

Software Design
CSE 335, Spring 2006

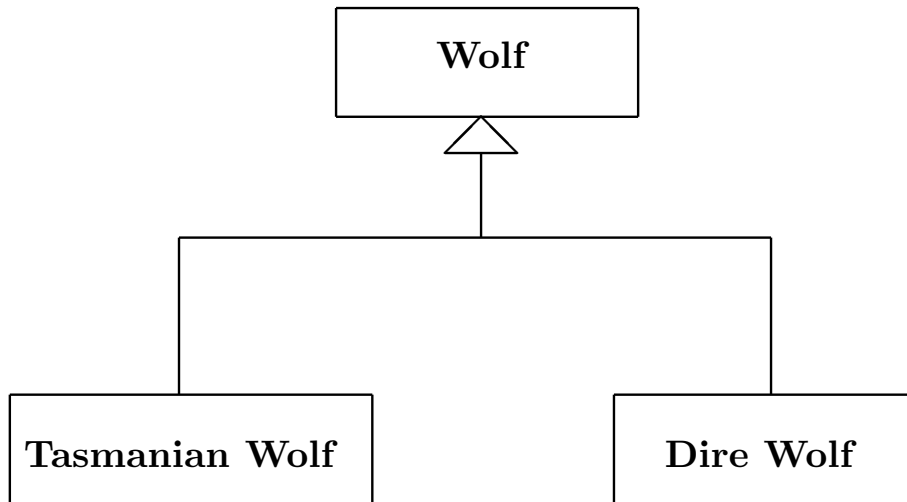
Constraints and Tricky Hierarchies

Stephen Wagner

Michigan State University

A Tricky Natural Hierarchy

What is wrong with this model?



Tricky Hierarchies

- It is not always obvious how things are related
- Multiple inheritance helps
- Sometimes it helps to think in terms of exceptions
 - Not all Birds can fly
 - It is still useful to think that birds fly, and just account for the special cases

Shapes Again

- How would you organize the following into a generalization hierarchy (using multiple inheritance if necessary)?
 - Polygon
 - Square
 - Shape
 - Rhombus
 - Trapeziod
 - Quadrilateral
 - Rectangle
- What should the Square class look like?

Inheritance/Generalization

- public inheritance \Rightarrow Generalization always
- Generalization \Rightarrow inheritance when the derived class extends the base class
 - specializes the base class
 - makes the base class *play a role* in a new collaboration
- \neg (Generalization \Rightarrow inheritance) when derived class *restricts* base with a *constraint* on the attributes or operations

