Class vs. Object

- A class is a collection, a group, a set, or a type of...
e.g.
  - People
  - Student

- An object is an instance or an individual of a class
e.g.
  - People Tom
  - Student Marry

Class and its UML Box

- Data
  - State, attributes

- Behavior
  - Action, transformation, operation
    - Triggered by receipt of particular message, or entrance into a particular state

<table>
<thead>
<tr>
<th>Class name</th>
<th>Bill</th>
</tr>
</thead>
<tbody>
<tr>
<td>Attributes</td>
<td>item, date, due, price, amount</td>
</tr>
<tr>
<td>Operations</td>
<td>purchase()</td>
</tr>
</tbody>
</table>

Relations between Classes

- Subclass
e.g.
  - People
  - Student

- Hierarchy

- Inheritance (is-a)

Association relation between classes

- Public Class Employee

```java
public class Employee {
    public String lastName;
    public Address homeAddress;
    // ...
}
```

- Public class Address

```java
public class Address {
    // ...
}
```

Association relation between classes (bidirectional)

- Public Class Team

```java
public class Team {
    public LinkedList members;
    // ...
}
```

- Public class member

```java
public class Member {
    Team memberOf;
    // ...
}
```
### Composition and Aggregation

- **Subperson**
- **Order**

```
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>M</td>
</tr>
</tbody>
</table>
```

- **Ordered Item**
- **Customer**

```
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>*</td>
</tr>
</tbody>
</table>
```

- **Dependency**
  - A dependency exists between the two if a class uses another class in some fashion

```
[Diagram showing dependency]
```

### Using Class Diagram to show a design

- **Message Class**
- **Ordering System**
- **Credit Card System**

```
[Diagram showing class relationships]
```

### More UML diagrams

#### Requirements
- **Use Case Diagram**
- **UML Sequence Diagram**

#### Design
- **Activity Diagram**
- **UML Collaboration Diagram**

#### Coding
- **UML Class Diagram**
- **UML State Diagram**

### Use case diagram

- **Customer**
- **Credit Card System**

```
[Diagram showing use case]
```

### Activity Diagram

- **Activity A**
- **Activity B**

```
[Diagram showing activity flow]
```
**OO Design**

- Identify classes
  - A set of objects sharing a common structure and common behaviors
- Looking for Things (nouns)
  - Structures
  - External systems
  - Devices
  - Roles
  - Operating procedures
  - Places
  - Organizations
  - Things that are manipulated by the system to be builds
- Behaviors
  - Verbs

### OO Measurement: Metric

<table>
<thead>
<tr>
<th>Phase</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use cases</td>
<td>Class diagrams, Interaction diagrams, Class descriptions, State diagrams, Package diagrams</td>
</tr>
<tr>
<td>Number of scenario scripts</td>
<td>Number of key classes, Number of support classes, Average number of support classes per key class, Number of subsystems, Class size, Number of operations overridden by a subclass, Number of operations added by a subclass, Specialization index, Weighted methods in class, Depth of inheritance, Number of children, Coupling between objects, Response for a class, Lack of cohesion in methods, Average operation size, Average number of parameters per operation, Operation complexity, Percent public and protected, Public access to data members, Number of root classes, Fan-in/fan-out</td>
</tr>
</tbody>
</table>