

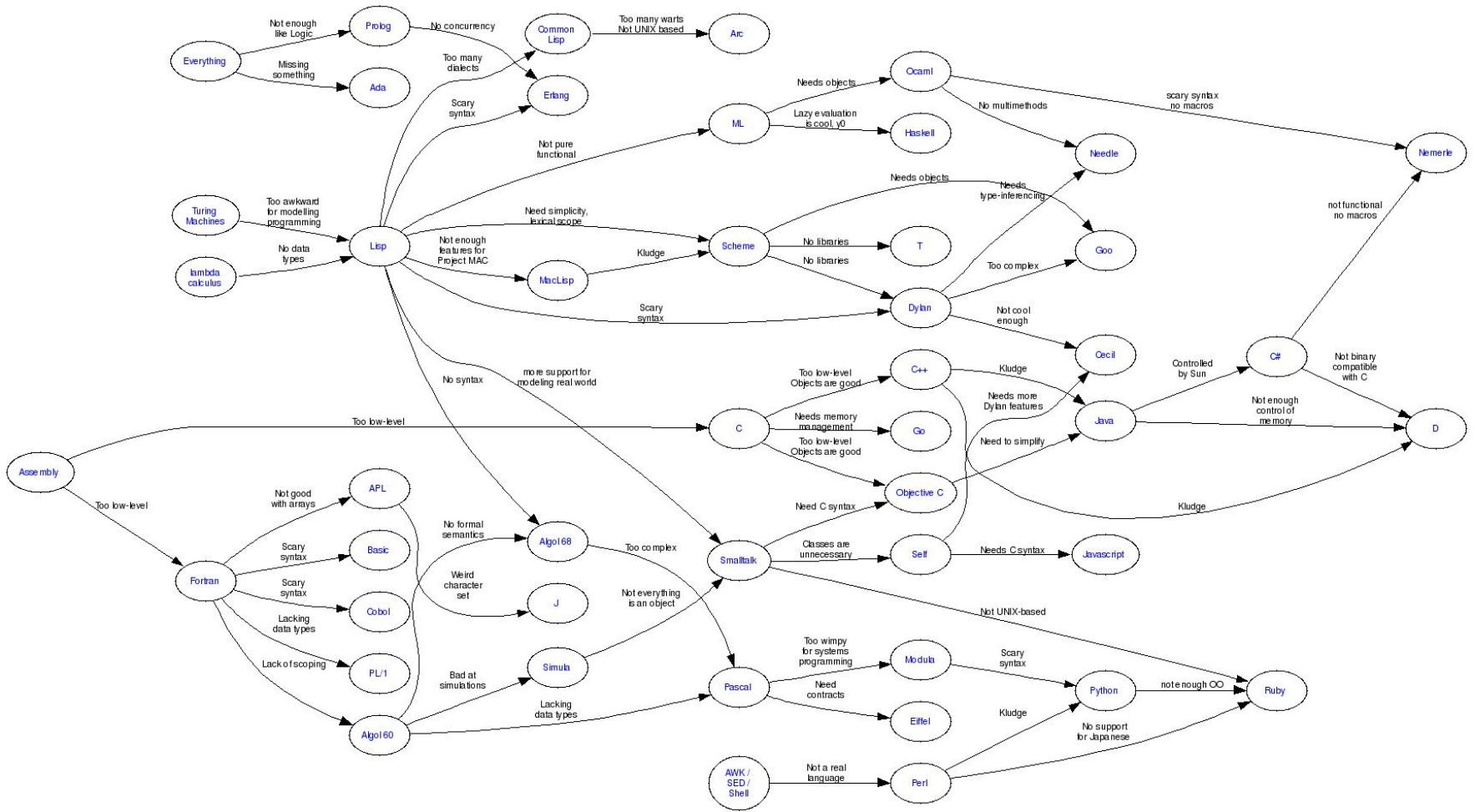
A brief history of programming languages

Pioneers of programming



- Charles Babbage 1837
 - Invented the Analytical Engine
 - For those interested in very beginning see https://www.youtube.com/watch?v=wOQuW6QFdos&ab_channel=RiceCompSci
- Konrad Zuse
 - Plankalkül (plan calculus) 1942-1945
 - notation (never implemented)
- Corrado Böhm (1951)
 - First language with a compiler

And Then ...



