

LECTURE-2

Basic concepts of OO

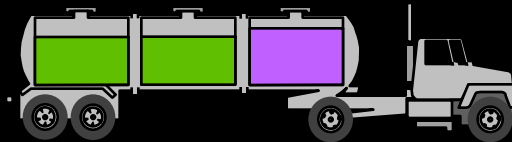
Basic Concepts of Object Orientation

- **Object**
- **Class**
- **Message**
- **Basic Principles of Object Orientation**
- **Abstraction**
- **Encapsulation**
- **Inheritance**
- **Polymorphism**
- **Interface and Abstract Class**

What Is an Object?

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

– Physical entity



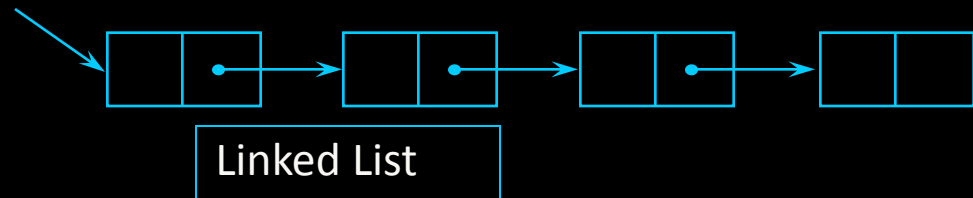
Truck

– Conceptual entity



Chemical
Process

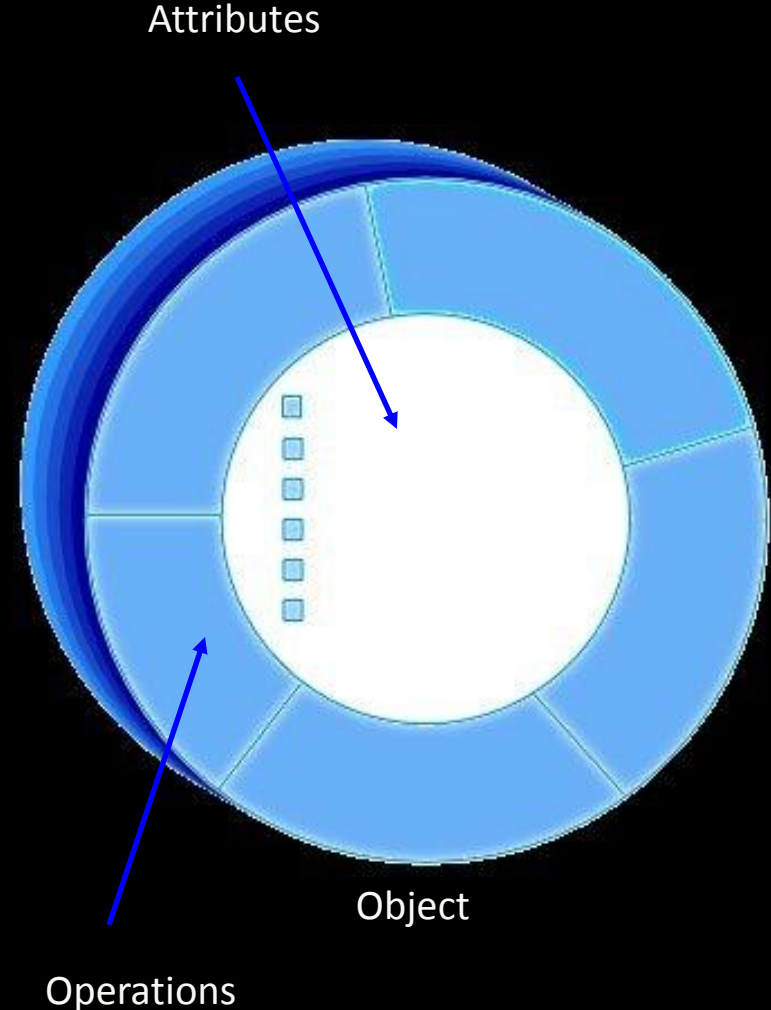
– Software entity



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

- The state of an object is one of the possible conditions in which an object may exist.
- The state of an object normally changes over



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 the set of messages it can respond to (operations the object can perform).