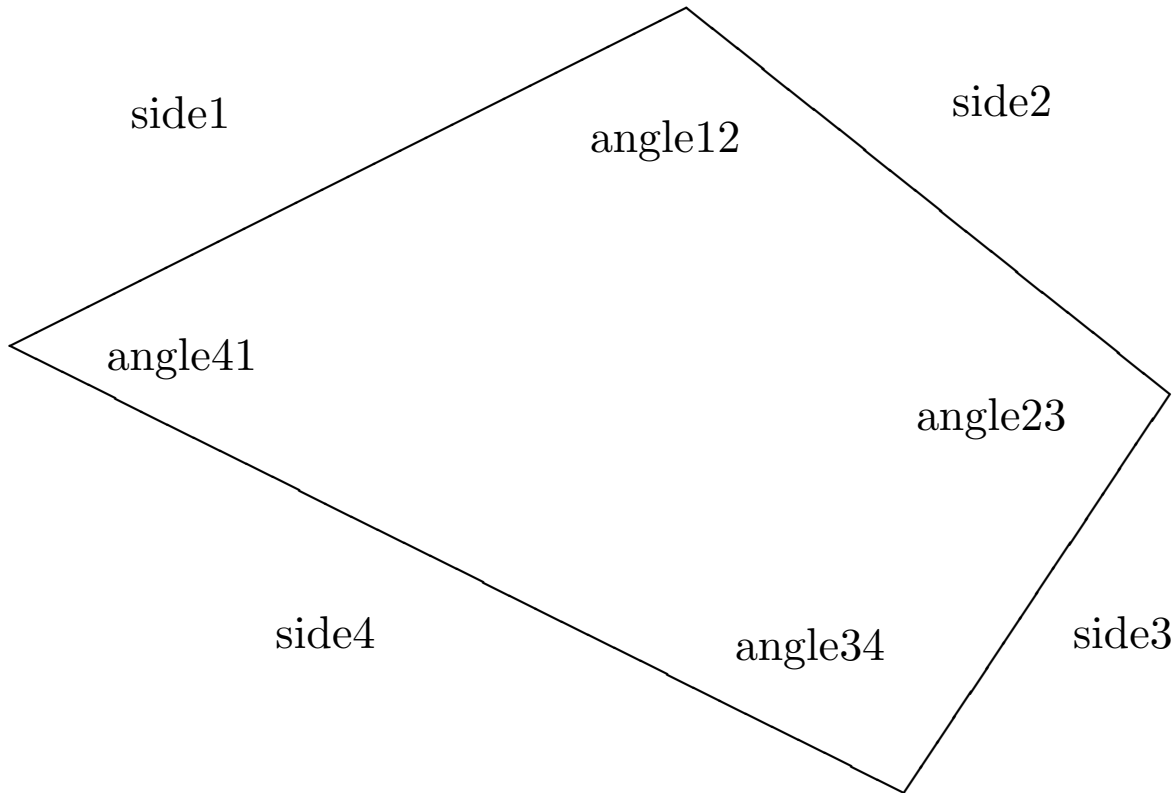
Constraint

- Defn: Functional relationship between entities in a class model
 - Entities include objects, classes, attributes and associations
 - Delimited by curly brackets and placed near the constrained entity
 - Expressed declaratively as in *invariant*
- e.g. no window will have an aspect ratio less than or equal to .8 or greater than or equal to 1.5

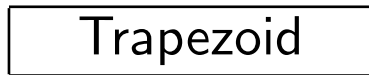
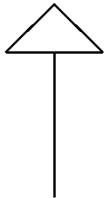
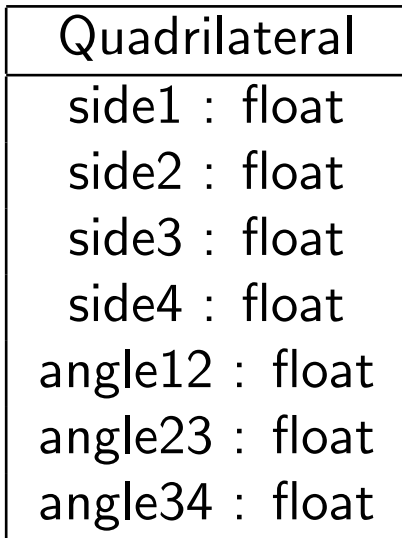
Window
length unsigned width unsigned

{ 0.8 < length/width < 1.5 }

Quadrilateral Reference Model

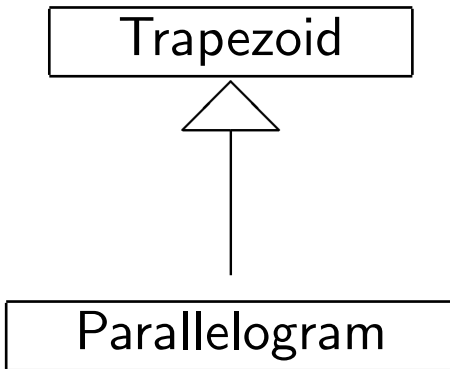


Example



$$\{ \text{angle23} + \text{angle34} = 180 \vee \\ \text{angle12} + \text{angle23} = 180 \}$$

