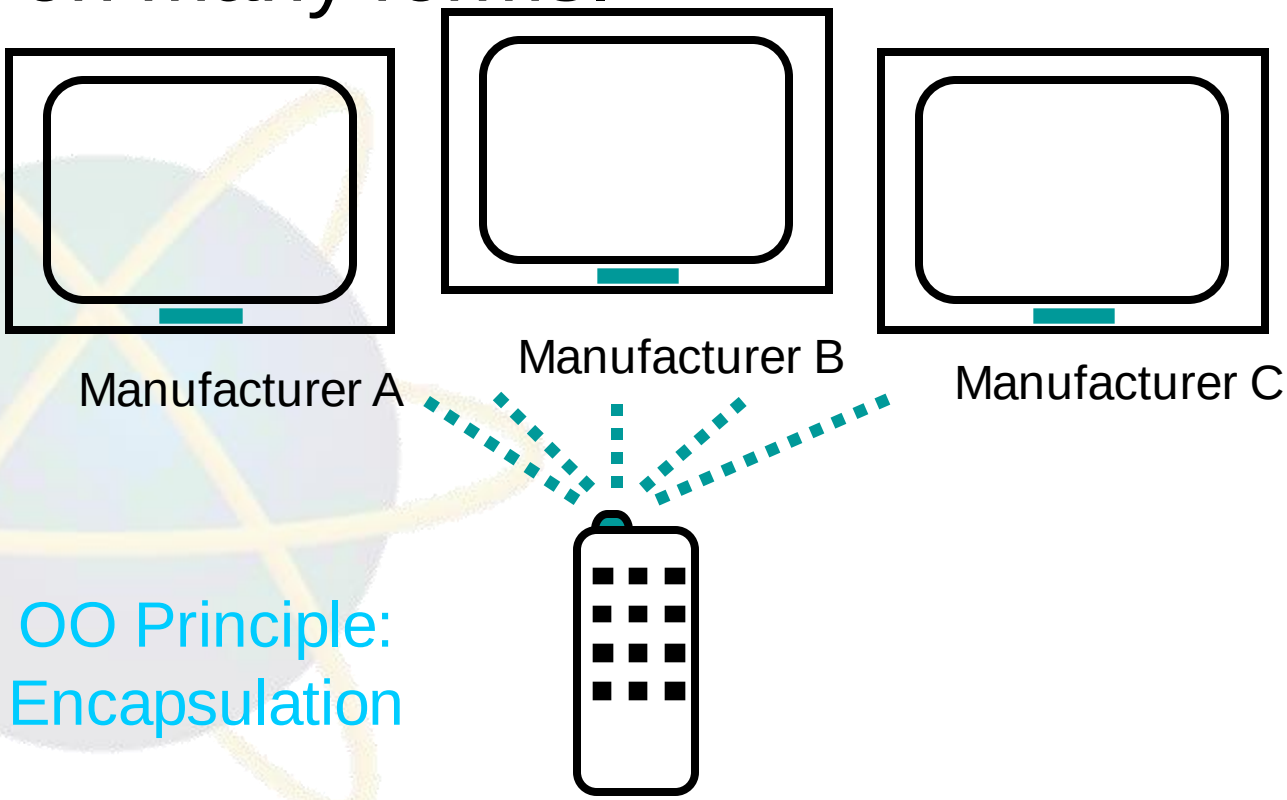
What Is an Operation?

- 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



What Is Polymorphism?

