A Brief History of Computing

Xmas Lecture

Prof. Dr. Jörg Schäfer

23 December, 2010

Prologue **3**
Disclaimer . . . 4
Famous (Computer) Scientists Quotes . . . 5
Famous (Computer) Engineers Quotes . . . 6

A Brief History of Computers **7**
Stonehenge . . . 8
Abacus . . . 9
Al-Chwarizmi . . . 10
Leonardo da Vinci . . . 11
Pascaline . . . 12
Leibniz 01 . . . 13
Charles Babbage's Difference Engine . . . 14
Charles Babbage's Analytical Engine . . . 15
The First Programmer . . . 16
Inspiration for Cyberpunk . . . 17
Inspiration for Steampunk . . . 18
Power (Jacquard) Looms . . . 19
IBM, formerly known as Hollerith . . . 20
Konrad Zuse – Z1 . . . 21
Alan Turing [Tur01] . . . 22
John von Neumann [Neu45] . . . 23
Von Neumann Architecture . . . 24
Grace Hopper . . . 25
The First Bug . . . 26
Analog Computers . . . 27
Other Analog Computers . . . 28
Moore's Law . . . 29
Unix [RT74] . . . 30
Macs . . . 31
PCs . . . 32
Ubiquitous Computers . . . 33
Quantum Computing . . . 34

A Brief History of Programming Languages **35**
Why are Languages Important? . . . 36
Languages as Tools for Thinking . . . 37
Language Evolution . . . 38
1940s . . . 39
1940s cont. . . . 40
Machine Language . . . 41
1950s . . . 42
1950s cont. . . . 43

Assembly . . . 44
Fortran – I . . . 45
Fortran – II . . . 46
Algol68 – I . . . 47
Algol68 – II . . . 48
Lisp . . . 49
1960s . . . 50
1960s . . . 51
Basic . . . 52
Cobol . . . 53
1970s . . . 54
1970s . . . 55
Hackers . . . 56
C . . . 57
Prolog . . . 58
Smalltalk . . . 59
1980s . . . 60
1980s . . . 61
C++ – I . . . 62
C++ – II . . . 63
Python . . . 64
1990s . . . 65
1990s cont. . . . 66
1990s cont. . . . 67
Haskell . . . 68
Java . . . 69
2000s . . . 70
MapReduce . . . 71
Summary: Programming Languages' Paradigms . . . 72
Interpretation of Paradigms . . . 73
Shooting Yourself . . . 74
Programming Languages as seen by Fans . . . 75
Languages as Tools for Programming . . . 76

A Brief History of Networks **77**
The World is Connected . . . 78
The Power of the Network . . . 79

A Brief History of Applications **80**
Important Applications . . . 81

A Brief History of Computer Scientists and Engineers **82**
Famous Computer Scientists and Engineers . . . 83

A Brief History of Challenges **84**
Different Disciplines . . . 85
Speaking Different Languages – Example I . . . 86
Speaking Different Languages – Example II . . . 87
What can we Compute? . . . 88
How to deal with Complexity? . . . 89

Outlook **90**
The Future of Computing . . . 91

Appendix **92**
References . . . 93
Credits – I/III . . . 94
Credits – II/III . . . 95
Credits – III/III . . . 96

Content

Prologue

A Brief History of Computers

A Brief History of Programming Languages

A Brief History of Networks

A Brief History of Applications

A Brief History of Computer Scientists and Engineers

A Brief History of Challenges

Outlook

Prologue

Disclaimer

The following slides contain an *entirely subjective* view of (computer) history!

Due to laziness, some notes contain Wikipedia as a "reference" – please consider this as an anti-pattern and *never* use Wikipedia as a "reference" in any serious scientific report!

Famous (Computer) Scientists Quotes

Computer science is no more about computers than astronomy is about telescopes.

Edsger Dijkstra

Computer science also differs from physics in that it is not actually a science. It does not study natural objects. Neither is it, as you might think, mathematics; although it does use mathematical reasoning pretty extensively. Rather, computer science is like engineering; it is all about getting something to do something, rather than just dealing with abstractions, as in the pre-Smith geology.

Richard Feynman, Feynman Lectures on Computation, 1970

Famous (Computer) Engineers Quotes

I think there is a world market for maybe five computers.

Thomas Watson, Chairman of IBM, 1943

There is no reason anyone in the right state of mind will want a computer in their home.

Ken Olson, President of Digital Equipment Corp, 1977

640k is enough for anyone, and by the way, what's a network?

William Gates III, President of Microsoft Corporation, 1984

The most profound technologies are those that disappear: they weave themselves into fabric of everyday life until are indistinguishable from it.

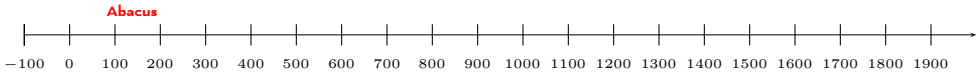
Mark Weiser, The Computer for the 21st Century, Scientific American, 1991

Stonehenge

Stonehenge is a prehistoric monument located in the English county of Wiltshire, about 3.2 kilometres (2.0 mi) west of Amesbury and 13 kilometres (8.1 mi) north of Salisbury. Archaeologists had believed that the iconic stone monument was erected around 2500 BC. (Source: Wikipedia)

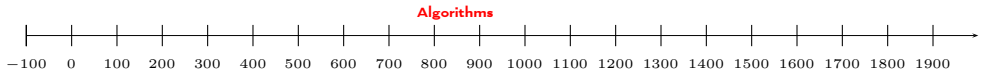
Abacus

Used in many cultures including Egypt, Persia, Greek, Roman, Chinese, India, Japan, Native American, ...



The abacus, also called a counting frame, is a calculating tool used primarily in parts of Asia for performing arithmetic processes. The use of the word abacus dates before 1387 AD, when a Middle English work borrowed the word from Latin to describe a sandboard abacus. The period 2700–2300 BC saw the first appearance of the Sumerian abacus, a table of successive columns which delimited the successive orders of magnitude of their sexagesimal number system. The use of the abacus in Ancient Egypt is mentioned by the Greek historian Herodotus. During the Achaemenid Persian Empire, around 600 BC, Iranians first began to use the abacus. The earliest archaeological evidence for the use of the Greek abacus dates to the 5th century BC. The normal method of calculation in ancient Rome, as in Greece, was by moving counters on a smooth table. The earliest known written documentation of the Chinese abacus dates to the 2nd century BC. Some sources mention the use of an abacus called a nepohualtzintzin in ancient Mayan culture. This Mesoamerican abacus used a 5-digit base-20 system. (Source: Wikipedia)

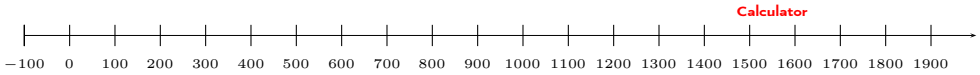
Al-Chwarizmi



Abu Abdallah Muhammad ibn Musa al-Khwarizmi (c. 780 – c. 850) was a Persian mathematician, astronomer and geographer, a scholar in the House of Wisdom in Baghdad. Al-Khwarizmi's contributions to mathematics, geography, astronomy, and cartography established the basis for innovation in algebra and trigonometry. His systematic approach to solving linear and quadratic equations led to algebra, a word derived from the title of his 830 book on the subject, "The Compendious Book on Calculation by Completion and Balancing". On the Calculation with Hindu Numerals written about 825, was principally responsible for spreading the Indian system of numeration throughout the Middle East and Europe. It was translated into Latin as Algoritmi de numero Indorum. Al-Khwarizmi, rendered as (Latin) Algoritmi, led to the term "algorithm". Some of his work was based on Persian and Babylonian astronomy, Indian numbers, and Greek mathematics. When, in the 12th century, his works spread to Europe through Latin translations, it had a profound impact on the advance of mathematics in Europe. He introduced Arabic numerals into the Latin West, based on a place-value decimal system developed from Indian sources. (Source: Wikipedia)

Leonardo da Vinci

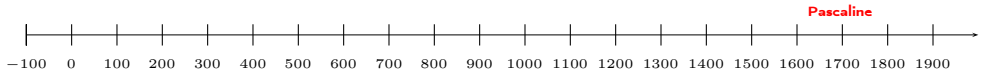
Content (Possibly) Copyright Protected



Device for Calculation: An early version of today's complicated calculator, Leonardo's mechanism maintains a constant ratio of ten to one in each of its 13 digit-registering wheels. For each complete revolution of the first handle, the unit wheel is turned slightly to register a new digit ranging from zero to nine. Consistent with the ten to one ratio, the tenth revolution of the first handle causes the unit wheel to complete its first revolution and register zero, which in turn drives the decimal wheel from zero to one. Each additional wheel marking hundreds, thousands, etc., operates on the same ratio. Slight refinements were made on Leonardo's original sketch to give the viewer a clearer picture of how each of the 13 wheels can be independently operated and yet maintain the ten to one ratio. Leonardo's sketch shows weights to demonstrate the equability of the machine. (Source: IBM)

Pascaline

In 1642 Blaise Pascal, at age 19, invented the Pascaline.



The Pascaline, invented by Blaise Pascal in France in 1642, was a mechanical calculator that could add and subtract directly. Its introduction launched the development of mechanical calculators in Europe first and then all over the world, development which culminated three centuries later by the invention of the microprocessor developed for a Busicom calculator in 1971. Pascal spent three years and went through 50 prototypes before presenting his first machine to the public in 1645. He dedicated it to Pierre Séguier, the chancellor of France at the time. He built around twenty more machines during the next decade, often improving on his original design. Nine machines have survived the centuries, most of them being on display in European museums. In 1649 a royal privilege, signed by Louis XIV of France, gave him the exclusivity of the design and manufacturing of calculating machines in France. The mechanical calculator industry owes a lot of its key machines and inventions to the pascaline. First Gottfried Leibniz invented his Leibniz wheels after 1671 while trying to add an automatic multiplication and division feature to the pascaline, then Thomas de Colmar drew his inspiration from Pascal and Leibniz when he designed his arithmometer in 1820, and finally Dorr E. Felt substituted the input wheels of the pascaline by columns of keys to invent his comptometer around 1887. The pascaline was also constantly improved upon, especially with the machines of Dr. Roth around 1840, and then with some portable machines until the creation of the first electronic calculators. (Source: Wikipedia)

Leibniz 01

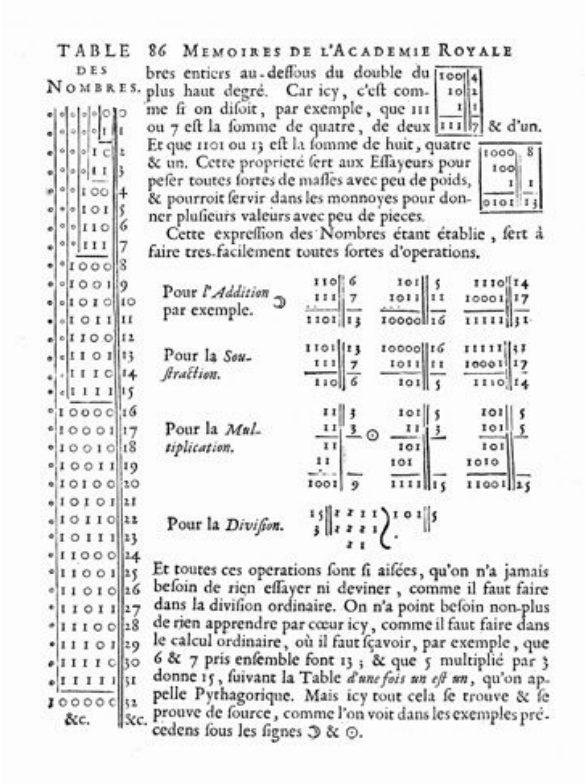


TABLE DES NOMBRES.

