Programming Language

History

MESSI NGUELE Thomas (PhD)

Yaoundé, March 13th 2025

Reasons to study programming language history

1. Langauges available today are only explicable by examining how they grew up
2. Understand why some languages are still used even if they are very old.
3. Pinpoint some of the errors made in the past and try to avoid repeating them.
4. Try to recapture the feelings of programmers at that time when each new language arrived on the scene.

Agenda

Agenda

Languages Classification

Generations of Languages

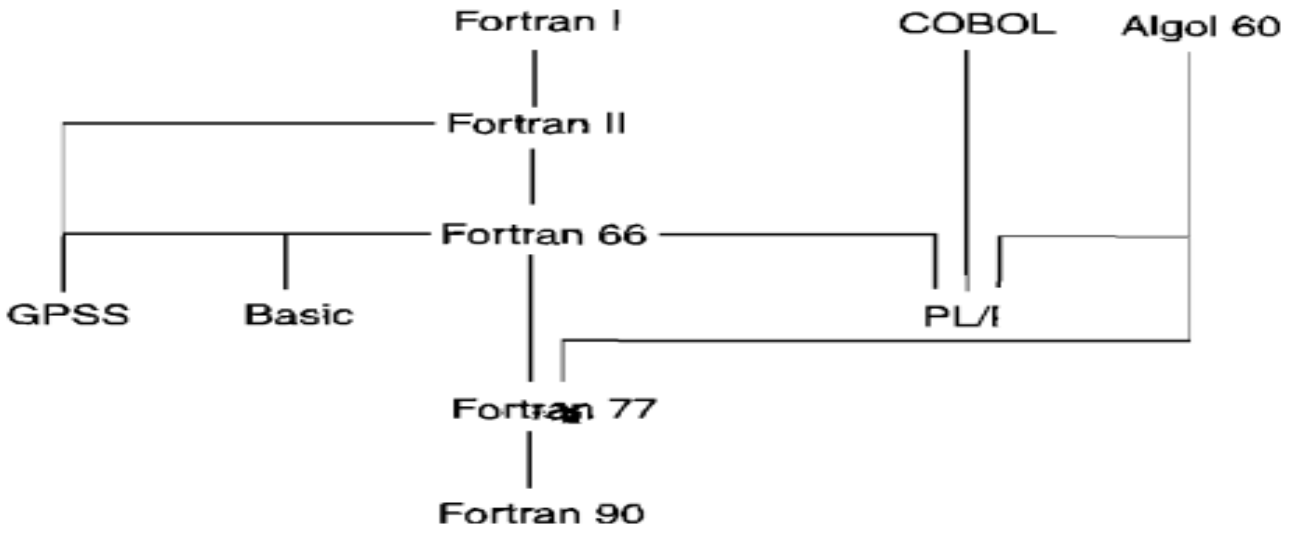
1. First generation : Machine Codes
2. Second generation : Autocodes and symbolic assemblers
3. Third generation : Langages are independent from particular computer hardware
4. Fourth generation : result of the dissatisfaction of business users with large conventional languages like COBOL.

DSLs vs GPLs

1. GPLs : General Purpose Languages. They were designed for developing any kind of applications.
2. DSLs : Domain Specific Languages. Each DSL is designed for a particular domain.
    1. Embedded DSLs (like library on GPLs)
    2. Stand alone DSLs (with their own language infrastructure).

Agenda

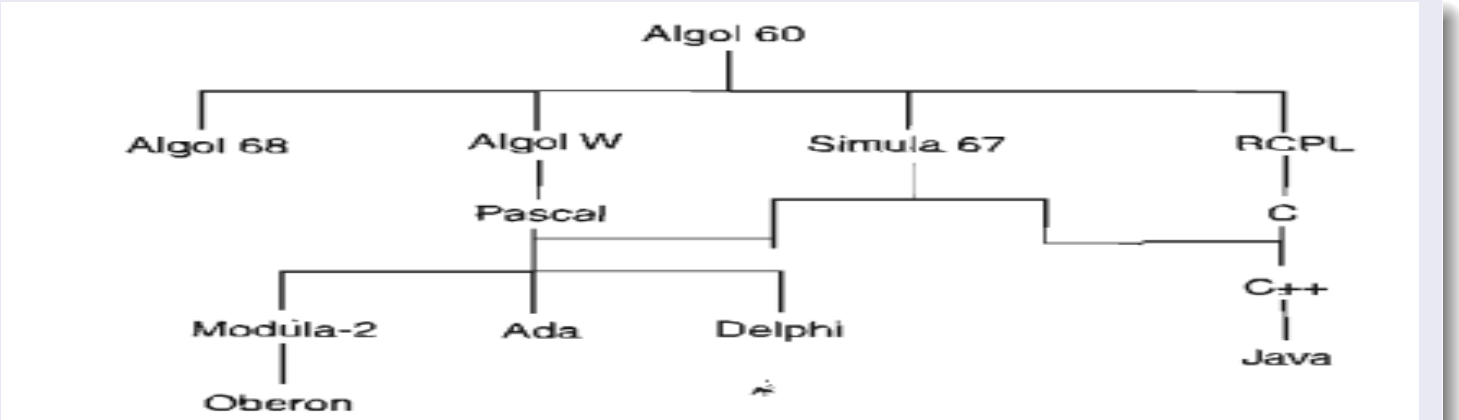
Fortran



1. IBM Mathematical **For**mula **Tran**slating System
2. 1956 (definition) – 1958 (implementation, compiler)
3. Third Generation ?
4. DSL ? (embedded or stand alone)

Agenda

Algol



1. Close as possible to standard mathematical notation
2. Be readable without too much additional explanation
3. Used for computating processes description in publications

- Was not able to supersede Fortran
  - It came 3 years after fortran (programmers were reluctant to change)
  - It has more features and was harder to learn
  - Fortran compilers were simpler and produced more efficient code

Agenda

Exercises

5 - According to language design, how can you compare the following programming languages ? C, C++, Java, Python, Pascal, Prolog, Mercury, GO, Haskel, Fortran, Algol, COBOL, ...

6 - Make history of previous languages.

7 - Make groups of student to study languages.

Bibliography

- WILSON, Leslie B. et CLARK, Robert George. Comparative programming languages. Pearson Education, 2001.
- MACLENNAN, Bruce J. Principles of programming languages : design, evaluation, and implementation. Holt, Rinehart & Winston, 1986.


Thank you for your attention!