

Programming Languages Overview

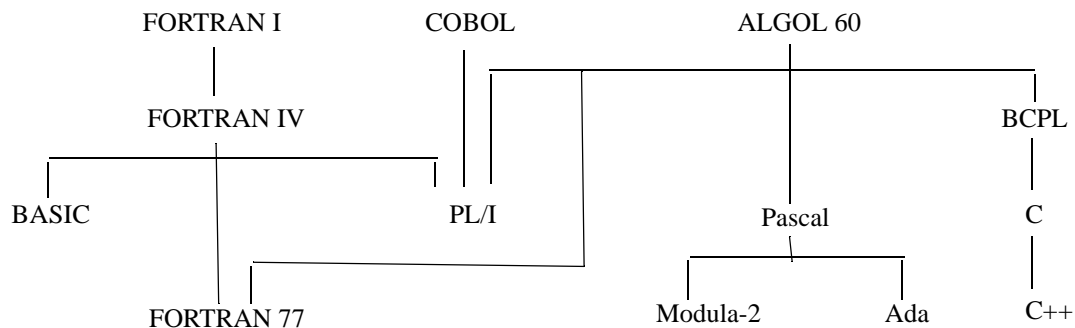
Overview of the different types of PL's & their historical relationship. (taken from Wilson, Leslie B., and Clark, Robert G. Comparative Programming Languages, Addison Wesley, 1993.

I. Different Types of PL's

- A. Imperative languages: FORTRAN, ALGOL, COBOL, C, Pascal, Ada, Modula-2
Program achieves its effect by changing the value of variables through assignment stmnts.
- B. Functional languages: LISP, ML, LOGO
State *what* to do, rather than *how* to do it. Pure Functional Language has no assignment.
- C. Logic languages: PROLOG
Goal oriented, *facts* and *rules* used with *questions*.
- D. Object-oriented languages: Smalltalk, Eiffel (C++ is a mixture between A and D)
Characterized by Inheritance, Overloading, Dynamic Binding, Message Passing

II. Imperative Languages

A. Historical Development



B. Select Languages:

1. FORTRAN

~1958, allowed high-level, efficiency was primary concern, used for scientific programming. Implemented comments, assignments w/complex r.h.s., DO loops, formatted I/O [See FORTRAN prog.]

2. ALGOL

1960, Designed as common language by European & American programming experts, syntax described using BNF, introduced compound stmnt. (block) for structure, variable bounded arrays, recursion [See Algol prog.]

3. COBOL

late 50's, COMmon Business Oriented Language, pushed by DoD, I/O oriented, divided into 4 parts: 1. Identification division (documentation), 2. Environment division (connection to external files), 3. Data division (logical desc. of data), 4. Procedure division (algorithms).

4. PL/I

early 60's, IBM wanted language to unify FORTRAN & COBOL users, featured *defaults*

5. BASIC

mid 60's, Beginner's All-purpose Symbolic Instruction Code, simplicity, interpretive interactive environment [See BASIC prog.]

6. Pascal

early 70's, designed for teaching programming, featured data-types

7. C & C++

late 70's, combines high-level lang. w/facilities & efficiency of assembly lang., lack of type checking, operations on addresses and bit patterns. C++ has more strict type-checking, has classes & inheritance. [See C prog]

8. Ada

mid 70's, DoD wanted standard lang. for real-time concurrent embedded systems. Has *packages*(similar to classes), libraries, is strongly typed, *exit when* from loops, task facilities for concurrency, exception handling

III. Functional languages: LISP, ML, LOGO

Note that there is no assignment statement. See example program to find the difference between the largest and smallest of three integer numbers using a functional subset of Ada. [See Ada prog.]

IV. Logic languages: PROLOG

Goal oriented, *facts* and *rules* used with *questions*. [See PROLOG prog.]

V. Object-oriented languages

We will spend most of this course dealing with this