

Practical Object-Oriented Design Course

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 instances versus creating new objects

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

- Polymorphism
- Inheritance
 - Problems suited to Inheritance
 - When to avoid inheritance
- Liskov revisited
- Concepts as Objects

Variations on Factories
Metaprogramming - the good, bad and ugly
Monkey Patching - the good, bad and ugly
Overview of the SOLID Design Principles

Techniques

Replace Conditional with Polymorphism

Day 3

Exercise: 'Farm'

Lessons

Recognizing fundamental code shapes that guide OOD
Isolating your code from external dependencies
Mocking, when and how

Techniques

Null Object Pattern

Exercise: Applying Lessons to the Real World

Lessons

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

Exercise: 'The House that Jack Built'

Lessons

Practicing 'shameless green'
Practicing refactoring via the 'flocking rules'
Using Inheritance
Converting from inheritance to composition

Techniques

Inheritance revisited
 Template Method Pattern
 Hook Methods
Dependency Inversion Principle
Inverting Dependencies by injecting roles