Programming Languages and Techniques (CIS120e)

Lecture 37
Dec. 10, 2010

Recap
FINAL EXAM

• Thursday, December 16th noon – 2:00
• Moore 212

• Comprehensive exam:
  – OCaml material
  – Java material
  – all course content

• Closed book, no notes
What We’ve Seen...
Basic design recipe:

1. Understand the problem
   What are the relevant concepts and how do they relate?
2. Formalize the interface
   How should the program interact with its environment?
3. Write test cases
   How does the program behave on typical inputs? On unusual ones? On erroneous ones?
4. Implement the required behavior
   Often by decomposing the problem into simpler ones and applying the same recipe to each
Unit Testing

• A topic unto itself!
  – Easy for declarative programs (or subprograms)
  – Trickier for stateful programs / subprograms
  – Very tricky for GUIs

• Write tests before code
  – “Tests first” methodology
Fundamental Design Elements

• recursion
• collections
  – lists, trees, sets, maps
• pointers and aliasing
• types (interfaces, generics)
• abstraction / information hiding
• abstract machines (stack, heap, class table, etc.)
  – abstract models of computation (turing machines...)
Programming Idioms

• declarative / recursive

• higher-order
  – “functions as data”

• imperative / iterative
  – iterators
  – looping over arrays

• object-oriented
  – encapsulate code with data

• event-driven

• domain-specific idioms
  – map-reduce
Style

• Locality
  – Narrow interfaces
  – Information hiding

• Non-repetition
  – Never say the same thing twice; instead, abstract

• Readability
Onward...
What Next?

• Classes:
  – CIS 121 – datastructures, performance, computational complexity
  – CIS 19x – programming languages (C++, C#, Python, Haskell,...)
    • Note: new Advanced Java Programming will be offered next Fall
  – CIS 240 – lower-level: hardware, gates, assembly, C programming
  – CIS 341 – compilers (projects in OCaml)
  – ... and many more! :-)

• Undergraduate research
The Craft of Programming

- **The Pragmatic Programmer:**
  *From Journeyman to Master*
  
  by Andrew Hunt and David Thomas
  
  - Not about a particular programming language, it covers style, effective use of tools, and good practices for developing programs.

- **Effective Java**
  
  by Joshua Bloch
  
  - Technical advice and wisdom about using Java for building software. The views we have espoused in this course share much of the same design philosophy.