86 MEMOIRES DE L'ACADEMIE ROYALE

bres entiers au-deſſous du double du plus haut degré. Car icy, c'eſt comme ſi on diſoit, par exemple, que 111 ou 7 eſt la ſomme de quatre, de deux & d'un. Et que 1101 ou 13 eſt la ſomme de huit, quatre & un. Cette proprieté ſert aux Eſſayeurs pour peſer toutes ſortes de maſſes avec peu de poids, & pourroit ſervir dans les monnoyes pour donner pluſieurs valeurs avec peu de pieces.

Cette expreſſion des Nombres étant établie, ſert à faire tres-facilement toutes ſortes d'operations.

Pour *l'Addition* par exemple.

Pour la *Soustraction.*

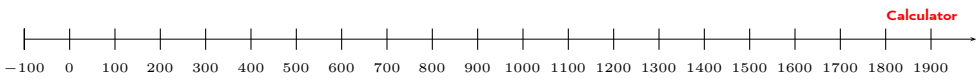
Pour la *Multiplication.*

Pour la *Diviſion.*

Et toutes ces operations ſont ſi aiſées, qu'on n'a jamais beſoin de rien eſſayer ni deviner, comme il faut faire dans la diviſion ordinaire. On n'a point beſoin non-plus de rien apprendre par cœur icy, comme il faut faire dans le calcul ordinaire, où il faut ſçavoir, par exemple, que 6 & 7 pris enſemble font 13; & que 5 multiplié par 3 donne 15, ſuivant la Table *d'une fois un eſt un*, qu'on appelle Pythagorique. Mais icy tout cela ſe trouve & ſe prouve de ſource, comme l'on voit dans les exemples précedens ſous les ſignes ☽ & ☉.

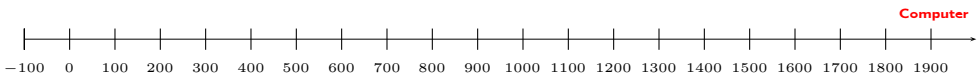
Gottfried Wilhelm Leibniz (July 1, 1646 – November 14, 1716) was a German mathematician and philosopher. Leibniz invented the binary number system and showed the connection with logic (De progressione Dyadica, 1679; oder Explication de l'Arithmetique Binaire, 1703).

Charles Babbage's Difference Engine



Charles Babbage (26 December 1791 – 18 October 1871) was an English mathematician, philosopher, inventor, and mechanical engineer who originated the concept of a programmable computer. He invented the Difference Engine, an automatic, mechanical calculator designed to tabulate polynomial functions. The Difference Engine is not a general purpose computer rather a calculator. Later, he designed the Analytical Engine – a mechanical general purpose computer. The input (programs and data) was to be provided to the machine via punched cards, a method being used at the time to direct mechanical looms such as the Jacquard loom. Both machines were never built during Babbage's life time. (Source: Wikipedia and other sources)

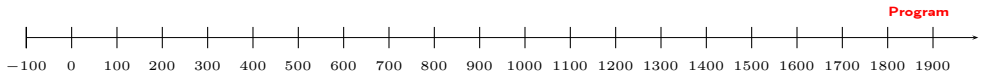
Charles Babbage's Analytical Engine



The analytical engine, an important step in the history of computers, was the design of a mechanical general-purpose computer by English mathematician Charles Babbage. In its logical design the machine was essentially modern, anticipating the first completed general-purpose computers by about 100 years. It was first described in 1837. Babbage continued to refine the design until his death in 1871. Because of the complexity of the machine, the lack of project management science, the expense of its construction, and the difficulty of assessing its value by Parliament relative to other projects being lobbied for, the engine was never built. (Source: Wikipedia and other sources)

The First Programmer

Augusta **Ada** King Byron, Countess of **Lovelace**



Ada Lovelace

- was the only legitimate child of the poet Lord Byron
- wrote programs for computers that did not (yet) exist such as the Analytical Engine (e.g. computing Bernoulli numbers)
- is considered the first programmer in history

Inspiration for Cyberpunk

Content (Possibly) Copyright Protected

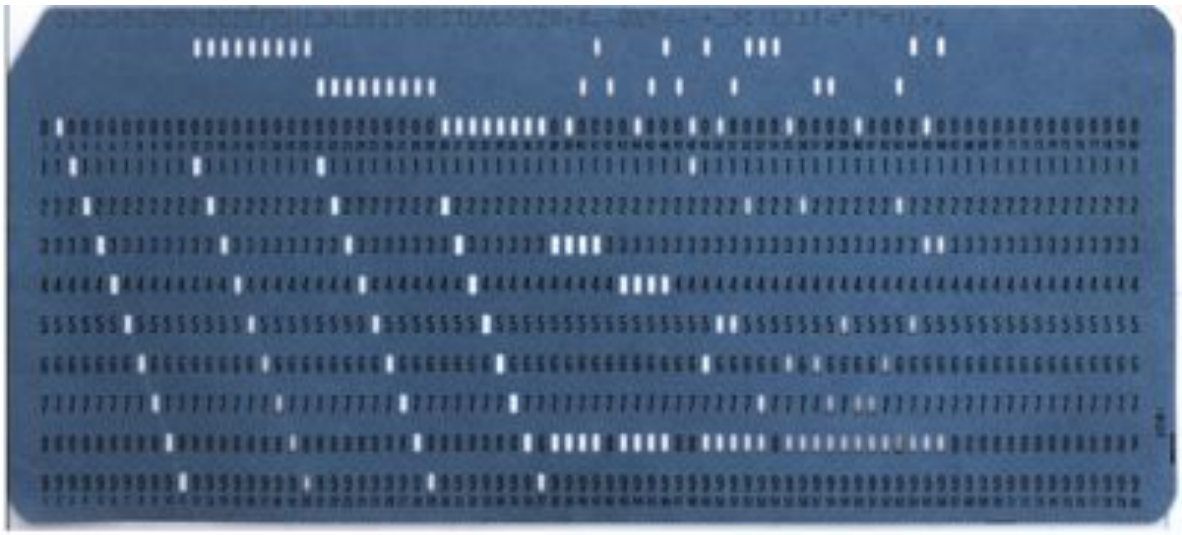
Inspiration for Steampunk

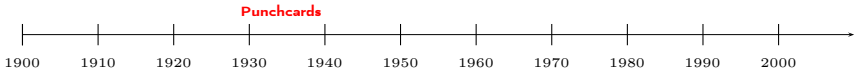
Power (Jacquard) Looms

We may say most aptly, that the Analytical Engine weaves algebraical patterns just as the Jacquard-loom weaves flowers and leaves. (Lady Ada Lovelace)

The Jacquard loom is a mechanical loom, invented by Joseph Marie Jacquard in 1801, that simplifies the process of manufacturing textiles with complex patterns such as brocade, damask, and matelasse. The loom is controlled by punched cards with punched holes, each row of which corresponds to one row of the design. Multiple rows of holes are punched on each card and the many cards that compose the design of the textile are strung together in order. Each position in the card corresponds to a "Bolus" hook, which can either be raised or stopped dependant on whether the hole is punched out of the card or the card is solid. The hook raises or lowers the harness, which carries and guides the warp thread so that the weft will either lie above or below it. The sequence of raised and lowered threads is what creates the pattern. Each hook can be connected via the harness to a number of threads, allowing more than one repeat of a pattern. A loom with a 400-hook head might have four threads connected to each hook, resulting in a fabric that is 1600 warp ends wide with four repeats of the weave going across. The Jacquard loom was the first machine to use punched cards to control a sequence of operations. Although it did no computation based on them, it is considered an important step in the history of computing hardware. The ability to change the pattern of the loom's weave by simply changing cards was an important conceptual precursor to the development of computer programming. Specifically, Charles Babbage planned to use cards to store programs in his Analytical engine. (Source: Wikipedia)

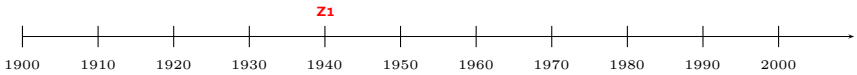
IBM, formerly known as Hollerith





Herman Hollerith (February 29, 1860 – November 17, 1929) was an American statistician who developed a mechanical tabulator based on punched cards to rapidly tabulate statistics from millions of pieces of data. He was the founder of the company that became IBM. Punched card (or punch card or Hollerith card or IBM card) is a piece of stiff paper that contains digital information represented by the presence or absence of holes in predefined positions. Now almost an obsolete recording medium, punched cards were widely used throughout the 19th century for controlling textile looms and in the late 19th and early 20th century for operating fairground organs and related instruments. They were used through the 20th century in unit record machines for input, processing, and data storage. Early digital computers used punched cards, often prepared using keypunch machines, as the primary medium for input of both computer programs and data. Some voting machines use punched cards. (Source: Wikipedia)

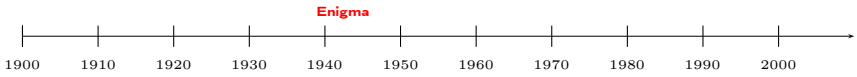
Konrad Zuse – Z1



Konrad Zuse (22 June 1910 Berlin – 18 December 1995 Hünfeld near Fulda) was a German engineer and computer pioneer. His greatest achievement was the world's first functional program-controlled Turing-complete computer, the Z3, in May 1941 (the program was stored on a perforated 35mm film). (Source: Wikipedia)

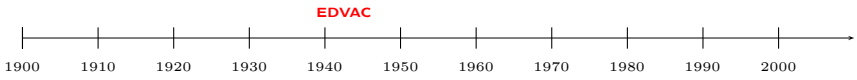
Alan Turing [Tur01]

