

Typology of Programming Languages

Introduction

May 2025

Practical information

- Teachers:
 - ▶ Paris: mael.cravero@epita.fr
 - ▶ Lyon: nicolas.schabanel@epita.fr
 - ▶ Rennes: gaetan.staquet@epita.fr
 - ▶ Strasbourg: paul.hervot@epita.fr
 - ▶ Toulouse: mathias.choquet@epita.fr
- Resources:
 - ▶ [Moodle](#)
 - ▶ [Slides](#)
- Questions:
 - ▶ [Newsgroup](#)

Goals

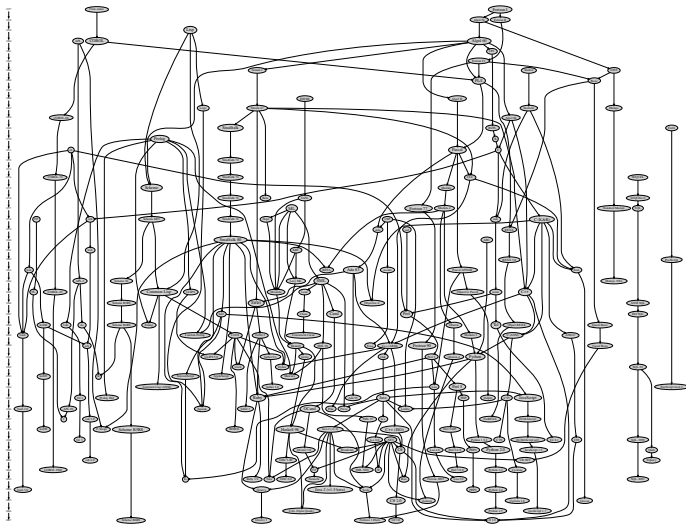
Programming languages are the tools of our trade.

- What are they? Where are they coming from?
- How can we compare and classify them?
- How can we work around their flaws?
- How can we learn them more efficiently?

Topics covered

| C | FOR COMMENT | CONTINUATION | FORTRAN STATEMENT | | | | IDENTI- FICATION | |
|---|----------------|--------------|---------------------|---|--|----|---------------------|----|
| | | | STATEMENT NUMBER | | | | | |
| 1 | 5 | 6 | 7 | | | 72 | 73 | 80 |
| C | | | | PROGRAM FOR FINDING THE LARGEST VALUE | | | | |
| C | | X | | ATTAINED BY A SET OF NUMBERS | | | | |
| | | | | DIMENSION A(999) | | | | |
| | | | | FREQUENCY 30(2,1,10), 5(100) | | | | |
| | | | | READ 1, N, (A(I), I = 1,N) | | | | |
| | 1 | | | FORMAT (I3/(12F6.2)) | | | | |
| | | | | BIGA = A(1) | | | | |
| | 5 | | | DO 20 I = 2,N | | | | |
| | 30 | | | IF (BIGA-A(I)) 10,20,20 | | | | |
| | 10 | | | BIGA = A(I) | | | | |
| | 20 | | | CONTINUE | | | | |
| | | | | PRINT 2, N, BIGA | | | | |
| | 2 | | | FORMAT (22H1THE LARGEST OF THESE 13, 12H NUMBERS IS F7.2) | | | | |
| | | | | STOP 7777 | | | | |

Topics covered



Programming languages genealogical tree (from rigaux.org)

The course

4 classes:

- ① Programming languages and computers
- ② Subroutines, memory safety and design by contract.
- ③ Genericity and metaprogramming.
- ④ Advanced topics

This will be completed by 2 practicals and some homework.

Evaluation

- Exercises (40%)
 - ▶ Small exercises in various programming languages.
- Practicals (40%)
 - ▶ More involved exercises in programming languages you may not know.
- Final exam (20%)
 - ▶ Programming languages history and typology questions.
 - ▶ Which language invented *feature*?
 - ▶ Who created *language*?
 - ▶ How does *language feature* work?
 - ▶ Open question comparing two languages in a given context.

Note

You are not expected to know dates by heart but you should be able to place events on a timeline.