Chapter 1

Preliminaries

Topics

- Motivation
- Programming Domains
- Language Evaluation Criteria
- Language Design Trade-Offs
- Influences on Language Design
- Language Categories
- Implementation Methods

Languages

- The purpose of language is communication
  - Natural languages
  - Programming languages
- Writing an English essay:
  - Many can write in English
  - Few write well

What is a Programming Language?

- What is a language?
  - A set of rules that enables communication of ideas between people (between people and machines).
- What is a program?
  - A set of instructions intended for machine execution.
- What is a programming language?
  - A set of rules that define a set of legal programs.

I already know a Programming Language

- Why do I need to learn the concepts of programming languages?
  - I already know the latest/greatest/coolest programming language.
  - I can solve any problem using the programming language that I already know.

What is the best Programming Language?

- Java
- C
- C++
- Perl
- Python
- Visual Basic
- Lisp
- Pascal
- Prolog
- ML
- Modula-2
- Fortran
- Cobol
- Smalltalk
- Haskell
- Algol

No clear winner

- Obviously there is no “best” language for all situations.
  - Type of program
  - Time available
  - Cost
  - Size and scope of program
  - Programmer familiarity

Real computer programmers do not get religious about their languages, they get pragmatic and understand the trade-offs in their language choices.

Reasons to study concepts of PLs

1. Improves ability to express ideas
   - Languages influence the way you think and approach problems.
   - As you study new language features it may help you utilize or extend your own language skills.

   - New features can be later simulated in other language.

Reasons to study concepts of PLs

2. Improves background for choosing appropriate languages
   - Helps you understand the trade-offs in languages.
   - Provides alternatives choices that suits the project’s scope.

Reasons to study concepts of PLs

3. Increases ability to learn new languages
   - There is significant similarity in the constructs provided by languages so that learning a language is often just a matter of syntax.
     - Loops (while, for, do)
     - Selection (if, case)
     - Data types (int, char, string, object)
     - Jumps (goto, break, continue)

Reasons to study concepts of PLs

4. Allows a better understanding of the significance of implementation
   - Why the languages are design the way they are.

   - *Recursion vs. Iteration (faster)*

Reasons to study concepts of PLs

5. Increases ability to design new languages
6. Overall advancement of computing
   - Is the most popular programming language the best choice?
   - Why a language is the most popular?

What impacts Programming Language Design?

- Application domain
- Evaluation Criteria
- Computer architecture
- Programming methodologies

What impacts Programming Language Design?

- Application domain

The set of applications (algorithms) that we want to implement using our programming language

- Evaluation Criteria
- Computer architecture
- Programming methodologies

Programming Domains

- Scientific
  - Focus on calculations
    - Simple data structures
    - Large numbers of floating-point arithmetic computations
  - Primary concern: efficiency
  - Languages: Fortran, Algol 60

Programming Domains

- Business
  - Focus on reports and calculations
    - Facilitates the production of reports
    - Have precise ways of describing and storing decimal numbers and character data
    - Specifies decimal arithmetic operations
  - Language: Cobol

Programming Domains

- Artificial Intelligence
  - Use symbolic rather than numeric computations.
  - Focus on string and list manipulation
  - Languages: Lisp family (Common Lisp, Scheme, ML), Prolog.
- Systems Programming
  - Focus on fast execution
    - Need efficiency because of continuous use
  - Low-level features
  - Languages: PL/S (IBM), Extended Algol, C

Programming Domains

- Scripting Languages
  - Putting a list of commands (script) in a file to be executed.
    - Little code
  - Generally domain specific
  - Usually interpreted
  - Languages: sh and ksh (for shell), awk (report-generation), tcl and tk (X Windows), Perl (CGI programming), JavaScript, PHP

What impacts Programming Language Design?

- Application domain
- Evaluation Criteria

The set of factors that are important to the users of the programming language

- Computer architecture
- Programming methodologies

Project Manager's Dilemma

- Which language shall we use in the next project?
  - To come to a decision one needs arguments in favor or against a language.
- There are 4 main criteria:
  - Readability
  - Writability
  - Reliability
  - Cost

Are there other factors?

Language Evaluation Criteria

- Readability
  - The ease with which programs can be read and understood.
- Writability
  - The ease with which a language can be used to create programs.
- Reliability
  - Reliable performance (according to specifications) under all conditions.

Language Evaluation Criteria

Characteristics	Criteria: Readability	Writability	Reliability
Simplicity / orthogonality	♦	♦	♦
Control structure	♦	♦	♦
Data type and structures	♦	♦	♦
Syntax design	♦	♦	♦
Support for abstraction		♦	♦
Expressivity		♦	♦
Type checking			♦
Exception handling			♦
Restricted aliasing			♦

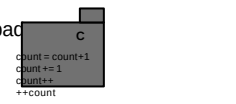
Evaluation Criteria: Readability

- Readability describes the ease of which programs can be read and understood.
- This is the most important criterion.
- It significantly affects the maintainability of code (mayor cost for programs).
- It must be considered in the context of problem domain.

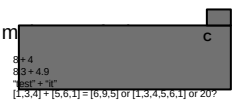
Evaluation Criteria: Readability

Factors

- Overall simplicity
  - Too many features is bad
  - Multiplicity of features is bad



  - Operator overloading m



Too simple can be just as much trouble (machine languages just 0 and 1)

Evaluation Criteria: Readability
Factors

- Orthogonality
  - A small number of primitive constructs combined in a relative small number of ways and everything can be combined with anything else.
    - Every possible combination is legal.
  - Meaning is context independent
    - Pointer is able to point to any type of variable.
  - Makes the language easy to learn and read.
  - Lack of orthogonality leads to exceptions to rules.

Evaluation Criteria: Readability
Factors

- Control statements
  - Program easier to read from top to bottom.
    - Unstructured: GOTO
    - Structured: loop

C
```
while( ( incr < 20) {
  while (sum <= 100) {
    sum += incr;
  }
  incr++;
}
```

ADA
```
loop 1:
  if ( incr >=20) go to out;
loop 2:
  if (sum > 100) go to next;
  go to loop2;
next:
  incr++;
  go to loop 1;
out:
```

- Data type and structures
  - Not enough structures leads to work-arounds which can reduce clarity.
    - Boolean type (vs. 0 and 1)
    - Record (vs. Arrays)

Evaluation Criteria: Readability
Factors

- Syntax considerations
  - Identifier length and form
    - Too short equals bad variable names.
  - Special words
    - Block structure
      - end vs. end-if and end-loop
    - Special words
  - Form and meaning
    - Semantics should follows directly from syntax.

C
```
for (i=1; i<10; i++)
{
  if (a)
  {
  }
}
```

```
for I in 1..10 loop
  if a then
  end if;
end loop;
```