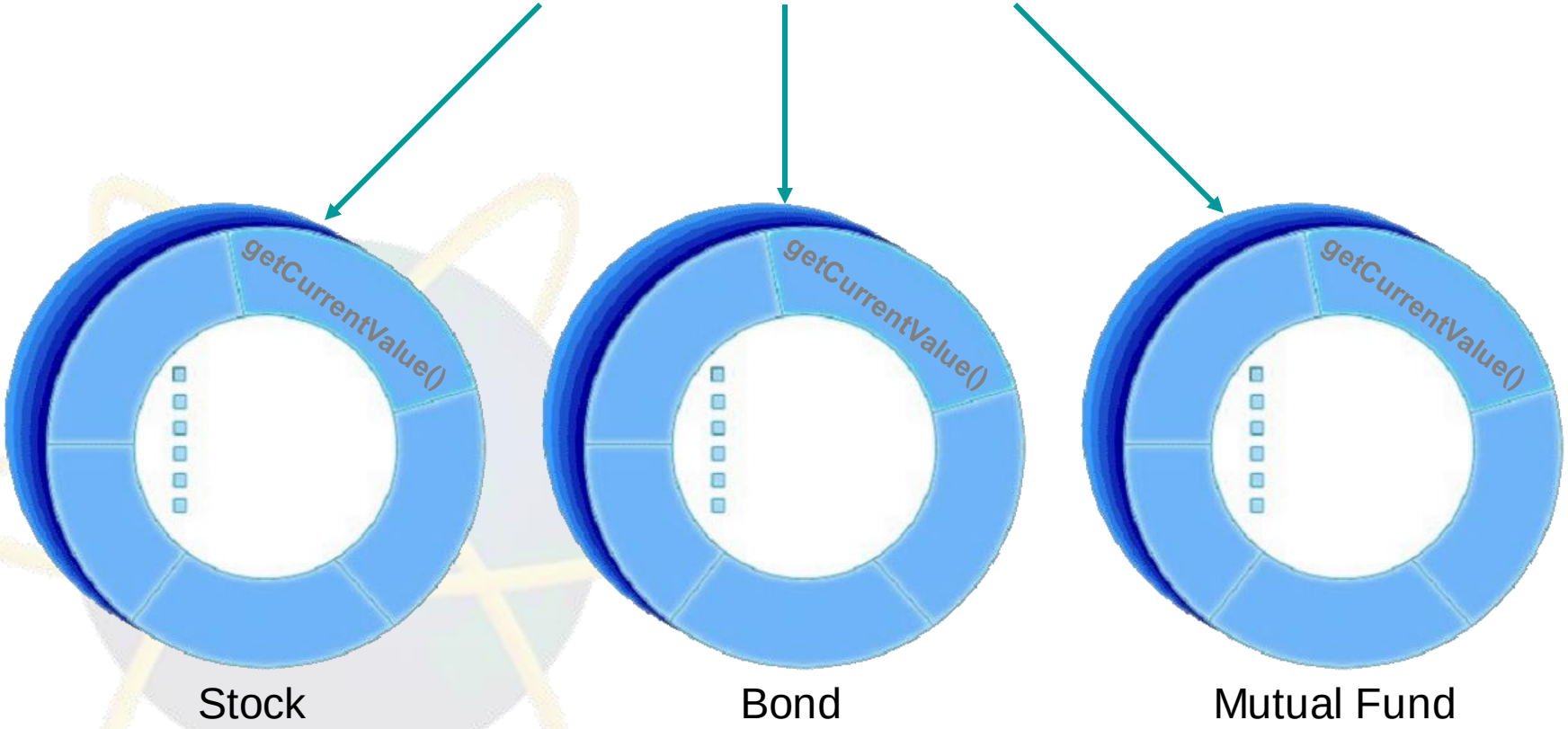
- Polymorphism is the ability of an object to take on many forms.



OO Principle:
Encapsulation

Example: Polymorphism

financialInstrument.getCurrentValue()