Professor Clark

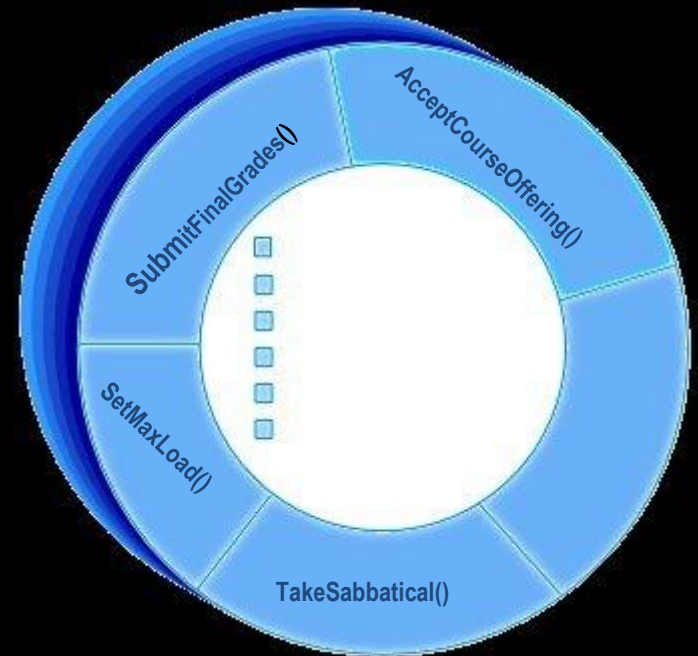
Professor Clark's behavior

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



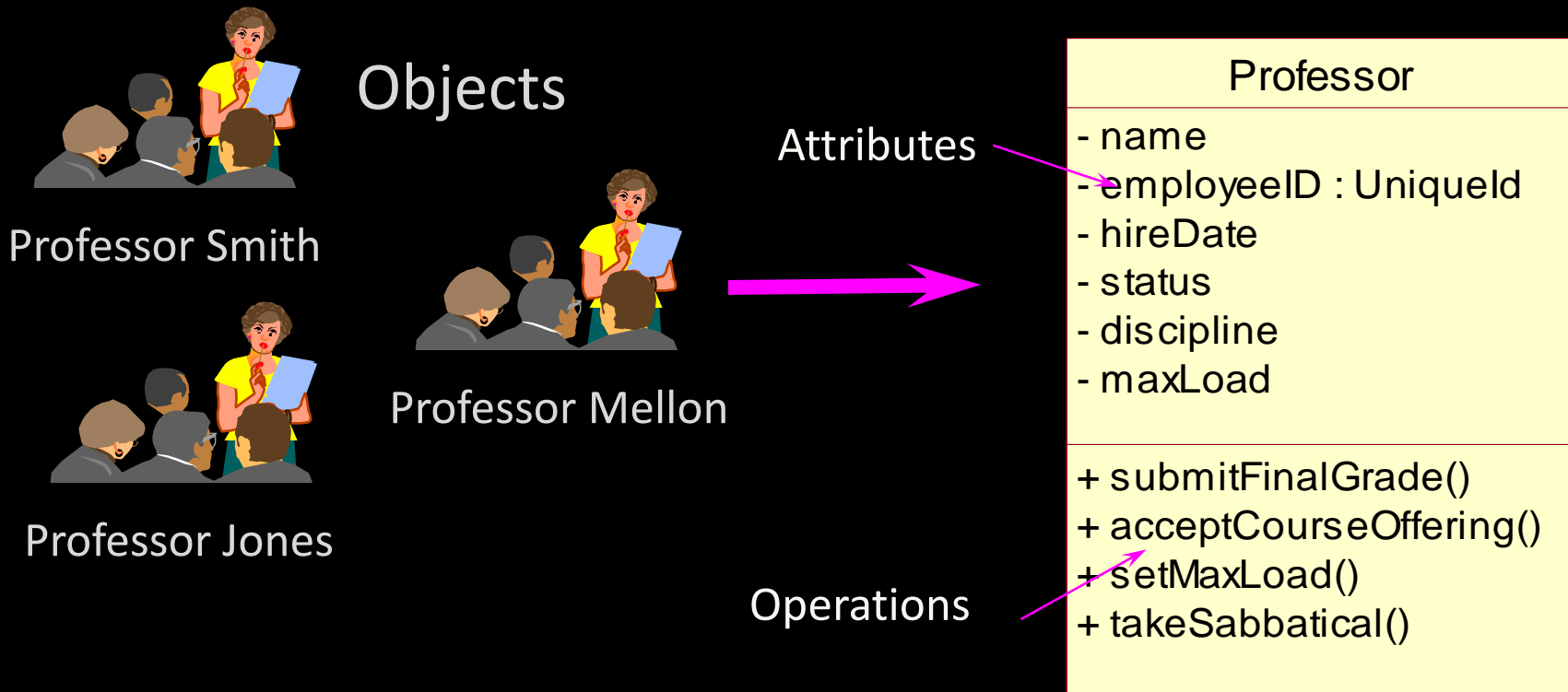
Professor "J Clark" teaches
Biology

Objects Need to Collaborate

- Objects are useless unless they can collaborate together to solve a problem.
 - Each object is responsible for its own behavior and status.
 - No one object can carry out every responsibility on its own.
- How do objects interact with each other?
 - They interact through messages.

