Introduction, or Why Bother With This Stuff, Anyway?

- Increased capacity to express ideas.
- Improved background for choosing languages.
- Increased ability to learn languages.
- Understanding of significance of implementation.
- Ability to design new languages.
- Overall advancement of computing.

Language Evaluation Criteria

- Readability & Writability
  - Simplicity
    - small # of basic components
      (subsets a poor solution!)
    - one syntax: one meaning
      Counter-example in C: four ways to increment x:
      \[ x++; \]
      \[ x=x+1; \]
      \[ x+=1; \]
      \[ ++x; \]
      In FORTRAN, two meanings for:
      \[ Y=\text{SUM}(I,J) \]
      -- array reference
      -- function call

Language Evaluation Criteria

→ Orthogonality
  ⇒ Any composition of basic primitives is allowed
  ✓ need small set of primitives, ways to combine them
  ✓ Pascal not very orthogonal.
    Functions can’t return structured types.
    Type of formal parameter must be stated in
    function/procedure heading unless parameter is a function
    or procedure.
    Enumerated types can’t be read or written.
    etc...
  ✓ Non-orthogonality is often to simplify implementation.
  ✓ LISP is much more orthogonal than Pascal.

Language Evaluation Criteria

→ Control Statements/Constructs
  Importance and desirability of various control mechanisms varies
  with the language.

→ Data Types
  Rich set of data types makes programs much easier to write and
  understand. Provides abstraction.

→ Syntax
  Matters more than you think!
  Identifier length, reserved words, layout, etc.

→ Abstraction
  Must be able to hide details, or complexity is too great.
  process abstraction
data abstraction
Language Evaluation Criteria

- Reliability
  - Definition: performs to specifications under all conditions
  - Impact from:
    - Type checking (or lack thereof)
      - Compile-time: best
      - Runtime: good
    - Exception handling
      - Special language features to help intercept and handle unusual situations. No magic.
      - Somewhat controversial.
    - Aliasing (Y)
      - Two or more names for same memory cell.
      - PASCAL: var p,q: ^int;
      - begin
        - new(p);
        - q:=p;
      - end;
      - FORTRAN: Character *20 Last, First
      - Character *40 Name
      - Equivalence (Name (1:20), Last),
      - (Name (21:40), First)

Language Evaluation Criteria

- Cost
  - More than just runtime!
    - Time to train programmers
    - *Program development time
    - Compile time
    - Runtime
    - *Maintenance time (>50%)
  - *Functions of writability and readability
    => Most important