Chapter 4
Impossible Perfection

• If perfect code is impossible, what strategies can we adopt to achieve the highest quality possible?
Design by Contract

• Compare Design by Contract with Test-Driven Development
• What does it mean that DbC code is “lazy”? 
Liskov Substitution Principle

• “Subclasses must be usable through the base class interface without the need for the user to know the difference”
  
  – i.e. a subtype “is-a-kind-of” the base type, it can be substituted
  
  – i.e. subtypes honor all the contracts of a base class
Crash Early and Asserts

- What assert mechanisms exist in the languages you are using for this project?
Exceptions

• “exceptions should rarely be used as part of a program’s normal flow; exceptions should be reserved for unexpected events.”
  – Code should still run even if all exception handler code is removed
Finish What You Start

• The routine that allocates a resource should deallocate it
  – Often done with constructors and destructors
  – Problems with exceptions
    • C++: deallocation must occur in normal and all exceptional paths
      – use local objects rather than pointers to objects
      – If not possible, use a wrapper (e.g. auto_ptr)
    • Java: garbage collector may not run immediately
      – use finally clause