What Is a Class?

- A class is a description of a set of objects that share the same properties and behavior.
 - An object is an instance of a class.



A Sample Class

Class: Automobile

Data Items:

- manufacturer's name
- model name
- year made
- color
- number of doors
- size of engine
- etc.

Methods:

- Define data items
(specify
manufacturer's name,
model, year, etc.)
- Change a data item
(color, engine, etc.)
- Display data items
- Calculate cost
- etc.

The Relationship Between Classes and Objects

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
- Objects are grouped into classes.
- An object is an instance of a class.



From Real World

Class: Professor

Professor Jones

Professor Smith

Objects

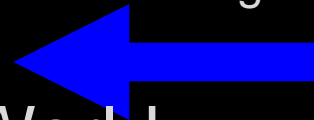
abstracting



Professor Mellon

Professor
- name
- employeeID : UniqueId
- hireDate
- status
- discipline
- maxLoad
+ submitFinalGrade()
+ acceptCourseOffering()
+ setMaxLoad()
+ takeSabbatical()

instanting



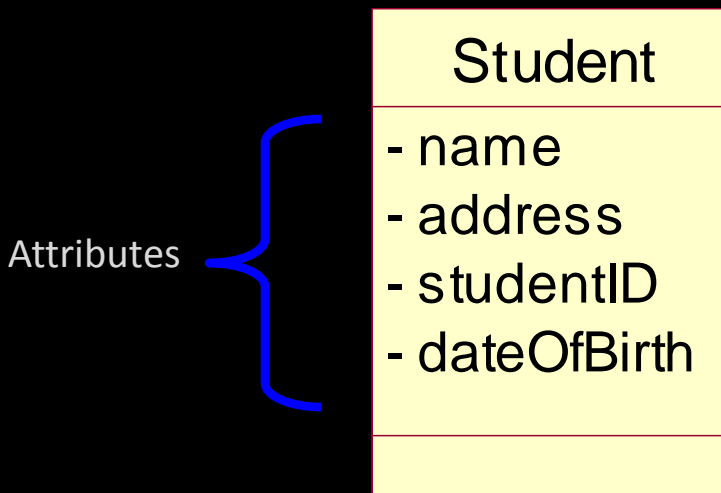
J Clark :
Professor

Objects

To computer World

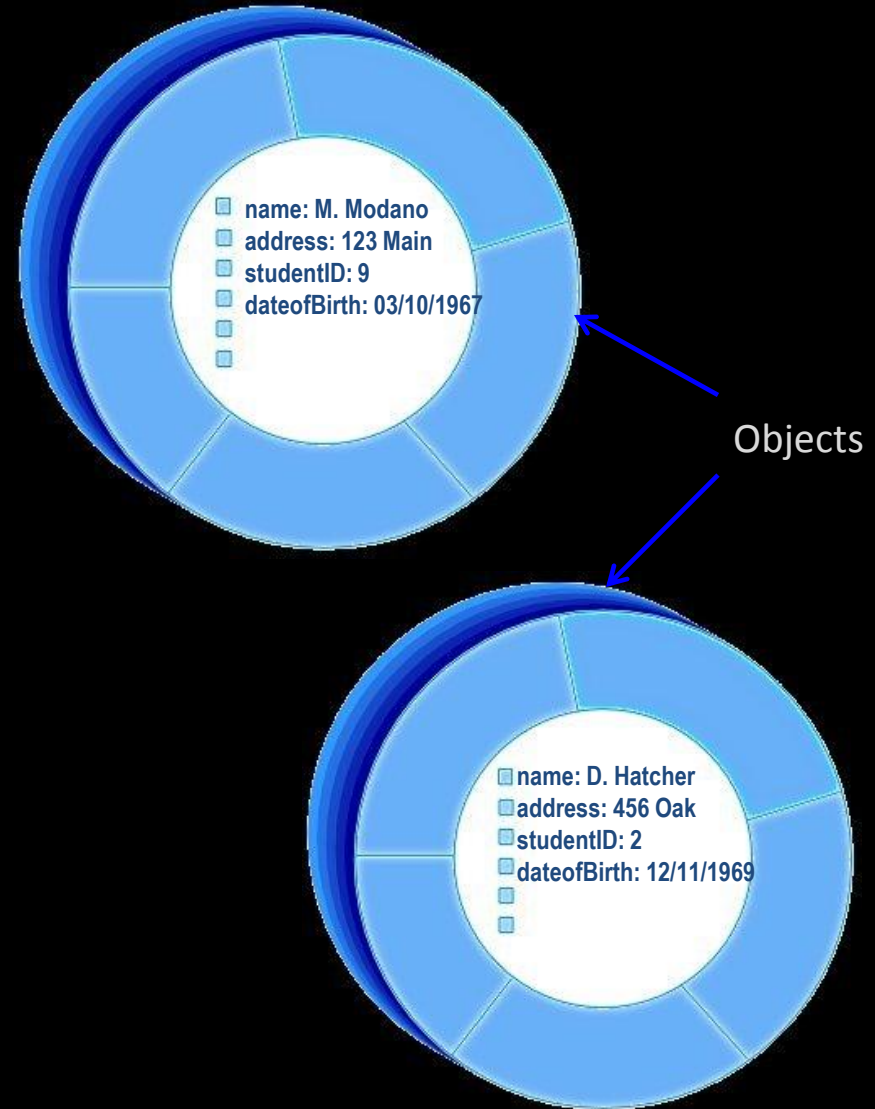
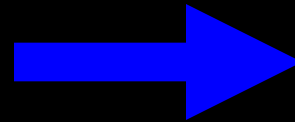