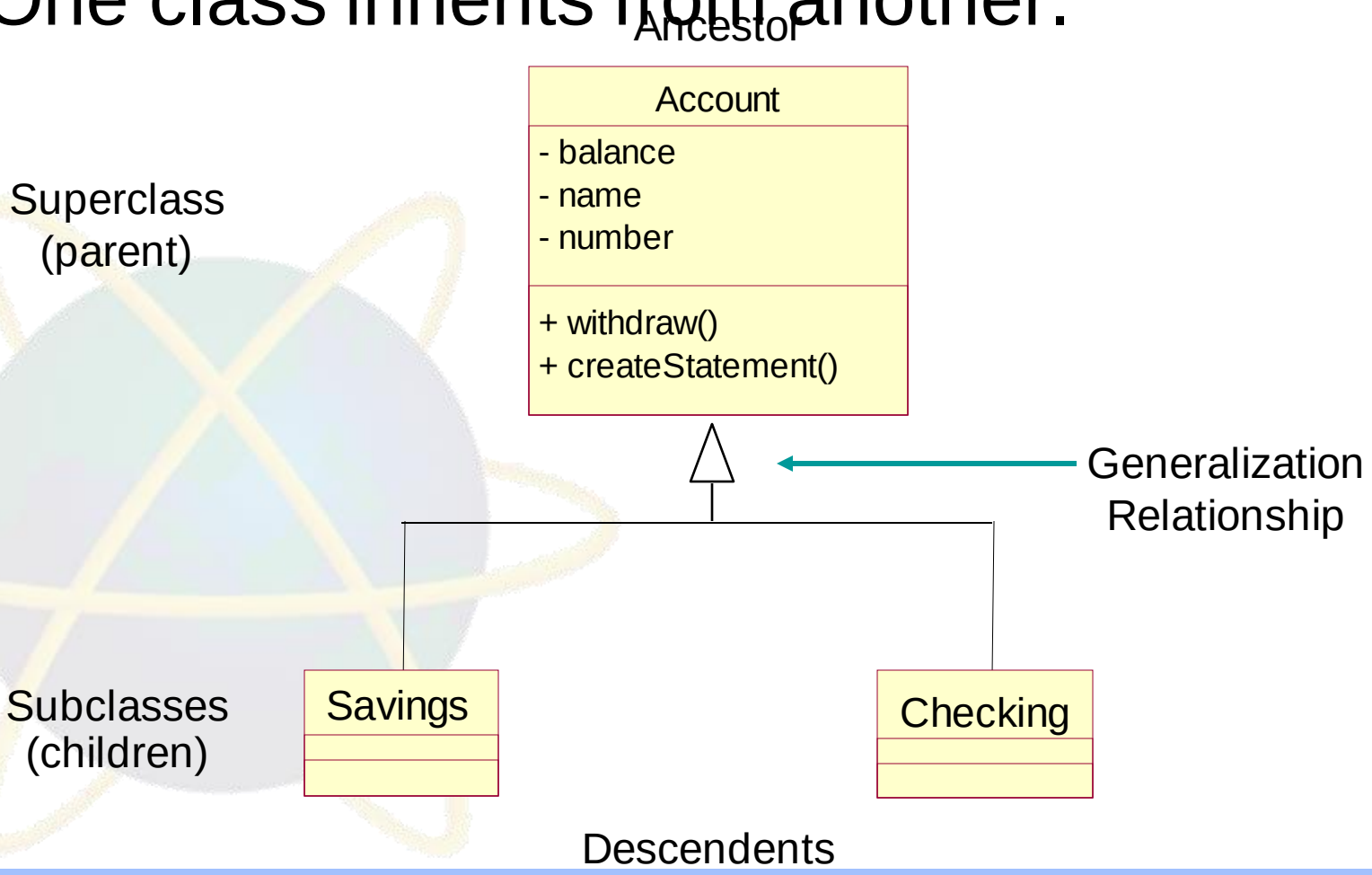


What Is Generalization?

- 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.

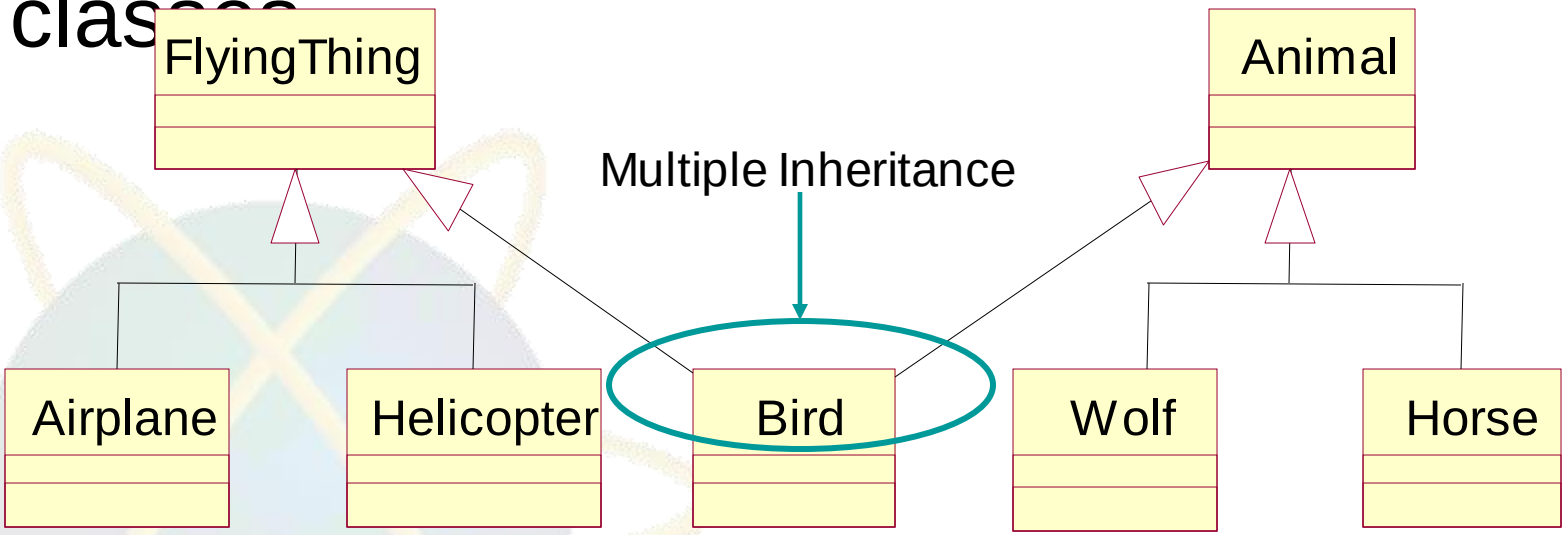
Example: Single Inheritance

- One class inherits from another.