Content (Possibly) Copyright Protected



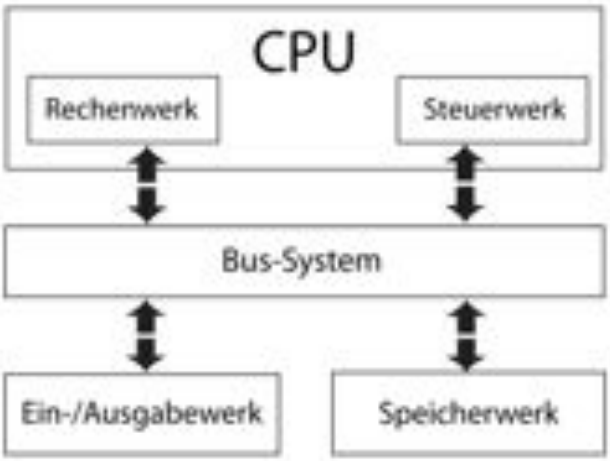
Alan Mathison Turing (born 23 June 1912 at 2 Warrington Crescent, London W9, died 7 June 1954 at his home in Wilmslow, Cheshire) contributed to mathematics, cryptanalysis, logic, philosophy, biology, and formatively to computer science, cognitive science, Artificial Intelligence and Artificial Life. Turing and the American logician Alonzo Church argued that every effective mathematical method can be carried out by the universal Turing machine, a proposition now known as the Church-Turing thesis. Working independently, Turing and Church had both shown that – contrary to mathematical opinion of the day – there are well-defined mathematical problems that cannot be solved by effective methods; each published this result in 1936. This, in conjunction with the work of the Austrian logician Kurt Godel, put paid to the Hilbert programme in mathematics. In the summer of 1938 Turing returned to his Fellowship at King's. At the outbreak of hostilities with Germany in September 1939 he left Cambridge for the wartime headquarters of the Government Code and Cypher School (GC&CS) at Bletchley Park, Buckinghamshire. Building on earlier work by Polish cryptanalysts, Turing contributed crucially to the design of electro-mechanical machines ('bombes') used to decipher Enigma, the code by means of which the German armed forces sought to protect their radio communications. Thanks to the bombes, by early 1942 GC&CS was decoding about 39,000 intercepted messages each month, rising subsequently to over 84,000 messages a month – approximately two every minute. Turing's work on the version of Enigma used by the German navy was vital to the battle for supremacy in the North Atlantic. He also contributed to the attack on the cyphers known as 'Fish'. Based on binary teleprinter code, Fish was used during the latter part of the war in preference to morse-based Enigma for the encryption of high-level signals, for example messages from Hitler and members of the German High Command. It is estimated that the work of GC&CS shortened the war in Europe by at least two years. Turing received the Order of the British Empire for the part he played. (Source: `http://www.alanturing.net/`)
Turing Machine: A Turing machine – described by Alan Turing in 1937 – is a theoretical device that manipulates symbols on a strip of tape according to a table of rules.
Despite his contributions to win the war, Turing was prosecuted by the British Authorities for being homosexual and forced into hormone "treatment". This eventually led to depressions and suicide.

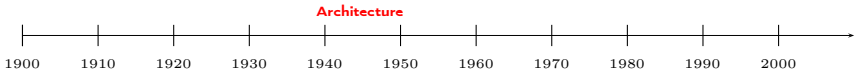
John von Neumann [Neu45]



John von Neumann (December 28, 1903 – February 8, 1957) was a Hungarian-born American mathematician who made major contributions to a vast range of fields, including set theory, functional analysis, quantum mechanics, ergodic theory, continuous geometry, economics and game theory, computer science, numerical analysis, hydrodynamics (of explosions), and statistics, as well as many other mathematical fields. He is generally regarded as one of the greatest mathematicians in modern history. While consulting for the Moore School of Electrical Engineering at the University of Pennsylvania on the EDVAC project, von Neumann wrote an incomplete First Draft of a Report on the EDVAC. The paper, which was widely distributed, described a computer architecture in which the data and the program are both stored in the computer's memory in the same address space. This architecture is to this day the basis of modern computer design, unlike the earliest computers that were 'programmed' by altering the electronic circuitry. Although the single-memory, stored program architecture is commonly called von Neumann architecture as a result of von Neumann's paper, the architecture's description was based on the work of J. Presper Eckert and John William Mauchly, inventors of the ENIAC at the University of Pennsylvania. (Source: Wikipedia)

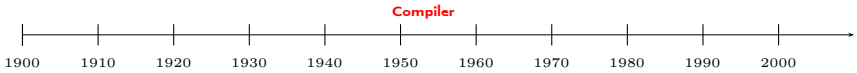
Von Neumann Architecture





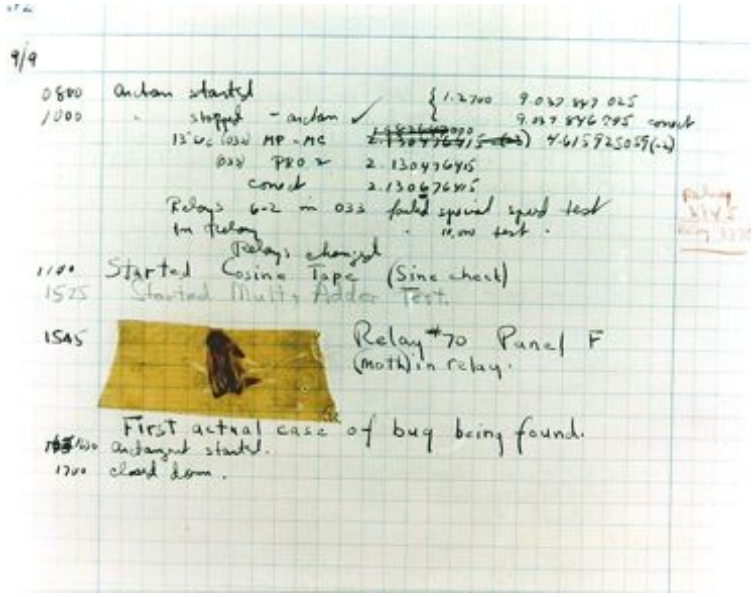
Grace Hopper

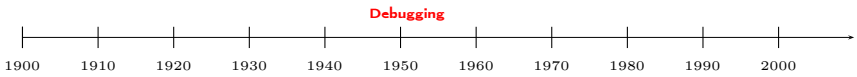
Grace Murray Hopper was an American computer scientist and United States Naval officer. One of the first programmers of the Harvard Mark I computer. She developed the first compiler for a computer programming language.



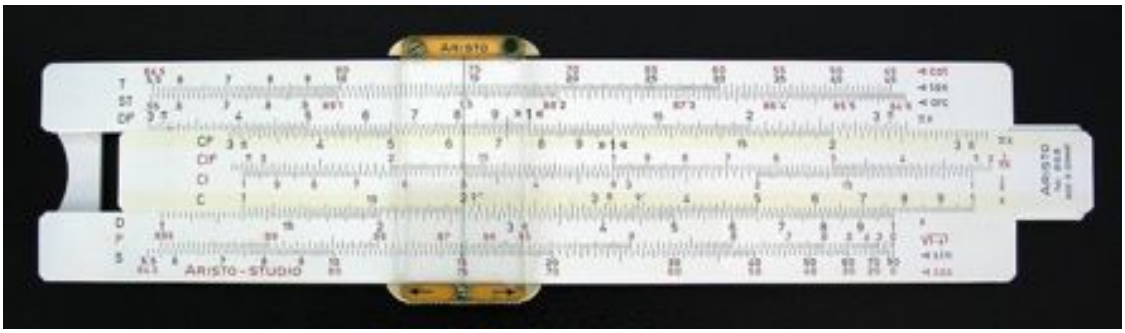
Rear Admiral Grace Murray Hopper (December 9, 1906 – January 1, 1992) was an American computer scientist and United States Naval officer. A pioneer in the field, she was one of the first programmers of the Harvard Mark I computer, and she developed the first compiler for a computer programming language. She conceptualized the idea of machine-independent programming languages, which led to the development of COBOL, one of the first modern programming languages. She is also credited with popularizing the term "debugging" for fixing computer glitches (motivated by an actual moth removed from the computer). Because of the breadth of her accomplishments and her naval rank, she is sometimes referred to as "Amazing Grace". The U.S. Navy destroyer USS Hopper was named for her. (Source: Wikipedia)