Generations

There are 4 "generations" of programming languages

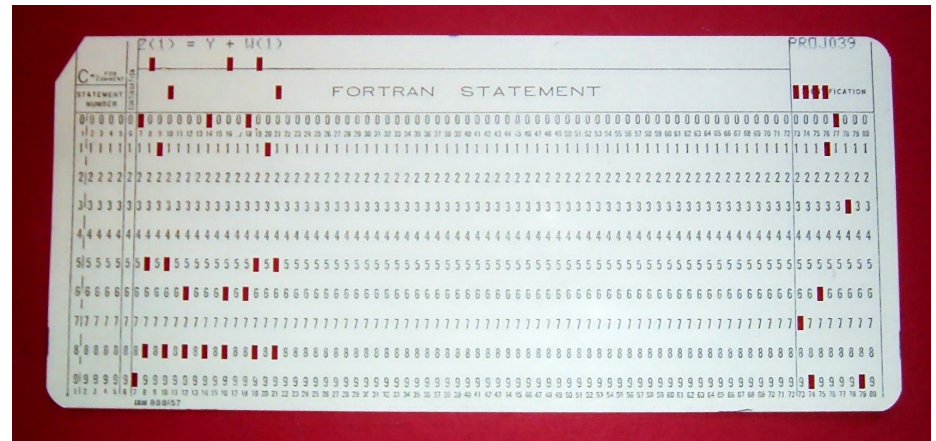
1. machine code
2. assembly
3. high level languages (Fortran, C, Java, ...)
4. ad-hoc/visual (Visual Basic, SQL, ...)

Programming Styles

- Imperative (+ object oriented)
- Functional
- Logic / Declarative
- Functional logic

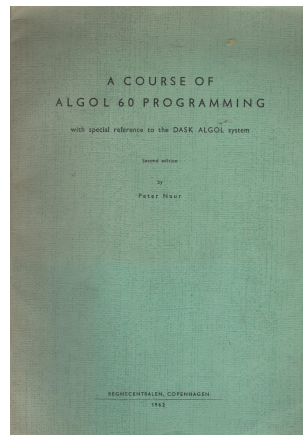
Fortran

- First effectively implemented high-level language by John W. Backus at IBM (1956)
- Introduced variables, loops, procedures, statement labels and much more
- Earliest versions of Fortran had many unique features, often awkward, later kept along for compatibility
- Still widely used in engineering applications that require much array manipulation
- Newest version
 - Fortran 2018



Algol 60

- Design in 1960 by a group of people including John McCarthy, Alan J. Perlis, Peter Naur, ...
- First to have block structure, recursion, and a *formal definition*
- Not used now, but it is the ancestor of most contemporary languages
- As far as design goes, Algol 60 was without doubt the most important innovation in the history of programming languages
- American VS European battle



Cobol

- Created by a consortium (1958)
- Business-oriented computations
 - very strict program organization
 - poor control structures
 - elaborate data structures, record type introduced for the first time
- Used to be very popular in business and government, much less at universities
- Still used → legacy applications



Basic

- The first in history language of personal computing (1964)
- Designed to be easy to learn (Beginner's All-purpose Symbolic Instruction Code)
- Very simple, limited, though still general-purpose
- Present-day versions of Basic are full-fledged languages—not "basic", and not easy to learn any more.

```
READY
10 FOR X=1 TO 10
20 PRINT "HOLA WIKIPEDIA"
30 NEXT X
RUN
HOLA WIKIPEDIA
HOLA WIKIPEDIA
HOLA WIKIPEDIA
HOLA WIKIPEDIA
HOLA WIKIPEDIA
HOLA WIKIPEDIA
HOLA WIKIPEDIA
HOLA WIKIPEDIA
HOLA WIKIPEDIA
HOLA WIKIPEDIA
READY
■
```

Simula 67

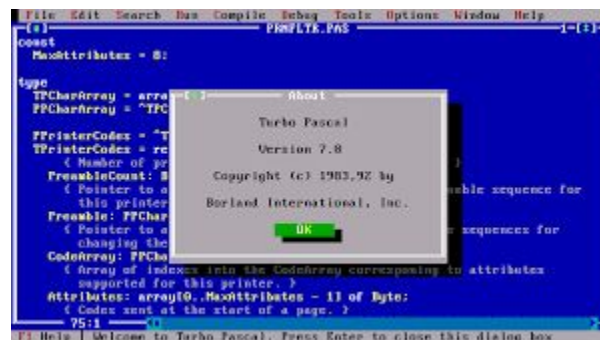
- An extension of Algol 60 designed for simulation of concurrent processes
- Introduced the central concepts of **object orientation**: classes and encapsulation
- Predecessor of Smalltalk and C++
- Now unused



