

Lecture 07

ALGOL 86 to Prolog

Mr. Mubashir Ali

Lecturer (Dept. of Computer Science)

dr.mubashirali1@gmail.com

Summary of Previous Lecture

- 1. Zuse's Plankalkul – 1945**
- 2. Assemblers and Assembly Languages**
- 3. FORTRAN I**
- 4. FORTRAN IV**
- 5. FORTRAN 77 & FORTRAN 90**
- 6. LISP**
- 7. ALGOL 58**
- 8. ALGOL 60**

Outline

- 1. ALGOL 68**
- 2. COBOL 60**
- 3. PL/1**
- 4. BASIC**
- 5. Early Dynamic Languages**
- 6. Simula 67**
- 7. Pascal**
- 8. C**
- 9. Prolog**

Algol 68 - 1968

- **Continued development of ALGOL 60 was released as Algol 68 in 1968, but it is not a superset of that language**
- **Design is based on the concept of orthogonality**
- **Contributions:**
 - **User-defined data structures**
 - **Reference types**
 - **Dynamic arrays (called flex arrays)**
 - **Had even less usage than ALGOL 60**
 - **Had strong influence on subsequent languages, especially Pascal, C, and Ada**

Cobol - 1960

- **Design goals:**
 - **Business-oriented computation – fixed point arithmetic**
 - **Must look like simple English**
 - **Must be easy to use, even if that means it will be less powerful**
 - **Must broaden the base of computer users**
 - **Must not be biased by current compiler**
- **Problems**
 - **Design committee were all from computer manufacturers and DoD branches**
 - **Design Problems:**
 - **Arithmetic expressions?**
 - **Subscripts?**
 - **Fights among manufacturers**

Cobol (continued)

- **Contributions:**
 - **Gave the first macro facility in a high-level language**
 - **Had hierarchical data structures (records)**
 - **Nested selection statements**
 - **Long names (up to 30 characters), with hyphens**
 - **Data Division**
 - **Fixed-point arithmetic**
- **Comments:**
 - **First language required by DoD; probably would have failed without its support**
 - **Still a very widely used business applications language**
 - **Very popular in business and government, much less at universities.**

PL/1 – 1965

Everything for Everybody

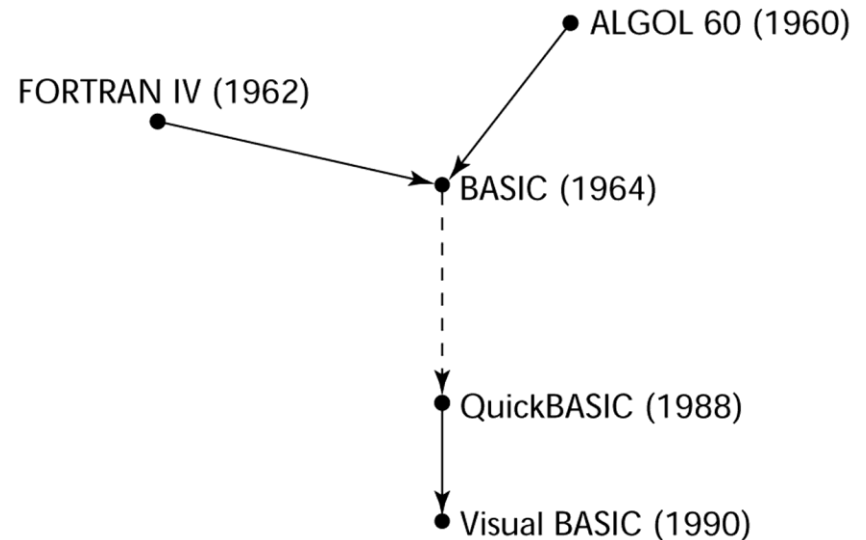
- **Computing situation in 1964 (IBM's point of view)**
 - **Scientific computing**
 - IBM 1620 and 7090 computers
 - FORTRAN
 - SHARE user group
 - **Business computing**
 - IBM 1401, 7080 computers
 - COBOL
 - GUIDE user group
- **User Community in 1963**
 - Scientific users began to need more elaborate I/O, like COBOL had
 - Business users began to need fl. pt. and arrays (MIS applications)
 - It looked like many shops would begin to need two kinds of computers, languages, and support staff- this solution was too costly
 - **The obvious solution:**
 - Build a new computer to do both kinds of applications
 - Design a new language to build both kinds of applications

PL/1 (Continued)

- **PL/1 contributions:**
 - **First language that gave unit-level concurrency**
 - **First language to provide exception handling**
 - **First language to provide pointer data type**
 - **First language to provide array cross sections**
- **Comments:**
 - **Many new features were poorly designed**
 - **Too large and too complex**
 - **Anyway, it was actually used for both scientific and business applications**

Basic - 1964

- **Designed by Kurtz & Kemeny at Dartmouth**
 - **Design Goals:**
 - **Easy to learn and use for non-science students**
 - **Must be "pleasant and friendly"**
 - **Fast turnaround for homework**
 - **Freely available for use**
 - **User time is more important than computer time**