The First Bug





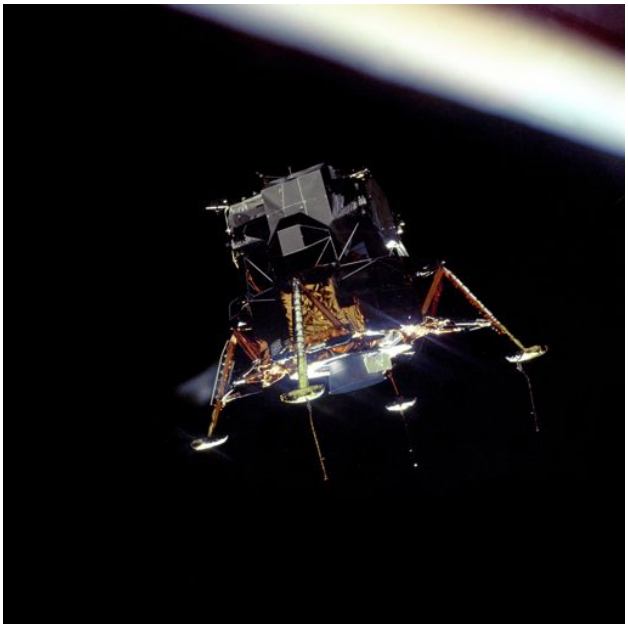
Analog Computers



Sliderule

1900 1910 1920 1930 1940 1950 1960 1970 1980 1990 2000

Other Analog Computers

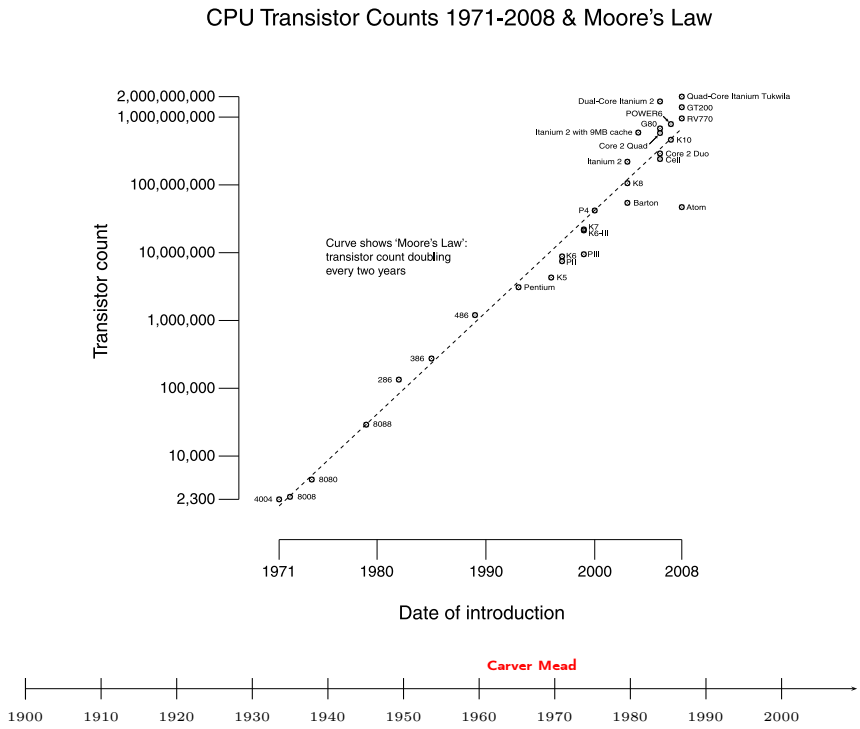


Apollo Lunar Module

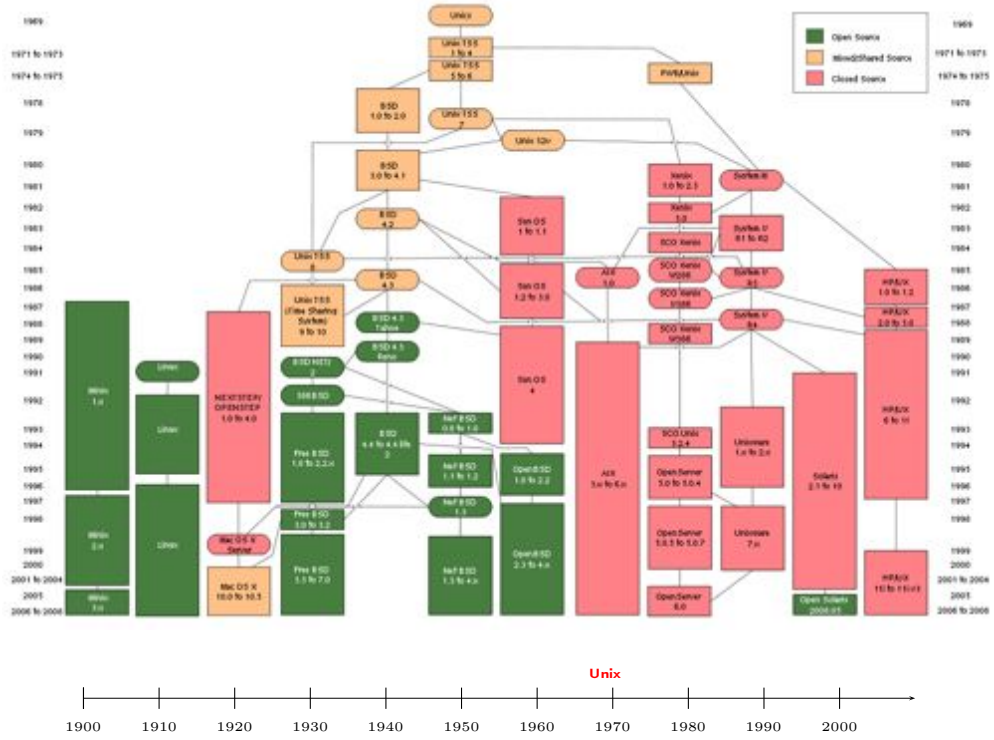
1900 1910 1920 1930 1940 1950 1960 1970 1980 1990 2000

Analog computers were used mainly to solve differential equations, in particular for weapons, missile control and space travel. Examples: Apollo Lunar Module.

Moore's Law

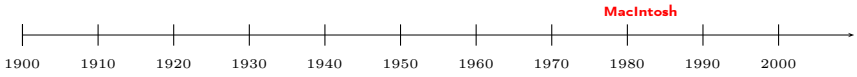


Unix [RT74]

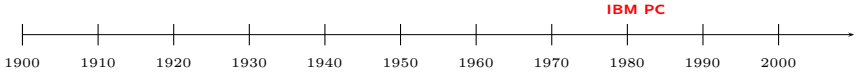


Unix is a computer operating system originally developed in 1969 by a group of AT&T employees at Bell Labs, including Ken Thompson, Dennis Ritchie, Brian Kernighan, Douglas McIlroy, and Joe Ossanna. Today's Unix systems are split into various branches, developed over time by AT&T as well as various commercial vendors and non-profit organizations. The Open Group, an industry standards consortium, owns the "Unix" trademark. (Source: Wikipedia)

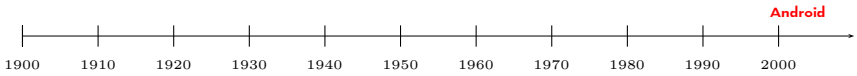
Macs


MacIntosh
1900
1910
1920
1930
1940
1950
1960
1970
1980
1990
2000

PCs

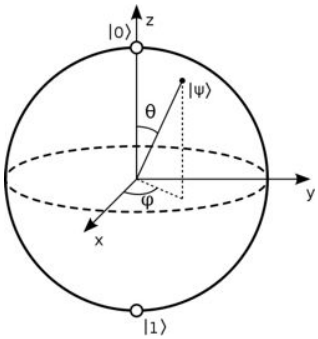

IBM PC
1900
1910
1920
1930
1940
1950
1960
1970
1980
1990
2000

Ubiquitous Computers



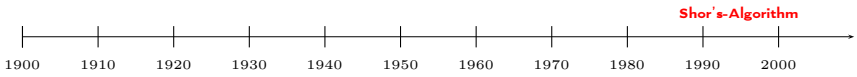
Quantum Computing

Quantum Computing:



Shor's-Algorithm [Sho97]:

$$q^{-1} \sum_{a=0}^{q-1} \sum_{c=0}^{q-1} e^{2\pi i a c/q} |c\rangle |x^a \bmod n\rangle$$



Quantum Computers make use of Quantum Effects and use Q-Bits. They can compute some computations *much* faster than classical computers. For instance, the Shor algorithm, named after mathematician Peter Shor, is a quantum algorithm for integer factorization discovered in 1994. It solves the problem of integer factorization in $O(\log^3 n)$.

A Brief History of Programming Languages 35 / 96

Why are Languages Important?

Human language appears to be a unique phenomenon, without significant analogue in the animal world.

Noam Chomsky

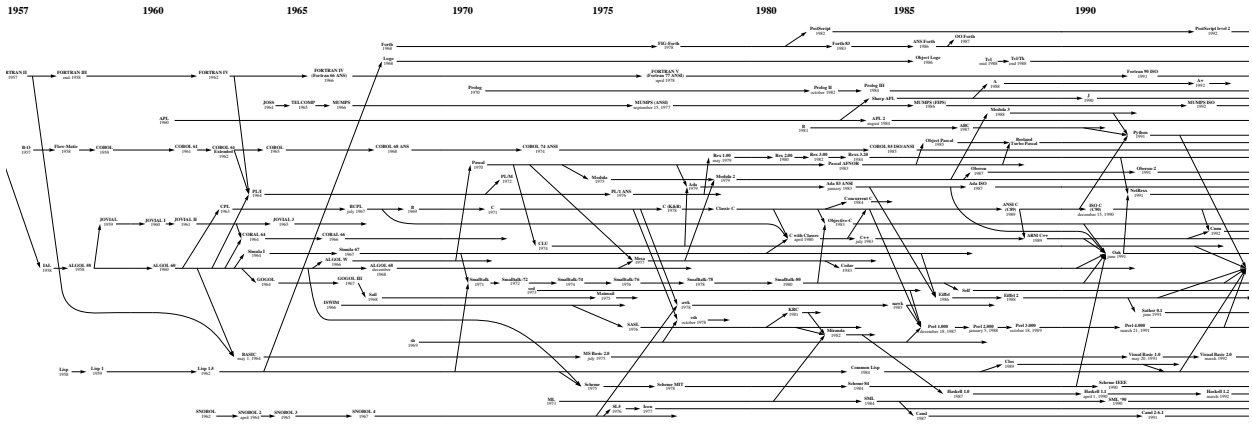
Language is a process of free creation; its laws and principles are fixed, but the manner in which the principles of generation are used is free and infinitely varied. Even the interpretation and use of words involves a process of free creation.

Noam Chomsky

Languages as Tools for Thinking

> Die Grenzen meiner Sprache bedeuten die Grenzen meiner Welt.
>
> Ludwig Wittgenstein [Wit22]

Language Evolution



Messy!

1940s

1940s: It is War. . .

1940s – Some Important Events:

- WW II
- A group of talented cryptographers breaks Enigma based on early work of the Polish Cipher Bureau and Alan Turing's ideas
- Computers play an important role in developing the first nuclear weapons at Alamo

1940s cont.

- No languages or
- Machine Language (i.e. no language)

Machine Language

```
11000111101110101001010010010010101011100110101010011000001111100
10110101011111010100111111111111001011011000000000101001000100100
01100101011011000110110001101111001000000101011101101111011100010
01101100011001000010000101000010011011110110111111000111101110101
10010100100100101010111001101010100110000011110010110101011111101
01001111111111100101101100000000101001000100100000110010101101100
01101100011011110010000001010111011011110111001001101100011001000
00100001010000100110111101101111110001111011101010010100100100010
10101110011010101001100000111100101101010111110101001111111111110
01011011000000001010010001001000011001010110110001101100011011111
00100000010101110110111101110010011011000110010000100001010000010
01101111011011111100011110111010100101001001001010101110011010101
10011000001111001011010101111101010011111111111100101101100000000
10100100010010000110010101101100011011000110111100100000001010111
01101111011100100110110001100100110001111011101010010100100100010
10101110011010101001100000111100101101010111110101001111111111110
01011011000000001010010001001000011001010110110001101100011011111
```

1950s

1950s: Innovation Big Bang!

1950s – Some Important Events:

- Cold war
- Cuban revolution
- Arab–Israeli conflict
- Sputnik shock
- Rock'n' roll

1950s cont.

- Assembly languages were first developed in the 1950s, when they were referred to as second generation programming languages. For example, SOAP (Symbolic Optimal Assembly Program) was a 1957 assembly language for the IBM 650 computer.
- Fortran, invented by John W. Backus optimized for speed (similar to assembly), used for numerical calculations in e.g. physics until today.
- Algol, the mother of procedural languages (such as Pascal)
- Lisp, the mother of functional languages

Fortran and Algol are *imperative* and *procedural*. Lisp is primarily *functional*.

