A (rather personal) history of languages and compilers

Talen & Compilers

Today

No lecture notes other than these slides

Learning goals

- Explain the role of languages and compilers in the history of computing (from a biased viewpoint)

What *is* a program I

[1]**Program** (noun) - a plan for the solution of a problem. A complete program includes plans for the transcription of data, coding for the computer and plans for the absorption of the results into the system. The list of coded instructions is called a **routine**.

[1] Grace Hopper (1954). First glossary of programming terminology of the ACM.
Hopper should have received a Turing award

Utrecht University

What *is* a program II

[1]Sequences of statements and declarations, when appropriately combined, are called **programs**

[1] Alan J. Perlis and Klaus Samelson (1958). Preliminary report - International algebraic language. Commun. ACM 1, 12 (Algol58)
Perlis was the first Turing award winner

Utrecht University

What *is* a program III

[1]An expression of a computational method in a computer language is called a **program**

[1] Donald E. Knuth (1968). The Art of Computer Programming (1st Edition). Vol. 1. Addison-Wesley.
Knuth was the youngest Turing award winner

What *is* a program IV

[1]A **program** is an abstract symbol manipulator, which can be turned into a concrete one by supplying a computer to it

[1]Edsger W. Dijkstra. 1988. On the cruelty of really teaching computer science. Personal communication. https://www.cs.utexas.edu/users/EWD/ewd10xx/EWD1036.PDF.
Dijkstra is the only Dutch Turing award winner

Utrecht University

When did people start developing programs?

Length and width is to be equal to the area.
You should proceed as follows
Make two copies of one parameter
Subtract 1.
Form the reciprocal.
Multiply by the parameter you copied.
This gives the width,

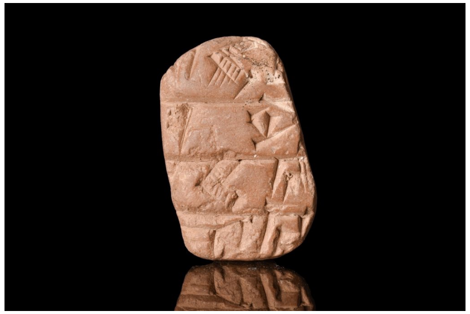
$x+y = xy$

$x=xy-y$

$x=(x-1)y$

$x(x-1)^{-1}=y$

Old Babylonia, 1900 - 1600 BCE, the Louvre



[1] Donald Knuth (1972). Ancient Babylonian Algorithms. CACM 15.

The ancient Greek

Many mathematical algorithms:

Euclides greatest common divisor
Sieve of Erathosthenes
Archimedes’ method of exhaustion
…

Programming languages[1]

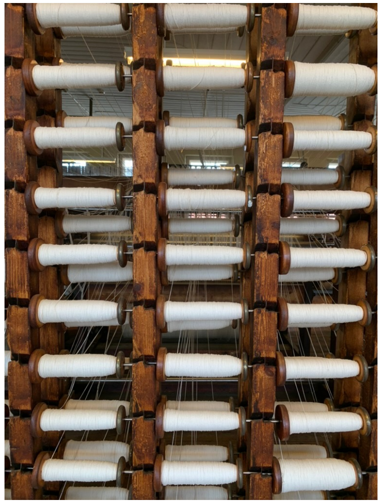
A programming language is a systematic notation by which we describe computational processes to others.

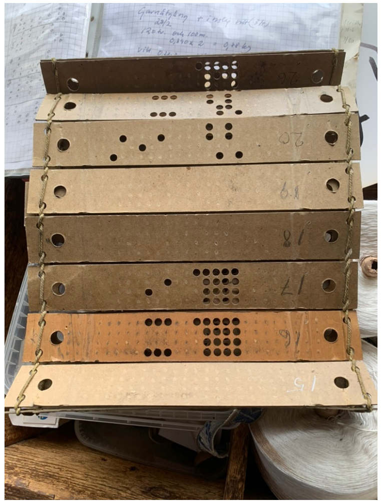
A computational process is a set of steps which a machine can perform for solving a task.