Example: Multiple Inheritance

- A class can inherit from several other classes



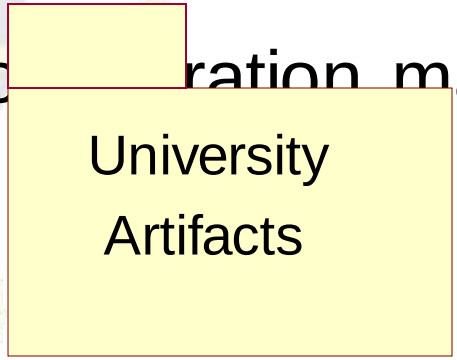
Use multiple inheritance only when needed and always with caution!

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.

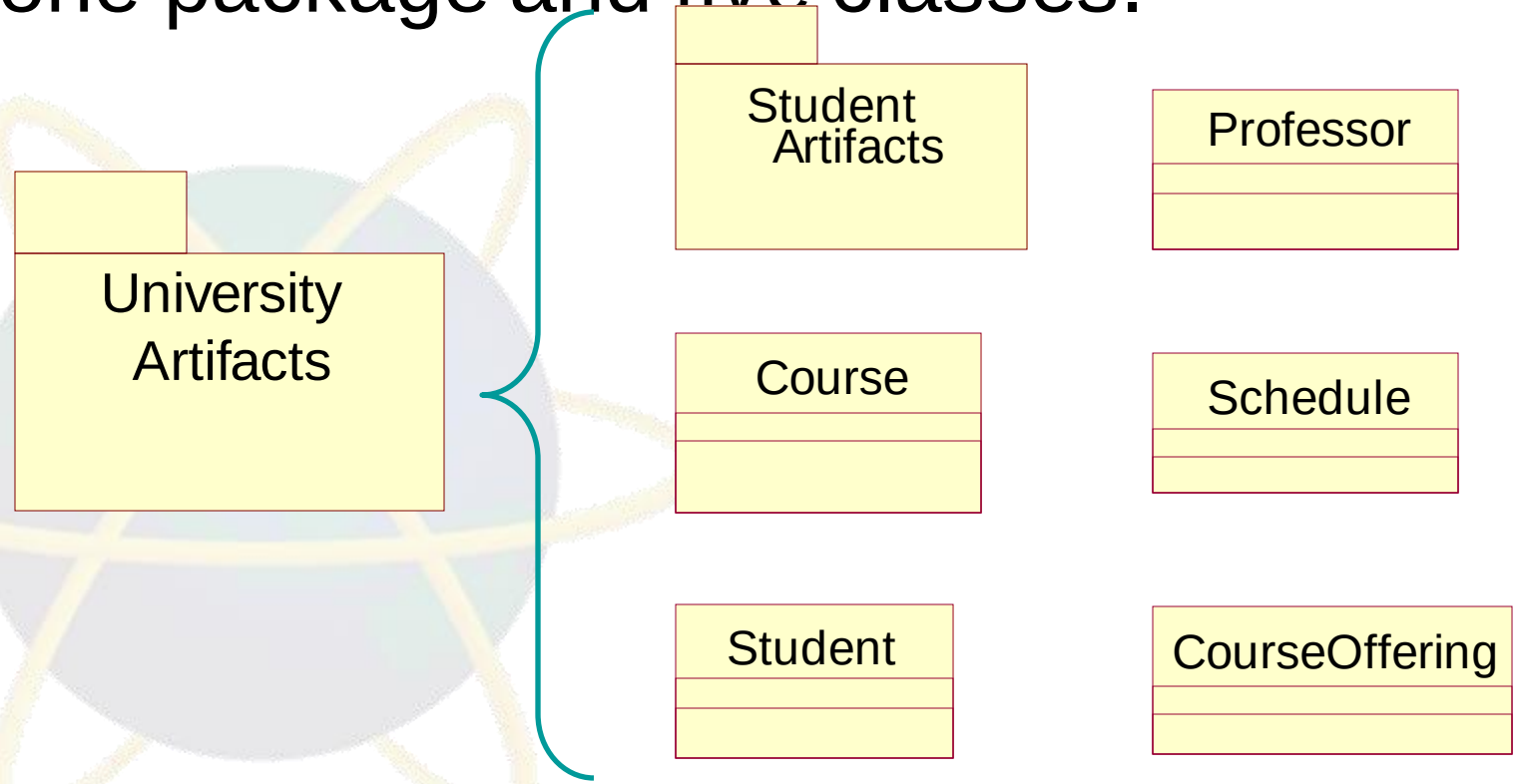
What Is a Package?