Assembly

```
_partition:
LFB3:
    pushq   %rbp
LCFI0:
    movq    %rsp, %rbp
LCFI1:
    pushq   %rbx
LCFI2:
    movl    %edx, %r10d
    movslq  %edx,%rax
    leaq    (%rdi,%rax,4), %rbx
    movl    (%rbx), %r11d
    leal    -1(%rsi), %r9d
    cmpl    %esi, %edx
    jle L2
    movslq  %esi,%rax
    leaq    (%rdi,%rax,4), %rcx
    .align 4,0x90
L4:
    movl    (%rcx), %r8d
    cmpl    %r8d, %r11d
    jl  L5
    incl    %r9d
    movslq  %r9d,%rax
    leaq    (%rdi,%rax,4), %rax
    movl    (%rax), %edx
    movl    %edx, (%rcx)
...
```

Assembly ideas

- use mnemonics that symbolize processing steps (instructions), processor registers, memory locations, and other language features rather than 0s and 1s
- needs a simple compiler (assembler) – more or less isomorphic
- debugging aids added later
- one and two-pass assemblers Elements:
  - Opcode mnemonics
  - Data sections
  - Assembly directives

Fortran – I

```
MODULE Qsort_Module

IMPLICIT NONE

CONTAINS

RECURSIVE SUBROUTINE Qsort(a)

  INTEGER, INTENT(IN OUT) :: a(:)
  INTEGER :: split

  IF(size(a) > 1) THEN
     CALL Partition(a, split)
     CALL Qsort(a(:split-1))
     CALL Qsort(a(split:))
  END IF

END SUBROUTINE Qsort
```

Fortran – II

```
SUBROUTINE Partition(a, marker)

  INTEGER, INTENT(IN OUT) :: a(:)
  INTEGER, INTENT(OUT) :: marker
  INTEGER :: left, right, pivot, temp

  pivot = (a(1) + a(size(a))) / 2  ! Average of first and last elements to prevent quadratic
  left = 0                         ! behavior with sorted or reverse sorted data
  right = size(a) + 1

  DO WHILE (left < right)
     right = right - 1
     DO WHILE (a(right) > pivot)
        right = right-1
     END DO
     left = left + 1
     DO WHILE (a(left) < pivot)
        left = left + 1
     END DO
     IF (left < right) THEN
        temp = a(left)
        a(left) = a(right)
        a(right) = temp
     END IF
  END DO

  IF (left == right) THEN
     marker = left + 1
  ELSE
     marker = left
  END IF

END SUBROUTINE Partition
END MODULE Qsort_Module
```


Fortran ([...] derived from IBM Mathematical Formula Translating System) is a general-purpose, procedural, imperative programming language that is especially suited to numeric computation and scientific computing. Originally developed by IBM at their campus in south San Jose, California in the 1950s for scientific and engineering applications, Fortran came to dominate this area of programming early on and has been in continual use for over half a century in computationally intensive areas such as numerical weather prediction, finite element analysis, computational fluid dynamics, computational physics and computational chemistry. It is one of the most popular languages in the area of high-performance computing and is the language used for programs that benchmark and rank the world's fastest supercomputers. In late 1953, John W. Backus submitted a proposal to his superiors at IBM to develop a more practical alternative to assembly language for programming their IBM 704 mainframe computer. Backus' historic FORTRAN team consisted of programmers Richard Goldberg, Sheldon F. Best, Harlan Herrick, Peter Sheridan, Roy Nutt, Robert Nelson, Irving Ziller, Lois Haibt, and David Sayre. [...] While the community was skeptical that this new method could possibly out-perform hand-coding, it reduced the number of programming statements necessary to operate a machine by a factor of 20, and quickly gained acceptance. Said creator John Backus during a 1979 interview with Think, the IBM employee magazine, "Much of my work has come from being lazy. I didn't like writing programs, and so, when I was working on the IBM 701, writing programs for computing missile trajectories, I started work on a programming system to make it easier to write programs." (Source: Wikipedia)

Algol68 – I

```
PROC partition =(REF [] DATA array, PROC (REF DATA, REF DATA) BOOL cmp)INT: (
  INT begin:=LWB array;
  INT end:=UPB array;
  WHILE begin < end DO
     WHILE begin < end DO
      IF cmp(array[begin], array[end]) THEN
        DATA tmp=array[begin];
        array[begin]:=array[end];
        array[end]:=tmp;
        GO TO break while decr end
      FI;
      end -:= 1
    OD;
    break while decr end: SKIP;
     WHILE begin < end DO
      IF cmp(array[begin], array[end]) THEN
        DATA tmp=array[begin];
         array[begin]:=array[end];
        array[end]:=tmp;
        GO TO break while incr begin
      FI;
      begin +:= 1
     OD;
     break while incr begin: SKIP
  OD;
  begin
);
```


Algol68 – II

```
PROC qsort=(REF [] DATA array, PROC (REF DATA, REF DATA) BOOL cmp)VOID: (
  IF LWB array < UPB array THEN
    INT i := partition(array, cmp);
    PAR ( # remove PAR for single threaded sort #
      qsort(array[:i-1], cmp),
      qsort(array[i+1:], cmp)
    )
  FI
);

MODE DATA = INT;
PROC cmp=(REF DATA a,b)BOOL: a>b;

main:(
  []DATA const l=(5,4,3,2,1);
  [UPB const l]DATA l:=const l;
  qsort(l,cmp);
  printf(($g(3)$,l))
)
```

ALGOL (short for ALGOrithmic Language) is a family of imperative computer programming languages originally developed in the mid 1950s which greatly influenced many other languages and became the de facto way algorithms were described in textbooks and academic works for almost the next 30 years. It was designed to avoid some of the perceived problems with FORTRAN and eventually gave rise to many other programming languages, including BCPL, B, Pascal, Simula, C, and many others. ALGOL introduced code blocks and the begin and end pairs for delimiting them and it was also the first language implementing nested function definitions with lexical scope. Fragments of ALGOL-like syntax are sometimes still used as pseudocode (notations used to describe algorithms for human readers). ALGOL was developed jointly by a committee of European and American computer scientists in a meeting in 1958 at ETH Zurich (cf. ALGOL 58). It specified three different syntaxes: a reference syntax, a publication syntax, and an implementation syntax. The different syntaxes permitted it to use different keyword names and conventions for decimal points (commas vs periods) for different languages. ALGOL was used mostly by research computer scientists in the United States and in Europe. Its use in commercial applications was hindered by the absence of standard input/output facilities in its description and the lack of interest in the language by large computer vendors. (Source: Wikipedia)

Lisp

```
(defun qs (list)
  (if (<= (length list) 1)
      list
      (let ((pivot (first list)))
        (append (qs (remove-if-not #'(lambda (x) (< x pivot)) list))
                (remove-if-not #'(lambda (x) (= x pivot)) list)
                (qs (remove-if-not #'(lambda (x) (> x pivot)) list))))))
```

In 1960, John McCarthy published a remarkable paper in which he did for programming something like what Euclid did for geometry. He showed how, given a handful of simple operators and a notation for functions, you can build a whole programming language. He called this language Lisp, for "List Processing," because one of his key ideas was to use a simple data structure called a list for both code and data.
It's worth understanding what McCarthy discovered, not just as a landmark in the history of computers, but as a model for what programming is tending to become in our own time. [...] It seems to me that there have been two really clean, consistent models of programming so far: the C model and the Lisp model. [...] Lisp wasn't designed to fix the mistakes in Fortran; it came about more as the byproduct of an attempt to axiomatize computation. (Source: Paul Graham, see `http://www.paulgraham.com`)

1960s

Content (Possibly) Copyright Protected

1960s: Swinging London (but not in IT)!

1960s – Some Important Events:

- Cold war
- Cuba crisis
- Vietnam war
- Apollo programme
- Rock music
- Black civil rights movement
- Great Proletarian Cultural Revolution in China (death toll 20 to 30 million)
- Mass socialist or communist movement in most European countries (particularly France and Italy)
- Student protest culture
- Computers becoming used outside science and military
- Gradual progress in programming language design

1960s

- Basic was developed by John George Kemeny and Thomas Eugene Kurtz at the Dartmouth College
- COBOL

Both are *imperative* and *procedural*.

Basic

```
Procedure qSort(Array a(1), firstIndex, lastIndex)
  Protected  low, high, pivotValue

  low = firstIndex
  high = lastIndex
  pivotValue = a((firstIndex + lastIndex) / 2)

  Repeat

    While a(low) < pivotValue
      low + 1
    Wend

    While a(high) > pivotValue
      high - 1
    Wend

    If low <= high
      Swap a(low), a(high)
      low + 1
      high - 1
    EndIf

  Until low > high

  If firstIndex < high
    qSort(a(), firstIndex, high)
  EndIf

  If low < lastIndex
    qSort(a(), low, lastIndex)
  EndIf
EndProcedure
```


The original BASIC was designed in 1964 by John George Kemeny and Thomas Eugene Kurtz at Dartmouth College in New Hampshire, USA to provide computer access to non-science students. (Source: Wikipedia)

Cobol

Implementation is not shown, because

1. Cobol is too ugly
2. Originally, you could not implement recursion in Cobol at all (it is a language for business people, after all)

The COBOL (Common Business Oriented Language) specification was created by Grace Murray Hopper during the second half of 1959. COBOL defines its primary domain in business, finance, and administrative systems for companies and governments.

1970s

1970s: Revolution – New Frontiers!

1970s – Some Important Events:

- Cold war continues
- Vietnam war
- Apollo programme cancelled
- Pop music
- Hippie movement
- Student protest culture
- Computers becoming smaller and affordable (Moore's Law)

1970s

- C, developed in 1972 by Dennis Ritchie at the Bell Telephone Laboratories for use with the Unix operating system. The one and only systems language. Imperative and procedural, a better high level assembly.
- Prolog , invented by Alain Colmeraue, is a general purpose declarative logic programming language mainly used for artificial intelligence and computational linguistics.
- Alan Kay, Dan Ingalls, Adele Goldberg, Ted Kaehler, Scott Wallace, and other created Smalltalk-80 at Xerox PARC, the mother of all (good) object-oriented languages, based on Simula (from the 60s!)[a]

[a] So OO is indeed a very old technology.

