

### Zuse's Plankalk

Program calculus had advanced features & Data structures arrays and records never implemented

### Pseudo Codes

Short code developed by John Mauchly 1949. Mathematical expressions were coded. Interpreted 50x slower than machine code

### Speedcoding

IBM 701 John Backus 1954 conditional & unconditional branching automatic incrementing of address register interpreted

### UNIVAL

Primitive compiler Grace Hopper Pseudocode expanded into machine code

### Fortran 77: 1978

Character string logical loop control statement If-Then-Else statement

### Fortran 95 & 2003 & 2008

Relatively minor changes

### ALGOL 58

Data types formalized Names could be any length

Arrays could have any number of subscripts Parameters were separated by mode(in & out)

[] for subscripts compound statements(begin ...end)

No I/O-would make it machine dependent if had an else-if clause

:= assignment ; as separation

### COBOL(common business oriented language)

specifically designed for business. based on Flow-matic.

### Dennis Ritchie & Brian Kernighan

wrote the 1st c book; can be used for most applications poor type checking; it was free. used to write UNIX operating systems

### PHP

Hypertext Preprocessor, Rasmus Lerdorf service side HTML

### Fortran(Formula Translator)

-Fortran 0: 1954 not implemented -Fortran 1: 1957

designed for the IBM 704 index registers & floating point hardware led to compiled programs -Environment Factors computer memory was small & unreliable

created for scientific applications no programming methodology or tools

hardware was expensive -Other Features no data types i,j,k,l,m,n were implicitly integers.

### Fortran II 1958

Fixed problems with Fortran I independent compilation of subroutines

### Fortran IV 1960-1962

Explicit type declarations Logical selection statement(if) subprogram names could be parameters (subprogram = functions) ANSI standard in 1966

### Fortran 90

modules dynamic arrays

pointers recursion

CASE statement parameter type checking

### ALGOL 60

BNF(Backus-Naur Form) 6 Days

block structure was introduced pass by value & pass by name

recursive procedures dynamic arrays

### ALGOL 60: Success/Failures

-Success standard way to publish algorithms for over 20yrs. all subsequent imperative lang. are based on ALGOL(java,c,c++... first machine independent language. first language whose syntax was formally defined(BNF)

-Failures never a commercial success not widely used in the U.S. lack of support from IBM. Lack of I/O. too flexible, hard to implement. popularity of Fortran.

### Scripted Lang.

-Perl(designed by Larry Wall 1987)  
wide spread use on the web  
used as a replacement for UNIX admin. lang.  
-JavaScript(began at Netscape)  
used to create dynamic HTML documents.  
pure interpretation;client side HTML  
related to Java only by syntax

### LISP

-List Processing  
designed at MIT by McCarthy  
-AI Research  
data is processed in Lists (rather than arrays)  
symbolic computation (rather than numeric)  
-2 data types: atoms & lists  
-syntax based on lambda calculus

### LISP Evaluation

-Pioneered functional programming  
no variables or assignment  
program controlled with application of functions: recursion, conditional expressions.

### Related to LISP

scheme developed at MIT  
ML  
common LISP(large & complex)  
Haskell & F#

### ALGOL

The goal was to create a universal language.  
Languages were machine dependent.  
GAMM 1955(fear of being dominated by america)  
attempted joint design process with ACM.

### Goals

Syntax close to math.  
Possible to use the language to describe algorithms.  
Must be mechanically translatable into machine code.

### Flow-Matic

Names up to 12 characters.  
English words for operators.  
Data & code was separate.

### Flow-Matic (cont)

First word of every statement was a verb.

### COBOL Design process

look like english.  
easy to use(even if less powerful)  
broader base of users.  
address current compiler problems.  
committee from manufacturers & D.o.D  
design problems: arithmetic expressions, subscripts

### COBOL contributions

1st macro facility in a high level language.  
records  
nested selection statements.  
long names(from flow-matic)  
still most widely used language for businesses  
mandated by D.o.D  
separate data division  
strong on reports

### C++(descendant of ALGOL)

Bell Labs by Stroustrup 1980  
Large & complex; Exception handling  
ANSI standard Nov. 1997;  
Microsoft vers. MC++  
Evolved from C & SIMULA 67

### Basic(Beginner's All-purpose Symbolic Inst. Code)

teaching language; no respect;  
easy to learn  
included on first PCs; small implementation  
Design to teach students their 1st. language.  
Mather & Waite 1971

### Basic design goal

Easy for non-science students  
must be pleasant & friendly.  
quick turn around on homework  
user time more important than computer time.  
free and private access.



### PL/I

cross between fortran & cobol  
teaching/never became a  
commercial success  
IBM design  
use for scientific computing and  
business computing

### Problem-PL/I

Tried to be everything for  
everybody.  
Original name NPL(New  
programming lang.)  
Too many features/constructs.

### Dynamic Languages

APL(a programm ing lang.):	SNOBOL: designed for string manipulative @ Bell labs 1964
Hardware description lang.	still used for test processing
difficult to read	Powerful for string pattern matching
1960s Ken Iverson	

### Dynamic Languages (cont)

Many operators  
Minimal impact on current  
languages.  
-SIMULA 67 extension of ALGOL

### Descendant of Algol

### Pascal 1971

Niklus Worth(Vert); small, simple  
designed for teaching; nothing new  
largest impact was on teaching.  
Mid 70s & late 90s students  
learned PASCAL

### C Programming

Bell Labs;Dennis Ritchie  
Early ancestors developed at  
Cambridge  
Designed for systems program  
1st high & low level lang.

### ADA-D.o.D

Ada Lovelace(world's first computer  
programmer)  
Mandated by the D.o.D

