

# Practical Object-Oriented Design Course 1

## Sandi Metz

### Day 1

#### Exercise: '99 Bottles of Beer'

##### Lessons

- Considering simplicity versus complexity
- Finding a "shameless green" solution
- Refactoring to increase the level of abstraction
- Modeling concepts as objects
- Understanding the Single Responsibility Principle
- Following the Open/Closed Principle
- Obeying the Liskov Substitution Principle
- Making immutable objects
- Caching existing versus creating new objects
- Using polymorphism to replace conditionals
- Sharing behavior via Inheritance

##### Techniques

- Code Smells
- Refactoring
- Flocking Rules
- The Transformation Priority Premise
- Extract Method
- Extract Class

##### Metrics

- Squint test
- Inebriation test

### Day 2

#### Exercise: '99 Bottles', continued

##### Lessons

- Isolating conditionals in Factories
  - Six common factory variants
  - Pros and cons of each
- Metaprogramming

Benefits, constraints, trade-offs  
Monkey Patching  
Benefits, constraints, trade-offs  
SOLID Design Principles Overview

## Exercise: 'Farm'

### Lessons

Recognizing fundamental code shapes that guide OOD  
Isolating code from external dependencies  
Mocking only code you that own

### Techniques

Null Object Pattern  
Decorator Pattern  
Chain of Responsibility Pattern

## Exercise: Applying Lessons to the Real World

### Lessons

Dealing with the real world  
Applying these lessons to your own code

## Day 3

## Exercise: 'The House that Jack Built'

### Lessons

Practicing 'shameless green'  
Practicing refactoring via the 'flocking rules'  
Practicing Inheritance  
Recognizing when inheritance fails  
Converting from inheritance to composition  
Injecting composed dependencies  
Simplifying tests using composition and dependency injection

### Techniques

Inheritance revisited  
Dependency Inversion Principle  
Inverting Dependencies by injecting roles  
Testing Commands vs Testing Queries  
Using mocks vs using stubs in testing