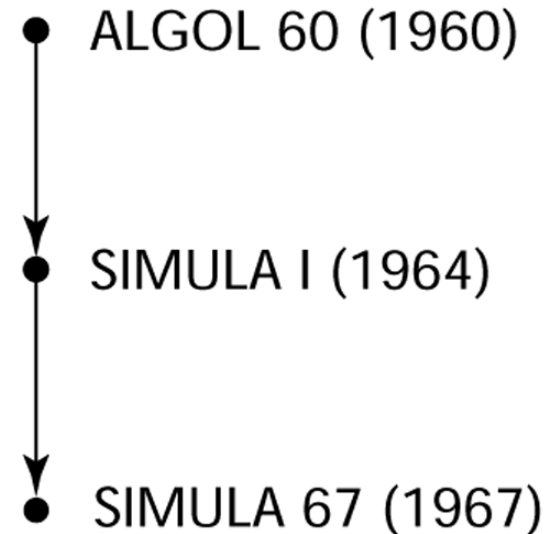


Early Dynamic Languages

- **Characterized by dynamic typing and dynamic storage allocation**
- **APL (A Programming Language) 1962**
 - **Designed as a hardware description language (at IBM by Ken Iverson)**
 - **Highly expressive (had many operators for both scalars and arrays of various dimensions)**
 - **Programs are very difficult to read - commonly known as “write-only” language**
- **SNOBOL (1964)**
 - **Designed as a string manipulation language (at Bell Labs by Farber, Griswold, and Polensky)**
 - **Powerful operators for string pattern matching**

Simula 67 - 1967

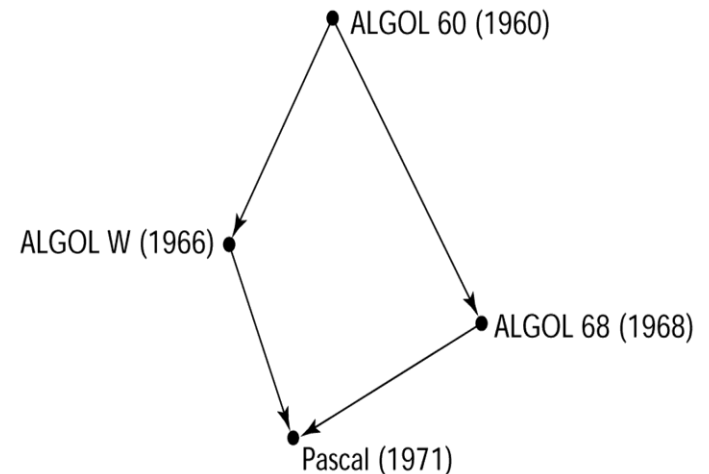
- **Designed primarily for system simulation (in Norway by Nygaard and Dahl)**
- **Based on ALGOL 60 and SIMULA I**
- **Primary Contributions:**
 - **Coroutines - a kind of subprogram - Implemented in a structure called a class**
 - **Classes are the basis for data abstraction**
 - **Classes are structures that include both local data and functionality**



Pascal – 1971

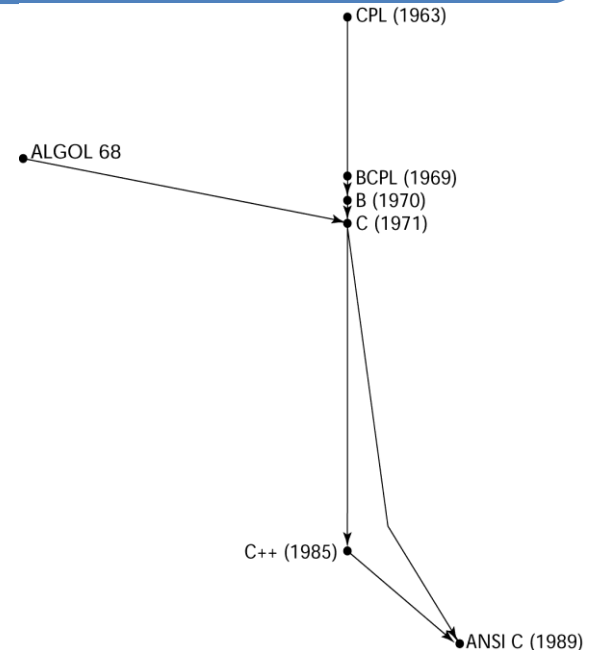
Simplicity by Design

- **Designed by Niklaus Wirth, who quit the ALGOL 68 committee because he didn't like the direction of that work**
- **Designed for teaching structured programming**
- **It was simple, small, nothing really new**
- **Most widely used language for teaching programming in colleges for many years**



C - 1972

- **Designed for systems programming (at Bell Labs by Dennis Richie)**
- **Evolved primarily from B, but also ALGOL 68**
- **Had powerful set of operators, but poor type checking**
- **Initially spread through UNIX**
- **It redefined portability**



Prolog - 1972

- **Developed at the University of Aix-Marseille, by Comerauer and Roussel, with some help from Kowalski at the University of Edinburgh**
- **Based on formal logic**
- **Non-procedural - Declarative, non-deterministic (builtin backtracking)**
- **Associative memory, pattern directed procedure invocation.**

Summary

- ✓ **ALGOL 68**
- ✓ **COBOL 60**
- ✓ **PL/1**
- ✓ **BASIC**
- ✓ **Early Dynamic Languages**
- ✓ **Simula 67**
- ✓ **Pascal**
- ✓ **C**
- ✓ **Prolog**