Chapter 6
Deliberate Programming

- Don’t Program by Coincidence
  - Know why something works
  - If something breaks, understand why
- How to program deliberately
  - Understand the technologies
  - Test everything with automated tests
  - Don’t let existing code dictate future code

Algorithm Speed

- O() fundamentals
  - You should know this from other courses
- Don’t optimize prematurely
- Confirm with code profilers as necessary

Refactoring

- Software is more like gardening than construction
- Refactoring: changing the internal structure of code without changing its external behavior
  - Don't try to refactor and add functionality at the same time
  - Have good tests and run them often when refactoring
  - Take short, deliberate steps
- Become familiar with automated refactoring tools
- Refactor early and often (don't put it off, you will never have more time)
- See www.refactoring.com

Testing

- Design to Test
- Test against contracts
- Use test harnesses

Evil Wizards

- Why is it acceptable to use library calls or OS services without fully understanding how they are implemented, but not acceptable to use code generated by wizards without fully understanding it?
  - Wizard code becomes an integral part of *your* application that *you* will have to maintain and extend