Hackers

C

```
void quick(int *left, int *right)
{
  if (right > left) {
    int pivot = left[(right-left)/2];
    int *r = right, *l = left;
    do {
      while (*l < pivot) l++;
      while (*r > pivot) r--;
      if (l <= r) {
        int t = *l;
        *l++ = *r;
        *r-- = t;
      }
    } while (l <= r);
    quick(left, r);
    quick(l, right);
  }
}

void sort(int *array, int length)
{
  quick(array, array+length-1);
}
```


C is a general-purpose computer programming language developed in 1972 by Dennis Ritchie at the Bell Telephone Laboratories for use with the Unix operating system. Although C was designed for implementing system software, it is also widely used for developing portable application software. C is one of the most popular programming languages of all time and there are very few computer architectures for which a C compiler does not exist. C has greatly influenced many other popular programming languages, most notably C++, which began as an extension to C. C is an imperative (procedural) systems implementation language. It was designed to be compiled using a relatively straightforward compiler, to provide low-level access to memory, to provide language constructs that map efficiently to machine instructions, and to require minimal run-time support. C was therefore useful for many applications that had formerly been coded in assembly language. Despite its low-level capabilities, the language was designed to encourage cross-platform programming. A standards-compliant and portably written C program can be compiled for a very wide variety of computer platforms and operating systems with little or no change to its source code. The language has become available on a very wide range of platforms, from embedded microcontrollers to supercomputers.
Design goal minimalism: C's design is tied to its intended use as a portable systems implementation language. It provides simple, direct access to any addressable object (for example, memory-mapped device control registers), and its source-code expressions can be translated in a straightforward manner to primitive machine operations in the executable code. (Source: Wikipedia)

Prolog

```
sort( [], [] ).
qsort( [X], [X] ).
qsort( [H|U], S ) :- splitBy(H, U, L, R), qsort(L, SL), qsort(R, SR),
                     combine(H, SL, SR, S).

% splitBy( H, U, LS, RS )
% True if LS = { L in U | L <= H }; RS = { R in U | R > H }
splitBy( H, [], LS, RS).
splitBy( H, [U|T], [U|LS], RS ) :- U =< H, splitBy(H, T, LS, RS).
splitBy( H, [U|T], LS, [U|RS] ) :- U  > H, splitBy(H, T, LS, RS).

% combine( H, L, R, S )
% True if S is L ++ [H] ++ R (in Haskell notation)
combine( H, L, R, S ) :- append(L, [H|R], S).
```


Prolog is a general purpose logic programming language associated with artificial intelligence and computational linguistics. Prolog has its roots in formal logic, and unlike many other programming languages, Prolog is declarative: The program logic is expressed in terms of relations, represented as facts and rules. A computation is initiated by running a query over these relations. The language was first conceived by a group around Alain Colmerauer in Marseille, France, in the early 1970s and the first Prolog system was developed in 1972 by Colmerauer with Philippe Roussel. Prolog was one of the first logic programming languages, and remains among the most popular such languages today, with many free and commercial implementations available. While initially aimed at natural language processing, the language has since then stretched far into other areas like theorem proving, expert systems, games, automated answering systems, ontologies and sophisticated control systems. (Source: Wikipedia)

Smalltalk

```
" This must be inserted into the class Array as an instance method "

quicksort: l to: r
" sorts the array between l and r after the quicksort method
  by Michael Neumann 1998
"
| m il ir temp |
   (r > l) ifTrue: [
      m ← self at: ((l + r) // 2).
      il ← l.
      ir ← r.
      [
         [(self at: il) < m] whileTrue: [il ← il + 1.].
         [(self at: ir) > m] whileTrue: [ir ← ir - 1.].
         (il < ir) ifTrue: [
            " swap "
            temp ← self at: il.
            self at: il put: (self at: ir).
            self at: ir put: temp.
         ].
      ] doUntil: [il >= ir.].
      self quicksort: l to: (il - 1).
      self quicksort: (il + 1) to: r.
   ].
↑self.
```


Smalltalk is an object-oriented, dynamically typed, reflective programming language. Smalltalk was created as the language to underpin the "new world" of computing exemplified by "human–computer symbiosis." It was designed and created in part for educational use, more so for constructionist learning, at the Learning Research Group (LRG) of Xerox PARC by Alan Kay, Dan Ingalls, Adele Goldberg, Ted Kaehler, Scott Wallace, and others during the 1970s. [...] Smalltalk was the product of researchers led by Alan Kay at Xerox Palo Alto Research Center (PARC); Alan Kay designed most of the early Smalltalk versions, which Dan Ingalls implemented. The first version, known as Smalltalk-71, was created by Ingalls in a few mornings on a bet that a programming language based on the idea of message passing inspired by Simula could be implemented in "a page of code." A later variant actually used for research work is now known as Smalltalk-72 and influenced the development of the Actor model. Its syntax and execution model were very different from modern Smalltalk variants. (Source: Wikipedia)
Smalltalk is pure OO – in Smalltalk *everything* is an object (including methods and classes).

1980s

1980s: OO is in the Air!

1980s – Some Important Events:

- Cold war ends
- End of communism (except North Korea, China and Cuba)
- Hip Hop
- PCs
- OO revolution

1980s

- C++ was invented as a better (really?!) C by Bjarne Stroustrup. It is a multi-paradigm language supporting procedural programming and OO concepts.
- Guido van Rossum created Python, a modern script language with functional and OO elements.

C++ – I

```
#include <iterator>
#include <algorithm> // for std::partition
#include <functional> // for std::less

// helper function for median of three
template<typename T>
 T median(T t1, T t2, T t3)
{
  if (t1 < t2)
  {
    if (t2 < t3)
      return t2;
    else if (t1 < t3)
      return t3;
    else
      return t1;
  }
  else
  {
    if (t1 < t3)
      return t1;
    else if (t2 < t3)
      return t3;
    else
      return t2;
  }
}
```

C++ – II

```
// helper object to get <= from <
template<typename Order> struct non_strict_op:
  public std::binary_function<typename Order::second_argument_type,
                              typename Order::first_argument_type,
                              bool>
{
  non_strict_op(Order o): order(o) {}
  bool operator()(typename Order::second_argument_type arg1,
                  typename Order::first_argument_type arg2) const
  {
    return !order(arg2, arg1);
  }
private:
  Order order;
};

template<typename Order> non_strict_op<Order> non_strict(Order o)
{
  return non_strict_op<Order>(o);
}

...

template<typename RandomAccessIterator>
 void quicksort(RandomAccessIterator first, RandomAccessIterator last)
{
  quicksort(first, last,
    std::less<typename std::iterator_traits<RandomAccessIterator>::value_type>());
}
```


Python

```
def qsort(list):
    if not list:
        return []
    else:
        pivot = list[0]
        less = [x for x in list      if x <  pivot]
        more = [x for x in list[1:] if x >= pivot]
        return qsort(less) + [pivot] + qsort(more)
```

1990s

1990s: Evolutionary Improvements!

1990s cont.

http://info.cern.ch/

1990s cont.

- Haskell, named after logician Haskell Curry is a standardized, general-purpose purely functional programming language, with non-strict semantics and strong static typing. "A proof is a program; the formula it proves is a type for the program". Designed by Simon Peyton Jones, Paul Hudak, Philip Wadler, et al. Used in the research community.
- Java invented by SUN (James Gosling, Mike Sheridan, and Patrick Naughton), is a general purpose OO language with C++ syntax.

1990s – Some Important Events:

- End of history announced
- Civil and irregular wars
- MTV
- OO becoming mainstream
- The Internet becomes mainstream

Haskell

```
qsort :: Ord a => [a] -> [a]
qsort [] = []
qsort (p:xs) = qsort lesser ++ [p] ++ qsort greater
    where
        lesser  = filter (< p) xs
        greater = filter (>= p) xs
```


Haskell is a standardized, general-purpose purely functional programming language, with non-strict semantics and strong static typing. It is named after logician Haskell Curry. In Haskell, "a function is a first-class citizen" of the programming language. As a functional programming language, the primary control construct is the function; the language is rooted in the observations of Haskell Curry and his intellectual descendants, that "a proof is a program; the formula it proves is a type for the program".[...] Following the release of Miranda by Research Software Ltd, in 1985, interest in lazy functional languages grew: by 1987, more than a dozen non-strict, purely functional programming languages existed. Of these, Miranda was the most widely used, but was not in the public domain. At the conference on Functional Programming Languages and Computer Architecture (FPCA '87) in Portland, Oregon, a meeting was held during which participants formed a strong consensus that a committee should be formed to define an open standard for such languages. The committee's purpose was to consolidate the existing functional languages into a common one that would serve as a basis for future research in functional-language design. (Source: Wikipedia)

Java

```
public static <E extends Comparable<? super E>> List<E> quickSort(List<E> arr) {
    if (arr.size() <= 1)
        return arr;
    E pivot = arr.getFirst(); //This pivot can change to get faster results

    List<E> less = new LinkedList<E>();
    List<E> pivotList = new LinkedList<E>();
    List<E> more = new LinkedList<E>();

    // Partition
    for (E i: arr) {
        if (i.compareTo(pivot) < 0)
            less.add(i);
        else if (i.compareTo(pivot) > 0)
            more.add(i);
        else
            pivotList.add(i);
    }

    // Recursively sort sublists
    less = quickSort(less);
    more = quickSort(more);

    // Concatenate results
    less.addAll(pivotList);
    less.addAll(more);
    return less;
}
```

Java refers to a number of computer software products and specifications from Sun Microsystems, a subsidiary of Oracle Corporation, that together provide a system for developing application software and deploying it in a cross-platform environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones on the low end, to enterprise servers and supercomputers on the high end. Java is used in mobile phones, Web servers and enterprise applications, and while less common on desktop computers, Java applets are often used to provide improved and secure functionalities while browsing the World Wide Web. Writing in the Java programming language is the primary way to produce code that will be deployed as Java bytecode, though there are bytecode compilers available for other languages such as Ada, JavaScript, Python, and Ruby. Several new languages have been designed to run natively on the Java Virtual Machine (JVM), such as Scala, Clojure and Groovy. Java syntax borrows heavily from C and C++, but object-oriented features are modeled after Smalltalk and Objective-C. Java eliminates certain low-level constructs such as pointers and has a very simple memory model where every object is allocated on the heap and all variables of object types are references. Memory management is handled through integrated automatic garbage collection performed by the JVM.
The Java platform and language began as an internal project at Sun Microsystems in December 1990, providing an alternative to the C++/C programming languages. Engineer Patrick Naughton had become increasingly frustrated with the state of Sun's C++ and C APIs (application programming interfaces) and tools. While considering moving to NeXT, Naughton was offered a chance to work on new technology and thus the Stealth Project was started. The Stealth Project was soon renamed to the Green Project with James Gosling and Mike Sheridan joining Naughton. Together with other engineers, they began work in a small office on Sand Hill Road in Menlo Park, California. They were attempting to develop a new technology for programming next generation smart appliances, which Sun expected to be a major new opportunity. The team originally considered using C++, but it was rejected for several reasons. Because they were developing an embedded system with limited resources, they decided that C++ demanded too large a footprint and that its complexity led to developer errors. The language's lack of garbage collection meant that programmers had to manually manage system memory, a challenging and error-prone task. The team was also troubled by the language's lack of portable facilities for security, distributed programming, and threading. Finally, they wanted a platform that could be easily ported to all types of devices. Bill Joy had envisioned a new language combining Mesa and C. In a paper called Further, he proposed to Sun that its engineers should produce an object-oriented environment based on C++. Initially, Gosling attempted to modify and extend C++ (which he referred to as "C++ ++ −") but soon abandoned that in favor of creating an entirely new language, which he called Oak, after the tree that stood just outside his office. (Source: Wikipedia)

