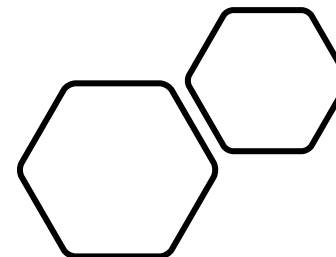




**Northumbria  
University**  
NEWCASTLE



# Patterns & Principles In Object Oriented Programming

KF6012  
Web Application Integration  
2020

# Lecture Overview

- Patterns in OO Programming
- Principles in OO Programming

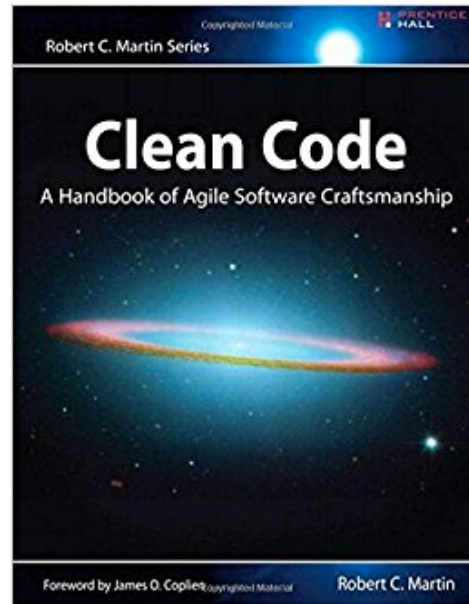
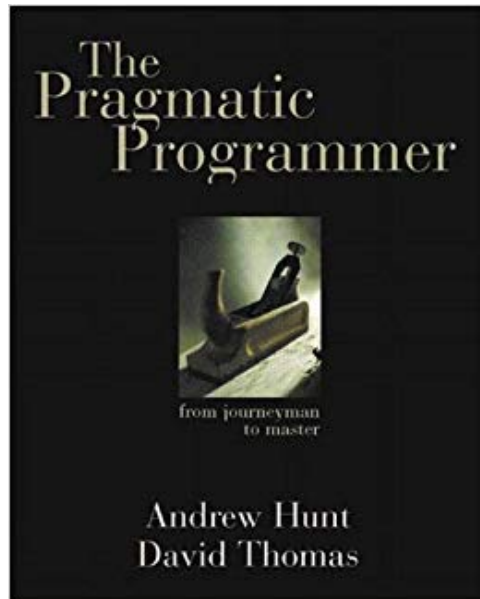
# Patterns and Principles

- OO is a way of (better) organising code
  - So that:
    - You can manage the complexity
    - You can maintain it
    - You can scale up
    - You can work with others
- There is more to OO beyond valid code
  - OO is an “art” or “social skill”
  - There are patterns and principles you *ought* to follow



# Principles

- There are high level principles associated with OO programming, which offer general guidance about good practice.
- “Higher level” programming books that go beyond how-to and look at programming as practice explain and explore principles and “craft”



# Principles

DRY: Don't Repeat Yourself

# Principles

## DRY: Don't Repeat Yourself

- Every piece of knowledge must have a single, unambiguous, authoritative representation within a system.
  - Saves you from having to make multiple changes to do one thing.

# Principles

- S.O.L.I.D.
  - **S**ingle Responsibility
  - **O**pen-Close Principle
  - **L**iskov Substitution
  - **I**nterface Segregation
  - **D**ependency Inversion

# Principles

- S.O.L.I.D.

- **S**ingle Responsibility

- **O**pen-Close Principle

- **L**iskov Substitution

- **I**nterface Segregation

- **D**ependency Inversion

A Class should be responsible for a single task.

A Class should be open to extension and closed to modification..

A derived Class can be substitutable for wherever a base Class is used.

Make many client specific interfaces rather than one general purpose

Depend on abstractions, not on concretions.



# Patterns

- Patterns are like principles, but often at a lower level and more directly applicable to your code and software architecture.

# Patterns

- Factory
- Front Controller
- MVC
- Singleton
- Strategy

# Patterns

- Factory
- Front Controller
- MVC
- Singleton
- Strategy

A Class that creates and returns an object (usually variants of objects)

A single entrance point to your app (e.g. index.php or api.php)

The app is broken up into a “model” manage data access, a “controller “ to manage requests, and a “view” to handle responses/display.

Allowing access to one and only one instance of a class

The class includes various algorithms or strategies to choose from but the client has no knowledge of or interest in this choice.

- [John.rooksby@northumbria.ac.uk](mailto:John.rooksby@northumbria.ac.uk)