[1] Ellis Horowitz (2012). Fundamentals of programming languages. Springer

Utrecht University

Machines and programs I

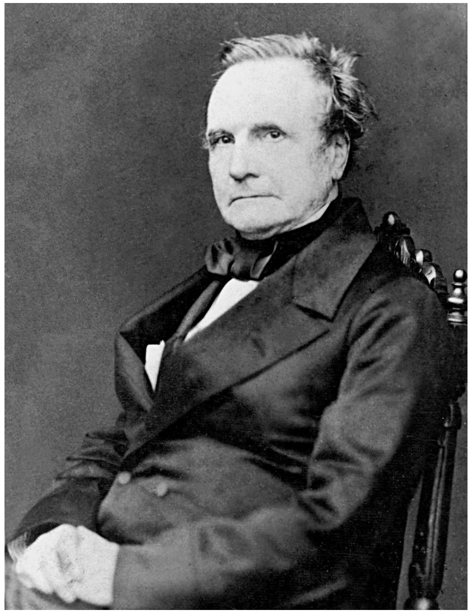




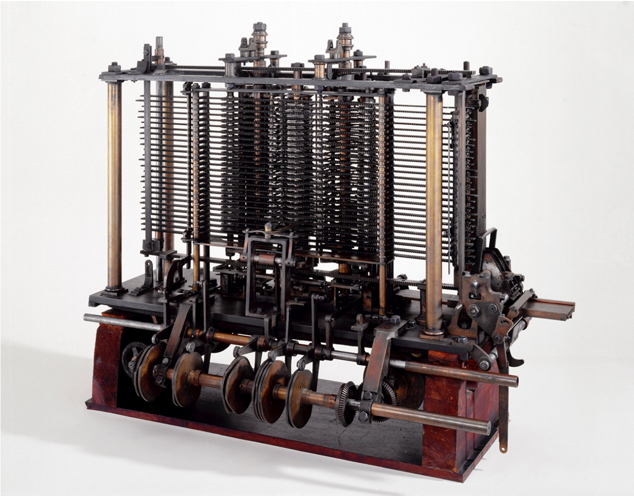
Jacquard

Utrecht University

Machines and programs II



Charles Babbage



The Analytical Engine

Ada Lovelace

Many real machines…

Z3 – Konrad Zuse (beginning 2nd world war)

ENIAC (Electronic Numerical Integrator and Computer) – Mauchly and Eckert (just after)

ARRA (Automatische Relais Rekenmachine Amsterdam) – Scholten and Loopstra (CWI Amsterdam, 1952)

ZEBRA (Zeer Eenvoudige Binaire Reken Automaat) – Willem van der Poel (PTT, 1958)

X1 – CWI (mainly; then a company; sold to Philips)

Compilers

The first compilers were paper-software:

- Zuse – Plankalkül (1942-1945)
- Rutishauser – Superplan (1949-1951)

The first real compiler:

- Grace-Hopper – A-O system for the UNIVAC (loader and linker; 1951-1952)

She coined the term

FORTRAN

The first commercial compiler was for FORTRAN (IBM), with the development led by Backus.

Backus won a Turing award (1977), and in his lecture discussed how programming should be liberated from the von Neumann style

COBOL

COBOL was a successor of FLOW-MATIC, which was a successor of the A-O system. All projects led by Grace-Hopper

Grace-Hopper got a lot of recognition, but not a Turing award

The first ALGOL languages

Superplan (Rutishauser)

↓

ALGOL 58 (+Bauer, Bottenbruch, Samelson, Perlis, Backus, Katz, Wegstein)

↓

ALGOL 60 (+van Wijngaarden, Naur, Vauquois, Woodger, Green McCarthy) BNF!

“Here is a language so far ahead of its time that it was not only an improvement on its predecessors but also on nearly all its successors” (Tony Hoare)

IFIP WG2.1

IFIP started a Working Group (nr 2.1) to take care of ALGOL 60, and possible innovations, in 1962

The working group designed ALGOL 68

Naur’s proposal for ALGOL-W didn’t make it

Naur subsequently developed Pascal

The group split up in 1968

(Probably the WG with the largest amount of Turing award winners ever collaborating.)

Functional programming

LISP (McCarthy, 1956 – 1958, inspired by the lambda-calculus)

↓

ISWIM (Landin, 1966)

↓ ↓

Miranda (Turner, 1986) ML (Milner, 1973)

↓ ↓

Haskell (Peyton Jones, … 1990) Kotlin (JetBrains, 2011)

The world’s most popular language

ALGOL 60 / ALGOL 68

↓

SETL (Schwartz et al, 1960s, New York)

↓

ABC (Meertens, 1980s, CWI, Amsterdam)

↓

Python (Van Rossum, 1990s, CWI, Amsterdam)

https://www.youtube.com/watch?v=GfH4QL4VqJ0

Summary

The development of programming languages is a complex and intertwined story
People involved in the development of some of the first programming languages are still alive!

There are many fundamental principles developed for or applied in the design of programming languages (functions, function types, recursion, data types, binding, scope, evaluation strategies, lambda calculus, polymorphism)

But also all kinds of ad-hoc decisions:
https://www.destroyallsoftware.com/talks/wat (1:21)