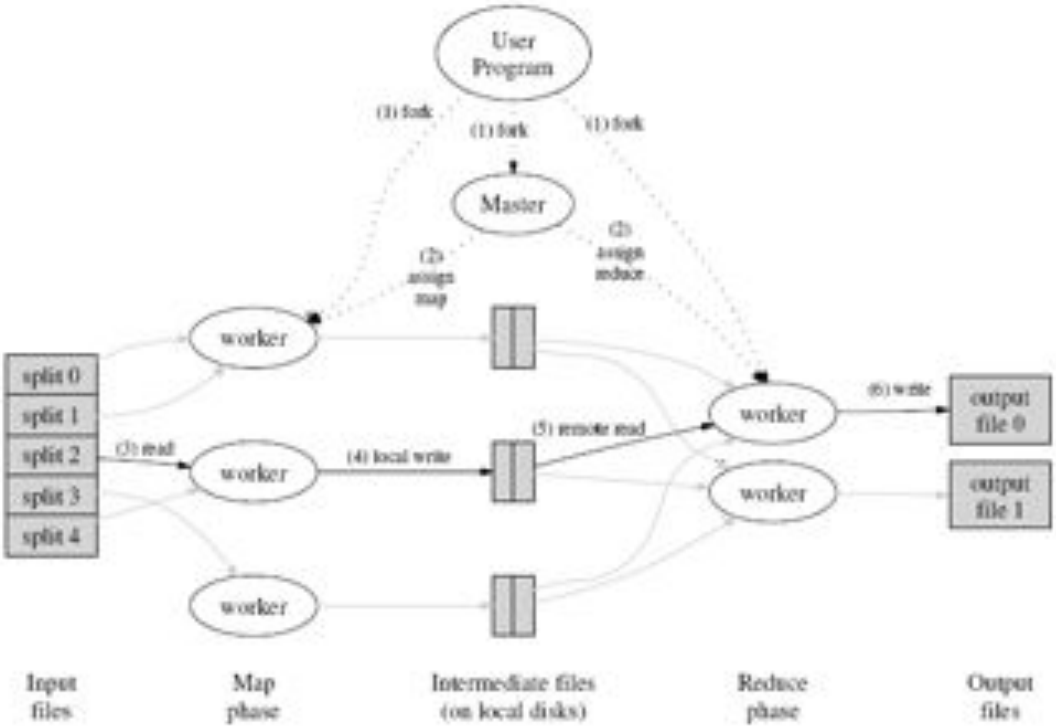
2000s

2000s: Clouds. . .

2000s – Some Important Events:

- World Trade Center
- War on terrorism
- The Internet becomes indispensable
- Social Networks
- Cloud Computing

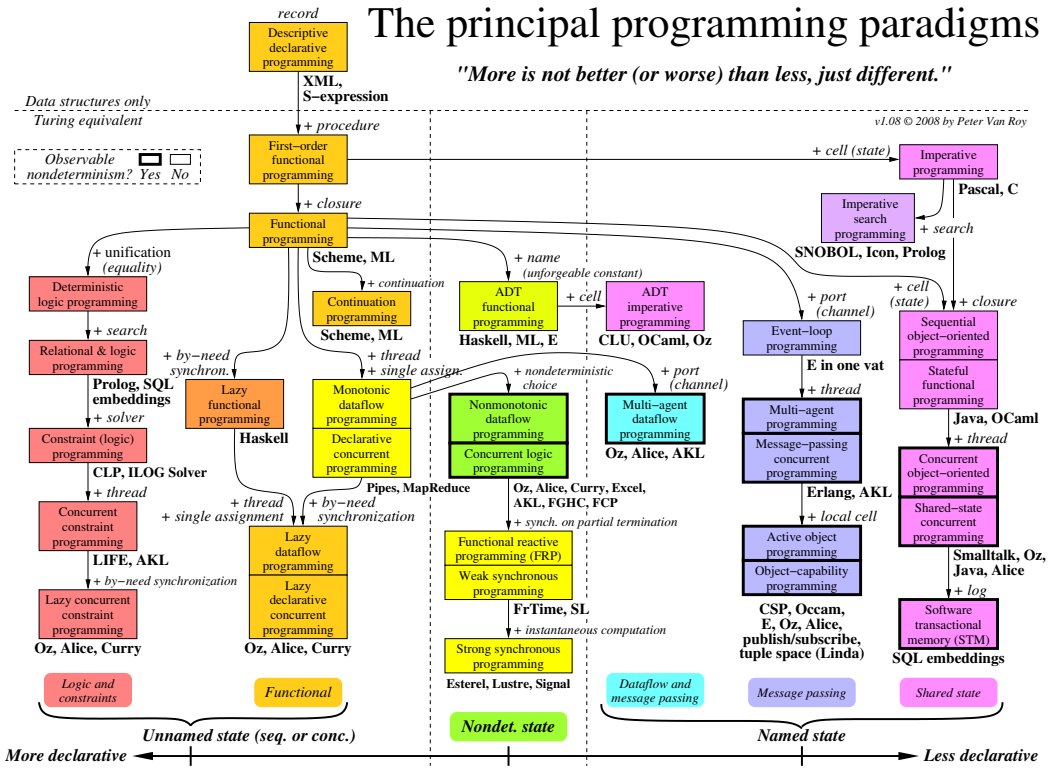
MapReduce



Summary: Programming Languages' Paradigms

- Imperative: Computation in terms of program state and statements that change the program state. Specifies steps the program must take to reach the desired state. Examples: Assembly, C, C++, Java
- Declarative: Describes *what* rather than *how*. Examples: HTML, SQL, Prolog
- Procedural: A subset of imperative languages consisting of procedures (aka routines, subroutines, methods, or functions), each containing a series of computational steps to be carried out. Examples: C, C++, Java
- Functional: Computation is the evaluation of mathematical functions. Avoids state and mutable data. Examples: Lisp, ML, Haskell
- Constraint: Relations between variables are stated in the form of constraints. Constraints do not specify a sequence of steps to execute, but rather the properties of a solution to be found. Searches for a state of the world in which a number of constraints are satisfied at the same time. Examples: Prolog
- Logic: Pattern-directed invocation of procedures from assertions and goals. When a set of predicates is satisfied, an action is taken. Examples: Prolog

Interpretation of Paradigms



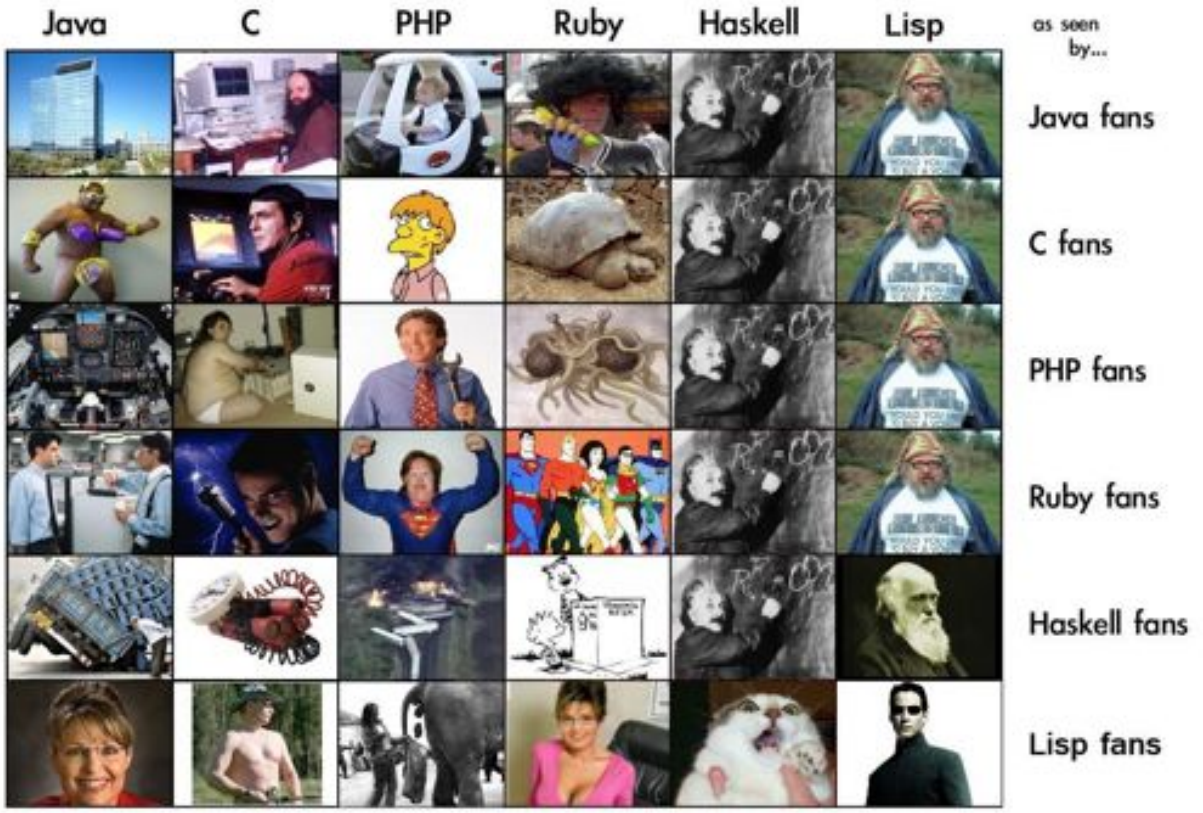
Shooting Yourself

How to Shoot Yourself In the Foot Using Any Programming Language:

- **C:** You shoot yourself in the foot and then nobody else can figure out what you did.
- **C++:** You accidentally create a dozen clones of yourself and shoot them all in the foot. Emergency medical assistance is impossible since you can't tell which are bitwise copies and which are just pointing at others and saying, "That's me, over there."
- **Java:** After importing java.awt.right.foot.* and java.awt.gun.right.hand.*, and writing the classes and methods of those classes needed, you've forgotten what the hell you're doing.
- **Haskell:** You shoot yourself in the foot very elegantly, and wonder why the whole world isn't shooting itself this way.
- **Lisp:** You shoot yourself in the appendage which holds the gun with which you shoot yourself in the appendage which holds the gun with which you shoot yourself in the appendage which holds the gun with which you shoot...
- **Smalltalk:** You shoot yourself in the foot and your foot sends doesNotUnderstand: Pain to your brain.
- **XML:** You can't actually shoot yourself in the foot; all you can do is describe the gun in painful detail.