Dahl and Nygaard at the time of Simula's development

Pascal

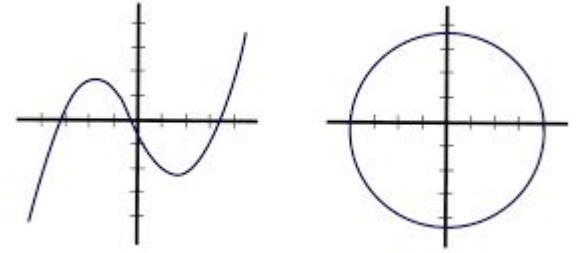
- A conceptually simplified and cleaned-up successor (1970) of Algol 60
- A good language for teaching structured programming
- Its later extensions (for example, Delphi) are full-fledged systems programming packages



C

- The implementation language of Unix (1972)
- A great tool for systems programming and a software development language on personal computers
- Once fashionable, still in use, but usually superseded by C++
- Dangerous if not used properly: segmentation faults!
- Relatively low-level

Lisp



- One of the earliest programming languages (John McCarthy - 1958)
- Based on the concept of computing by evaluating **functions**. Very good for symbolic computing
- For years, the only language for Artificial Intelligence (Prolog is 12 years younger)
- Many dialects, two standards (Scheme, Common Lisp)
- Lisp's successors are very elegant (Miranda, ML, **Haskell**)

Prolog

- A very high-level programming language (1972)
- Declarative, based on a subset of **logic**, with proofs interpreted as computation
- Very powerful:
 - Non-deterministic (built-in backtracking)
 - Elaborate, flexible pattern matching
 - Associative memory
- In skilled hands, it is a very strong tool

P	Q	$P \Rightarrow Q$
T	T	T
T	F	F
F	T	T
F	F	T

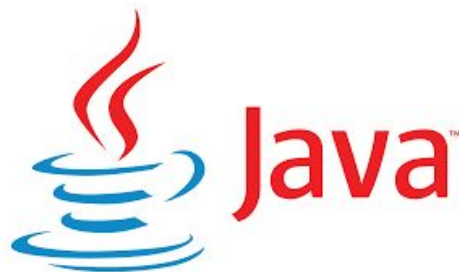
C++

- An *object-oriented* extension of the *imperative* language C, designed in 1980
- This is a hybrid design, with object orientation added to a completely different base language



Java

- Sized-down reworking of C++ designed in 1995
- Full object orientation
- Designed for Internet programming, but general-purpose
- Very used (in the top 5 of the most used languages)
- **To not be confused with JavaScript!**



Scripting languages

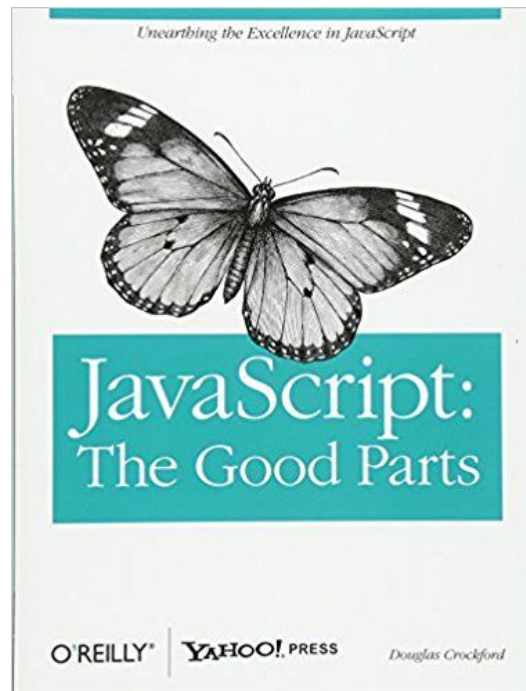
- Text processing:
 - Perl (1987)
 - **Python** (1991)
- Web programming
 - **JavaScript** (1995)
 - PHP (1995)

"New" ones

- Golang (channels for concurrency)
- Kotlin (type inference)
- Rust (ownership system)
- Typescript (optional static typing)

Bad things to avoid

- <https://www.destroyallsoftware.com/talks/wat>
from minute 1.22



Or Fun Things To Do :)



IT'S SHOWTIME

TALK TO THE HAND "hello world"

YOU HAVE BEEN TERMINATED

False I LIED

True NO PROBLEMO

If BECAUSE I'M GOING TO SAY PLEASE

Else BULLSHIT

EndIf YOU HAVE NO RESPECT FOR LOGIC

While STICK AROUND

EndWhile CHILL

Homeworks

- Read chapter 13 of the Programming Language book