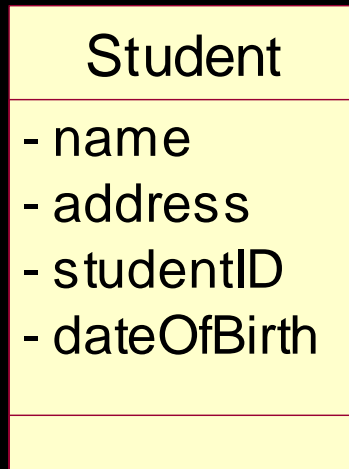
What Is an Attribute?

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



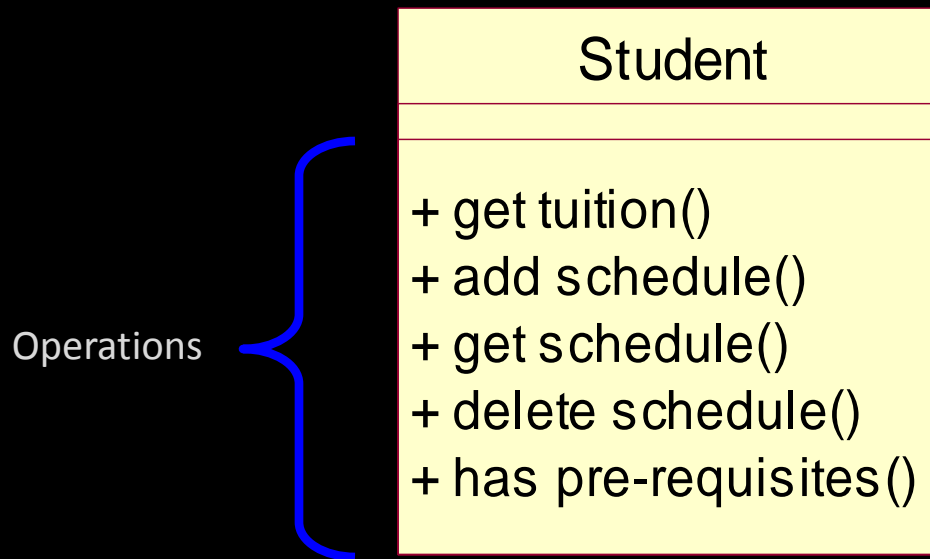
Attributes in Classes and Objects

Class



What Is an Operation?

- An operation is the implementation of a service that can be requested from any object of the class to affect behavior.
- A class may have any number of operations or none at all.



Example: class Professor

```
class Professor {  
    private String name;  
    private int age;  
    private String speciality;  
  
    public Professor (String sm, int ia, String ss) {  
        name = sm;  
        age = ia;  
        speciality = sst;  
    }  
    public String getName () { return name;}  
    public int getAge () { return age;}  
    public String getSpeciality () {  
        return speciality;}  
}
```

Professor

- name : String

- age : int

- speciality : String

+getName() : String

+getAge() : int

+getSpeciality() : String

Example : Instance of Professor

wang : Professor

name = “wang”

age = 35

speciality = “computer”

Professor wang = new Professor (“wang”, 35,
“computer”);