- 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.



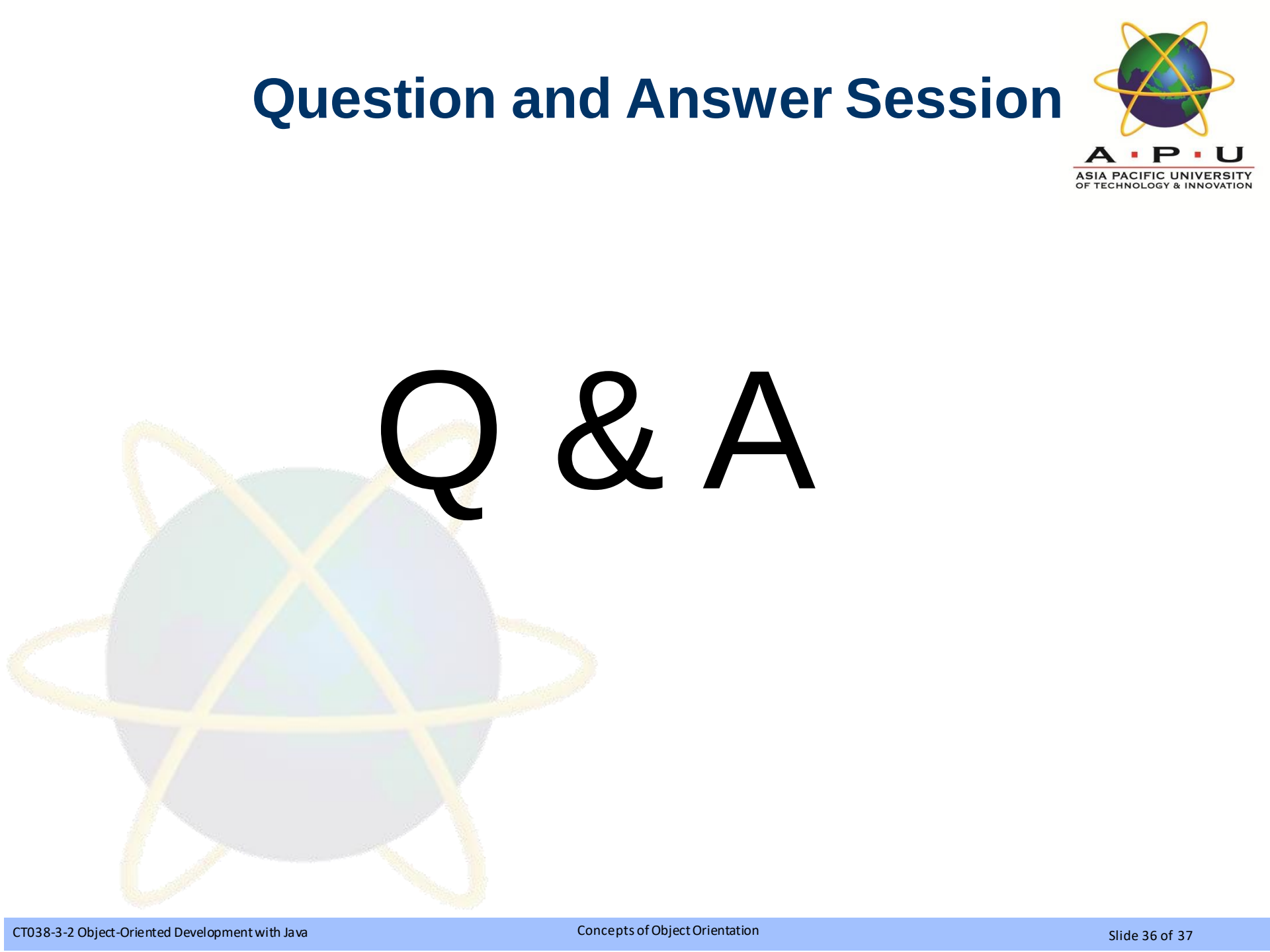
Quick Review Questions

- 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?

Summary of Main Teaching Points

- Object
- Class
- Abstraction
- Encapsulation
- Inheritance
- Polymorphism

Question and Answer Session



Q & A

Next Session

- Objects
- Methods
- Accessors and Mutators
- Constructors
- Static members
- Visibility modifiers