Programming Languages: The Big Picture

• The computer speaks binary
• You speak natural language

• Programming languages seek to bridge the gap
  – ease of coding
  – efficiency of final code

• No programmer is an island (anymore)
• Neither are computers

• Same themes, new variations
The “Translator”

- Same goal, many manifestations
- Compiled, direct executable
- Interpreted source code, fed into a running interpreter
- Compiled, intermediate code, executed within a virtual machine

Additional issues
- Don’t reinvent the wheel: managing modules and libraries, linking
- Combining compilation and interpretation for fun and profit
  - Intermediate code
  - Just-in-time compilation
- Supporting multiple machine and processor types and architectures
  - “front-end” vs. “back-end” compilation
  - Preprocessing: macro expansion, external conditionals
- “Build a better mousetrap” — the search for smaller, faster code never ends
  - Optimization at different levels
  - Theoretical foundations — proving a program’s correctness

The Translation Process

- A program = a character stream
  - Scanner (lexical analysis)
- Token stream
  - Parser (syntax analysis)
- Parse tree
  - Semantic analysis and intermediate code generation
- Abstract syntax tree or other intermediate form
  - Machine-independent optimization
- Modified intermediate form
  - Target code generation
- Assembly, machine, or other target language
  - Machine-specific optimization
- Modified target language
  - Rinse and repeat
A Language for Everything

- Programming languages
  - declarative
    - functional
      - Lisp/Scheme, ML, Haskell
  - data flow
    - Id, Val
  - logic, constraint-based
    - Prolog, VisiCalc

- imperative
  - Von Neumann
    - Fortran, Pascal, Basic, C
  - object-oriented
    - Smalltalk, Eiffel, C++, Java

- “Other” languages
  - What is “programming,” and what is not?

- document specification
  - “paper-like” documents
    - HTML, XML, LaTeX
  - graphs and diagrams
    - ER, UML

- information systems
  - querying
    - SQL, Query-by-Example
  - data definition
    - SQL, ER, UML

- focused applications
  - development
    - make, ant
  - game scripting
    - QuakeC, UnrealScript
  - multimedia
    - Lingo