Object Orientation

Max J. Egenhofer
University of Maine
Orono, USA
max@spatial.maine.edu
http://www.spatial.maine.edu/~max
Before Objects

- **Typed Programming Languages**
  - **Simple data types**
    - integers, floats, characters, strings
    - records, arrays
  - **Abstract data types**
- **Database Systems**
  - Hierarchical, network, relational
- **Major drawback: impedance mismatch**
Object Orientation

- First concepts of objects and classes in SIMULA

Ole-Johan Dahl and Kristen Nygaard
Object Orientation

- First concepts of objects and classes in SIMULA
- Smalltalk as first object-oriented programming languages (with message passing)
Object Orientation

- First concepts of objects and classes in SIMULA
- Smalltalk as first object-oriented programming languages (with message passing)
- C++ led to the rise of object-oriented programming
Object-Oriented Databases

- Model, store, access, and manipulate entire objects (at a higher abstraction level than tuples)
- Necessary for complex applications domains (engineering, GIS)
- Overcome the impedance mismatch when programming with databases
Three Levels of Object Orientation

*structural*
any entity—indeed, independent of complexity or structure—may be represented by exactly one object

*operational*
operations on a complex object do not require to decompose it into simpler objects

*behavioral*
access to and modification of objects is possible only through a set of operations specific to that object’s type
Inheritance

- **Operations** are defined upon a class
- In generalization, all operations of the superclass apply to its subclasses (and transitively to their subclasses)
Single Inheritance

A class inherits *behavior* from a single immediate superclass (and recursively from that class’s superclass)
Common properties of area?
- >0
- \(\leq\) MBR
How to implement this operation?

Area of Shape has no implementation.

A = \frac{1}{2} \sum_{i=0}^{n-1} x_i y_{i+1} - x_{i+1} y_i

A = a^2

A = a \cdot b

A = \text{base} \cdot \text{height} / 2

A = \pi r^2

A = \pi a b
Inherited Behavior

• Subclasses may be the implementors of inherited behavior

• Subclasses may overwrite the inherited behavior (e.g., by imposing further restrictions)
Multiple Inheritance

A class may inherit behavior from multiple immediate superclasses
Multiple Inheritance

A class may inherit behavior from multiple immediate superclasses

```
<table>
<thead>
<tr>
<th>toy</th>
<th>truck</th>
</tr>
</thead>
<tbody>
<tr>
<td>size</td>
<td>transportsItems</td>
</tr>
<tr>
<td>weight</td>
<td>hasWheels</td>
</tr>
</tbody>
</table>

toyTruck
```
Inheritance Conflicts

The operation with the same name is inherited from multiple superclasses

No issue if they are identical, but otherwise …
Inheritance Conflicts

average (rounded to 4 decimals)

average (rounded to 2 decimals)
Inheritance Conflict Resolution

- Select the implementation of the *first / last* class that passes on the property
- Select the *most restrictive* implementation among those passed on
- Create the most restrictive implementation as the *intersection of the constraints* of all inherited properties with the same name
- Manually (i.e., needs to be specified explicitly)
Multiple Inheritance

A class inherits behavior from multiple immediate superclasses (and, recursively their superclasses)

Inheritance conflicts even more tricky