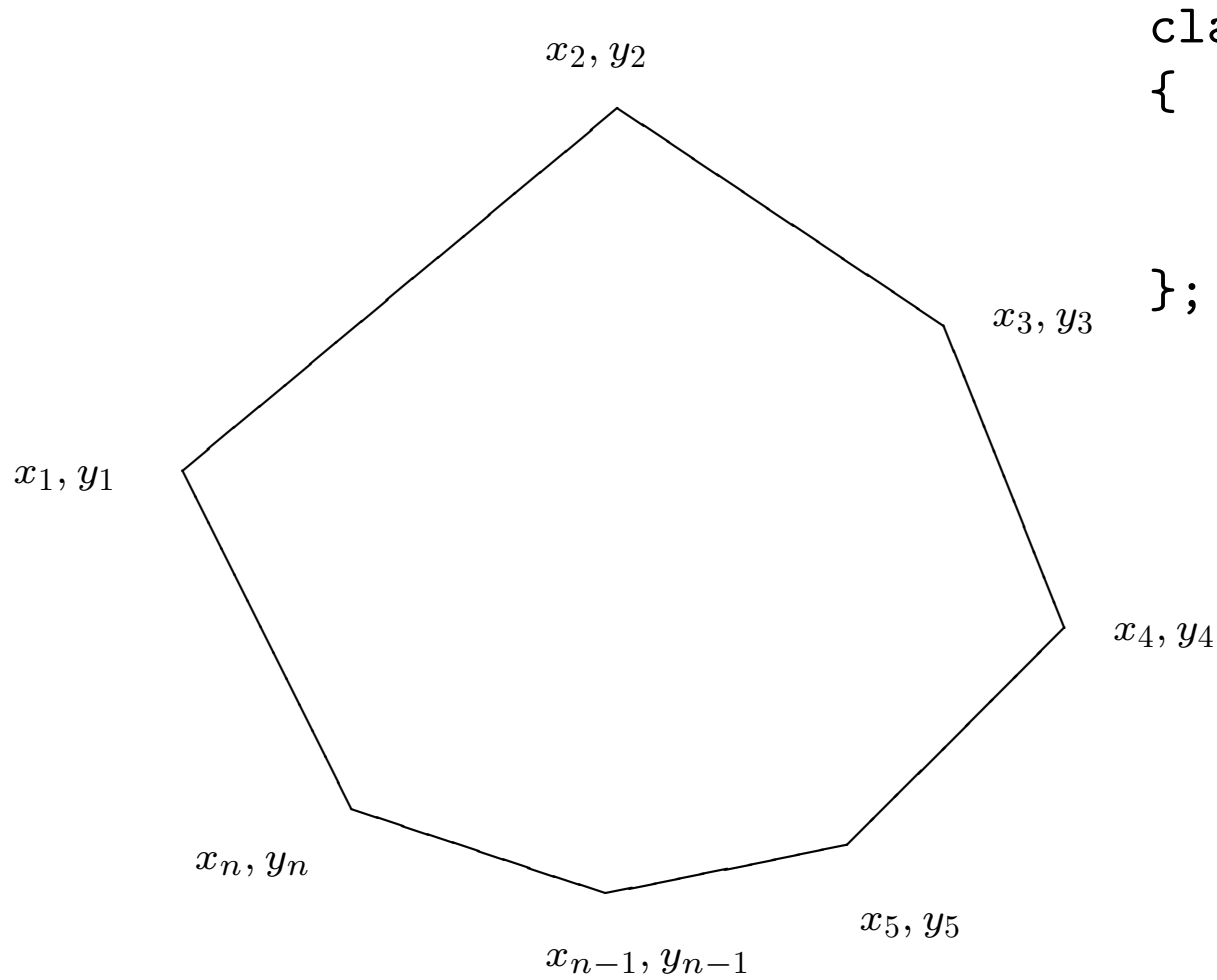
Other Examples



$$\{ \text{angle23} + \text{angle34} = 180 \vee \\ \text{angle12} + \text{angle23} = 180 \}$$

$$\{ \text{side1} = \text{side3} \wedge \\ \text{side2} = \text{side4} \}$$

A Model for Polygons



```
class Polygon
{
    protected:
    vector<point> points;
};
```

Constraints

1. What are the constraints for a quadrilateral?
2. What are the constraints for a trapezoid?
3. What are the constraints for a square?

Questions

Suppose we wanted to implement the Polygon hierarchy using C++ classes and inheritance

1. Suppose Polygon defines the functions

```
void MovePoint (int pointnum, int newx, int newy );  
void InsertPoint (int before, int newx, int newy );
```

How would we implement these in Rhombus or Square?

2. What are the practical disadvantages of having Square inherit all the attributes of Polygon

Restrictions

- Defn: A derived class that constrains one or more attributes of one of its base classes
- Two general schemes for implementing a restriction
 - Outlaw operations that violate the constraint
 - Allow objects to change classes
- Languages such as C++ do not support either option very well