

# Table des matières

- Reviews
- Preface (p. 6)
- **0 Introduction** (p. 8)
  
- 1. **Abstraction** (p. 12)
- 2. **Classes** (p. 27)
- 3. **Handles** (p. 67)  
4–5 [non listés]
- 4. **Designing for Inheritance** (p. ?)
- 5. **Templates** (p. 144)
  - 7.1 A Pair class template
  - 7.2 Some template details
  - 7.3 Template instantiation
    - 7.3.1 Implementation details
  - 7.4 Smart pointers
    - 7.4.2 A use-counted smart pointer template
  - 7.5 Expression arguments to templates
  - 7.6 Function templates
  - 7.7 In short
  - 7.8 Questions
- 6. **Advanced Templates** (p. 168)
- 7. **Reusability** (p. 207)
- 8. **Exceptions**
  - 10.2 Why exceptions
  - 10.3 An example exception
  - 10.4 Exceptions should be exceptional
  - 10.5 Understanding exceptions
  - 10.6.3 The provider of a resource
  - 10.6.4 The user
  - 10.7 Designing the exception object
    - 10.7.1 Exceptions and inheritance
  - 10.8 In short
  - 10.9 Questions
- 9. **Moving your project to C++**
  - 11.1.1 Learning curve
  - 11.1.2 Language and tools maturity
    - 11.2.1 C++ as a better C
    - 11.2.4 Three steps vs. one step
  - 11.3 Design and implementation
  - 11.4 Developing an asset base
  - 11.6 In short
  - 11.7 Questions