(Sources: http://www.toodarkpark.org/computers/humor/shoot-self-in-foot.html **and** http://thealmightyguru.com/Humor/Docs/ShootYourselfInTheFoot.html**)**

Programming Languages as seen by Fans



Languages as Tools for Programming

A language that doesn't affect the way you think about programming, is not worth knowing.

Alan Perlis

The World is Connected

> The network is the computer
>
> Scott McNealy, co-founder of Sun Microsystems

Unfortunately, this is the topic for another lecture, please ask Prof. Hoffmann for next year's Xmas lecture!

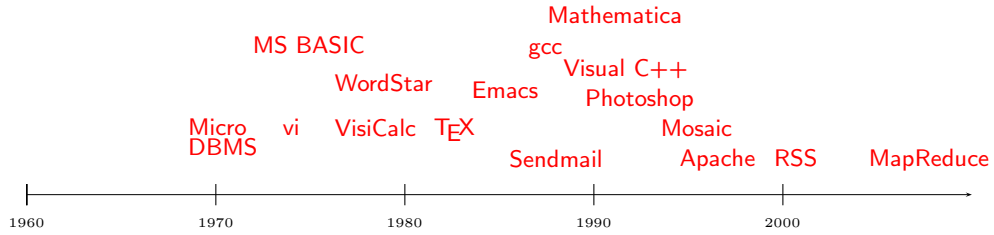
The Power of the Network

213.251.145.96

A Brief History of Applications

Important Applications

Warning: The following choice is ridiculously subjective!



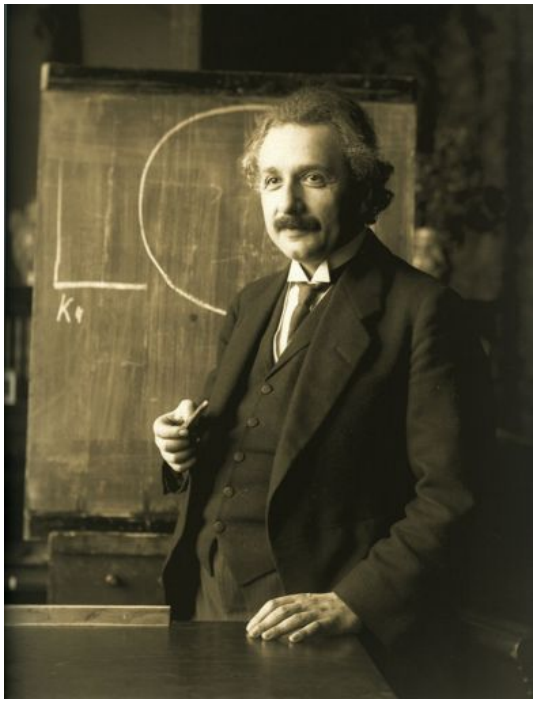
A Brief History of Computer Scientists and Engineers

Famous Computer Scientists and Engineers

Gottfried Wilhelm Leibniz Charles Babbage Ada Lovelace John von Neumann Alan Turing Kurt Gödel John Atanasoff Alonzo Church Claude Shannon Konrad Zuse Vannevar Bush C. A. R. Hoare Stephen Cole Kleene Edgar F. Codd John Backus Marvin Minsky Barbara Liskov Ron Rivest Adi Shamir Leonard Adleman Dennis M. Ritchie Alan Kay Bill Joy Peter Naur Ken Thompson Gordon E. Moore Fred Brooks Niklaus Wirth Donald Knuth Linus Torvalds Andrew S. Tanenbaum Eric S. Raymond Steve Jobs Richard M. Stallman William Gates III Joseph Weizenbaum Tim Berners-Lee...

Different Disciplines

Engineering



Science

Speaking Different Languages – Example I

SOFTWARE ENGINEER: Shall I use C++ or Lisp?
COMPUTER SCIENTIST: Does not matter!
SOFTWARE ENGINEER: ?
COMPUTER SCIENTIST: Aren't both Turing complete?
SOFTWARE ENGINEER: ?
COMPUTER SCIENTIST: Sigh, I guess they both support loops or recursion, conditional statements and the like?
SOFTWARE ENGINEER: Well, yes, of course!
COMPUTER SCIENTIST: Then it does not matter!
SOFTWARE ENGINEER: WTF?!

Speaking Different Languages – Example II

SOFTWARE ENGINEER: Haskell does not have side effects?
COMPUTER SCIENTIST: Haskell is a pure functional language!
SOFTWARE ENGINEER: ?
COMPUTER SCIENTIST: Sigh, a pure functional language does not have *any* side effects, so it was designed!
SOFTWARE ENGINEER: So I cannot change state?
COMPUTER SCIENTIST: You do not understand - there is no state!
SOFTWARE ENGINEER: So, please – how do I implement input/output?
COMPUTER SCIENTIST: You have to use *monads*!
SOFTWARE ENGINEER: For goodness' sake, what is a *monad*?
COMPUTER SCIENTIST: A monad is a monoid in the category of endofunctors, what's the problem?
SOFTWARE ENGINEER: WTF?!

We teach computer science, but the industry desperately needs (software) engineers [Par97]. The effect is roughly the same as it would be if you assigned a physics Ph.D. to design electrical equipment!

What can we Compute?

Two important questions:

1. "What does it mean for a function from the natural numbers to themselves to be computable?"
2. "Can noncomputable functions be classified into a hierarchy based on their level of noncomputability?"

Related to the Church–Turing thesis, which states that any function that is computable by an algorithm is a computable function.

Big open question:

$$P = NP?$$

Further aspects:

> "Can (quantum) physics be (efficiently) simulated by (classical) computers?" Richard Feynman (1981)

This led to theory of quantum computers.

How to deal with Complexity?

Politically correct answer:

Software Engineering!

The truth is:

We still have no clue!

Outlook

90 / 96

The Future of Computing

The best way to predict the future is to invent it!

Alan Kay

References

[Neu45] John von Neumann. First draft of a report on the edvac. *IEEE Ann. Hist. Comput.*, 15(4):27–75, October 1945.

[Par97] David Lorge Parnas. Software engineering (extended abstract): an unconsummated marriage. In *Proceedings of the 6th European SOFTWARE ENGINEERING conference held jointly with the 5th ACM SIGSOFT international symposium on Foundations of software engineering*, ESEC '97/FSE-5, pages 1–3, New York, NY, USA, 1997. Springer-Verlag New York, Inc.

[RT74] D. M. Ritchie and K. Thompson. The unix time-sharing system. *Communications of the ACM*, 17:365–375, 1974.

[Sho97] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26:1484–1509, October 1997.

[Tur01] Alan M. Turing. *Collected works of A. M. Turing. Mathematical logic*. Elsevier, Amsterdam, The Netherlands, 2001. Edited by R. O. Gandy and C. E. M. Yates. Including prefaces by Solomon Feferman.

[Wit22] Ludwig Wittgenstein. *Tractatus Logico-Philosophicus*. Kegan Paul, Trench, Trubner & Co., 1922.

Credits – I/III

- ☐ **Stonehenge**: ©2009 David Ball – www.davidball.net
- ☐ **Abacus**: public domain
- ☐ **Al-Chwarizmi**: John L. Esposito. The Oxford History of Islam. Oxford University Press, copyright expired
- ☐ **Da Vinci**: ©IBM
- ☐ **Pascaline**: CC 2005 David Monniaux
- ☐ **Leibniz**: Page of the Article "Explication de l'Arithmétique Binaire", 1703/1705, copyright expired
- ☐ **Babbage Diff. Engine**: CC 2005 Carsten Ullrich
- ☐ **Babbage Ana. Engine**: CC 2009 Bruno Barral
- ☐ **Ada Lovelace**: Margaret Carpenter (1793-1872), copyright expired
- ☐ **Steampunk**: ©2007 Jake von Slatt
- ☐ **Power (Jacquard) Looms**: ©2009 Lutz Marx
- ☐ **IBM**: public domain
- ☐ **Zuse**: CC ComputerGeek, public domain
- ☐ **Alan Turing**: ©1951 National Portrait Gallery, London, http://www.npg.org.uk/
- ☐ **Von Neumann**: public domain
- ☐ **Von Neumann Architektur**: CC 2007 Lukas Grossar

Credits – II/III

- Grace Hopper: public domain
- Bug: public domain
- Sliderule: Creative Commons Attribution-Share Alike 3.0 Unported
- Lunar Module: public domain
- Moore's Law: Creative Commons Attribution-Share Alike 3.0 Unported
- Unix: Creative Commons Attribution-Share Alike 3.0 Unported
- MacIntosh: Creative Commons Attribution-Share Alike 2.5 Italy http://www.allaboutapple.com/
- PC: Creative Commons Attribution-Share Alike 3.0 Unported
- Android: Creative Commons Attribution-Share Alike 3.0
- Quantum Computing: Smite-Meister, Creative Commons Attribution-Share Alike 3.0 Unported
- Programming Languages ©1999-2010 Éric Lévénez http://www.levenez.com/lang/
- Bletchley Park: Matt Crypto, public domain
- 1950s: 2006 Hans Weingartz http://www.pass-weingartz.de/hw.htm CC-by-sa 2.0/de
- Fortran and all further programming examples taken from http://rosettacode.org/wiki/Sorting_algorithms/Quicksort
- Twiggy: ©2001-2005 www.twiggylawson.co.uk
- Apollo 11: public domain

Credits – III/III

- Thompson Ritchie: public domain
- Berlin Wall: Unknown photographer, Creative Commons-Lizenz Namensnennung-Weitergabe unter gleichen Bedingungen 3.0 Unported
- Bill Clinton, Yitzhak Rabin, Yasser Arafat at the White House: public domain
- Clouds: ©2008 Jörg Schäfer
- Languages: Source http://i.imgur.com/1gF1j.jpg, copyright unknown
- MapReduce: ©2004, Jeffrey Dean and Sanjay Ghemawat, Google Inc.
- Principles Programming Paradigms: ©2007 Peter Van Roy http://www.info.ucl.ac.be/~pvr/paradigmsDIAGRAMeng.pdf
- Engineers: http://s882.photobucket.com/albums/ac30/Pvt_Hawkins/
- Einstein: ©1921, Ferdinand Schmutzer, copyright expired