Course Themes

- Programming Language Concepts
  - A language is a “conceptual universe” (Perlis)
  - Frameworks for problem-solving
  - Useful concepts and programming methods
  - Understand the languages you use, by comparison
  - Appreciate history, diversity of ideas in programming
  - Be prepared for new programming methods, paradigms, tools
- Critical thought
  - Identify properties of language, not syntax or sales pitch
  - Language and implementation
    - Every convenience has its cost
    - Recognize the cost of presenting an abstract view of machine
    - Understand trade-offs in programming language design

Value of Language Concepts

- Ancient history
  - I started programming in 1970’s
  - Languages were Fortran; no recursive functions
  - My algorithms and data structure instructor said:
    - Recursion is a good idea even though inefficient
    - You can use it in Fortran by storing stack in array
  - Today: recursive functions everywhere
- Moral
  - Futuristic ideas may be useful problem-solving methods now, and may be part of languages you use in the future
- More recent examples
  - Function passing: pass functions in C by building your own closures, as in STL “function objects”
  - Continuations: used in web languages for workflow processing
  - Monads: programming technique from functional programming
  - Concurrency: atomicity instead of locking

What’s new in programming languages

- Commercial trend over past 5+ years
  - Increasing use of type-safe languages: Java, C#, ...
  - Scripting languages, other languages for web applications
- Teaching trends
  - Java replaced C as most common intro language
  - Less emphasis on how data, control represented in machine
- Research and development trends
  - Modularity
    - Java, C++: standardization of new module features
    - Program analysis
      - Automated error detection, programming environment, compilation
    - Isolation and security
      - Sandbox, language-based security, ...
    - Web 2.0
      - Increasing client-side functionality, mashup isolation problems

What’s worth studying?

- Dominant languages and paradigms
  - Leading languages for general systems programming
  - Explosion of programming technologies for the web
- Important implementation ideas
- Performance challenges
  - Concurrency
- Design tradeoffs
- Concepts that research community is exploring for new programming languages and tools
First half of course

- Introduction, JavaScript
- Haskell
  - Functional lang. used in lectures 5-7 and concurrency
- Foundations
  - Lambda calculus, structured operational semantics, ...
- Scope and stack storage allocation
- Types and parametric polymorphism
- Type classes – ad hoc polymorphism in Haskell
- IO Monad – example of Haskell monad concept
- Exceptions and Continuations
- Modularity (not on midterm)

------------------ Midterm Exam ------------------

Wed October 26, 7-9 PM

Second half of course

- Objects
- Prototypes, classes, inheritance
- Object types and subtyping
- Implementation structures
- Templates and Generics
- Concurrency
  ------------------------ Thanksgiving Break ------------------------
- Software Transactional Memory
- Atomicity and advanced topics
- Review
  ------------------ Final Exam ------------------

Assignments and Grading

- Homework 40%
  - Weekly: Wednesday to Wednesday
    - 3 late days per quarter, MAXIMUM ONE EACH WEEK
- Exams
  - Midterm: two hour evening exam 25%
  - Final: regularly scheduled exam time 35%
- Programming (JS, Haskell, C++, Java)
  - Some programming assignments, auto-graded
  - Part of weekly homework

Course Logistics

- Policies and procedures
  - See information on web: CourseWare CS242 FAQ
  - Honor Code, Collaboration Policy
- Homework grader?
  - Send email to cs242@cs.stanford.edu email address
- CA's, Office hours, Email policy, ...
  - Use CourseWare for all discussion – no email discussion!!!
- Section
  - Friday afternoons, 2:15-3:05 (pending scheduling)
    - Optional discussion and review; no new material
- Reading material
  - Book available in bookstore (Concepts in Programming Lang)
  - Supplementary reading on web

See web site...

General suggestions

- There is something difficult about this course
  - May be hard to understand homework questions
    Thought questions: cannot always run and debug
    May sound like there is no right answer, but some answers are better than others
  - Many of you may be used to overlooking language problems, so it takes a few weeks to see the issues

Questions?