Language Evolution

c. 1950 machine pseudocodes

1956 FORTRAN I (no data types)
- compiled
- i/o formatting
- subroutines
- IF, DO (no nesting)

1958 LISP
- uniform data structure (lists)
- functional programming (a formal model of computation)

1960 ALGOL 60
- data typing
- block structure
- pass by value, pass by name
- recursion

1960 COBOL
- macros
- hierarchical data structures
- long names

1964 BASIC
- remote terminal access
- easy to use

1965 PL/I (ALGOL+FORTRAN+COBOL)
- general purpose
- concurrency
- runtime error handling
- pointers
- sub-array reference

1967 SIMULA 67
- co-routines
- classes, data abstraction

1968 ALGOL 68
- orthogonality (few constructs and combinations)
- user defined data types
- dynamic arrays

1971 PASCAL
- teaching language (simple, expressive)

1972 C

- rich operator set
- OS-based (Unix)

1972 PROLOG (inefficient, few applications)

- declarative
- formal model using logic

1980 Smalltalk

- pure object-orientation
- methods
- messages, send
- software development environment (windowing)

1985 Ada

- committee design (too large and complex)
- encapsulation
- exception handling
- generic procedures
- concurrent tasks and synchronization

1985 C++

- predefined classes
- overloading
- templating (parameterized classes)

1988 Mathematica

- very high level mathematical language
- string rewrite engine, string uniform data structure
- all programming styles
- highly integrated development environment

1993 Java

- reference types (no pointers)
- Boolean control (no control arithmetic)
- pure methods, applets (no functions or subprograms)
- threads
- garbage collection
- limited coercions