

# OBJECT-ORIENTED PRINCIPLES & PATTERNS

- This training course provides software engineers with a comprehensive introduction to Object-Oriented Design Principles and reusable Design Patterns.
- The OO Principles are the underlying guidelines for all OO designs. With this knowledge creating or extending an OO design is clearer and easier.
- Design Patterns have changed the way of software development. At a minimum its good OO design by example, fully applied, they are proven and elegant OO design solutions that will bootstrap your application design.
- The technical explanation is combined with examples, exercises and 'in practice' guidelines.
- This course is a formal training session designed for practicing software engineers with a minimum of 1-2 years experience with object-oriented design concepts and techniques.
- This is a one-day course that can be conducted in two half-day sessions.

## COURSE OUTLINE

### Design Principles:

- Open/Closed Principle
- Liskov Substitution Principle
- Principle of Dependency Inversion
- A Class Captures One Key Abstraction
- Generalise up the Inheritance Hierarchy
- Interface-Segregation



*"High applicability to my work... structure of morning and afternoon very good"*

Robert Powers-Martin, IT Specialist / Systems Analyst - BankWest



*"I learnt how to apply design patterns, was fun, good balance of theory and practical workshops"*

James Ioppolo, Software Engineer - Immersive Technologies

### Design Patterns:

- Benefits of Patterns
- Essential Elements of Patterns
- Creational Patterns
  - Abstract Factory
  - Singleton
- Structural Patterns
  - Decorator
  - Proxy
- Behavioural Patterns
  - Observer
  - Mediator

### Practical Sessions:



A workshop exercise is conducted to review and refactor an initial design to comply with several Design Principles



Another exercise is undertaken to refactor a "brute-force" design to improve its flexibility, reuse and memory usage by incorporating Design Patterns

