Object Oriented Design

Worked Example

What should be a class?

Class Identification

- Decompose domain into classes
  - Self-contained, has all data and functions for task
  - Focused on specific purpose
- Typically you can brainstorm possible classes
  - Take the list and look for has-a or is-a relations
  - Assign functions to entities or item
- Develop the class hierarchy
- Much of this may be done if expanding an existing system

Class Identification

- Need a program to simulate airport operations
- Need to study throughput of passengers through the airport
  - More passengers
  - Less time
  - Fewer facilities
  - More money

How Does an Airport Work?

Passengers Arrive	Land
Check baggage	Taxi
Wait at gate	Skyway positioned
Load on plane	Unload
Taxi	Baggage Claim
Take off	Leave Terminal

Is this all?

Do variations matter?

Such as checking baggage at the gate?

Class Identification

One way

- Classes for

Passengers	Aircrew
Gates	Food?
Airplanes	Gate Agents?
Taxiways	Baggage?
Runways	

- Physical items

Class Identification

Another way

- Classes for

Gate	Approach
Taxiing	Flight
Take Off	Landing
Runways	

- processes
- More abstract

Class Identification

- Which is “right”?

- It depends
- What do you want to do?

Classes as Physical Items

- Typically easier to “see”
- Visualize planes moving around airport
- Visualize people moving on/off planes
- Different types of planes are ‘obvious’ subclasses of a plane super class

Classes as Abstract Entities

- May not be as obvious
- Can easily spiral out of control
  - Subdivide too much
- Can focus more on processes
- May be less opportunity for inheritance
  - But is that a bad thing?

Refine class hierarchy

- As you develop the classes collect common features in a superclass
- You see that janitor, typist, and guard all have hourly pay
  - Create a waged employee class between them and employee
- You see that IT support has exempt/non-exempt, supervises or not, different network access, on-call or not
  - Create different IT support subclasses?

Refinement, Practical View?

- You find you have a class with many if or switch statements making distinctions
  - You are losing focus and probably need to create more sibling subclasses
- You find yourself pasting code into multiple classes
  - You are not sharing the structure and probably need to create a parent superclass

Summary

- Identify possible classes, physical or abstract
- Look for groupings, is-a or has-a relations
- Look for common data/functions to pull into a parent superclass
- Look for too much distinctions being made inside a class to identify possible sibling classes
- Create your hierarchy