io9.com

Alan Turing

23/6/12 – 7/6/54

Computability

The Church-Turing thesis states that "*everything algorithmically computable is computable by a Turing machine.*"

Cryptanalysis

Turing was instrumental in the British code breaking effort that cracked the German Enigma code during WWII. The impact of this effort was so great that it is said to have changed the outcome of WWII.

Turing Test and AI

Turing was a pioneer of the problem of artificial intelligence, and posed the Turing Test.

For things that we have to learn to do, we learn by doing them. Aristotle

MIT

Margaret Hamilton

1936 –

Lead Developer, Apollo Flight Software

Hamilton was Director o the Software Engineering Division of the MIT Instrumentation Laboratory, which developed on-board flight software for the Apollo space program. This is the software the controlled the spacecraft on their passage to the moon. Her team also worked on the flight software for Skylab.

Software Engineering

Credit as one of three to have coined the term ‘software engineering’, in order that software development be taken seriously (during the Apollo mission).

Presidential Medal of Freedom 2016

For things that we have to learn to do, we learn by doing them. Aristotle

euro-coins.info

Konrad Zuse

22/6/10 – 18/12/95

Z3
Zuse designed and built the world's first Turing-complete computer, May 1941.

Plankalkül
Realizing how impractical it was to write programs at a low level, Zuse developed the world's first high-level programming language (1941-1945). Unfortunately, the impact of this language was greatly reduced by the virtue of Zuse having made these developments in wartime Germany.

For things that we have to learn to do, we learn by doing them. Aristotle

U.S. Army

Grace Hopper

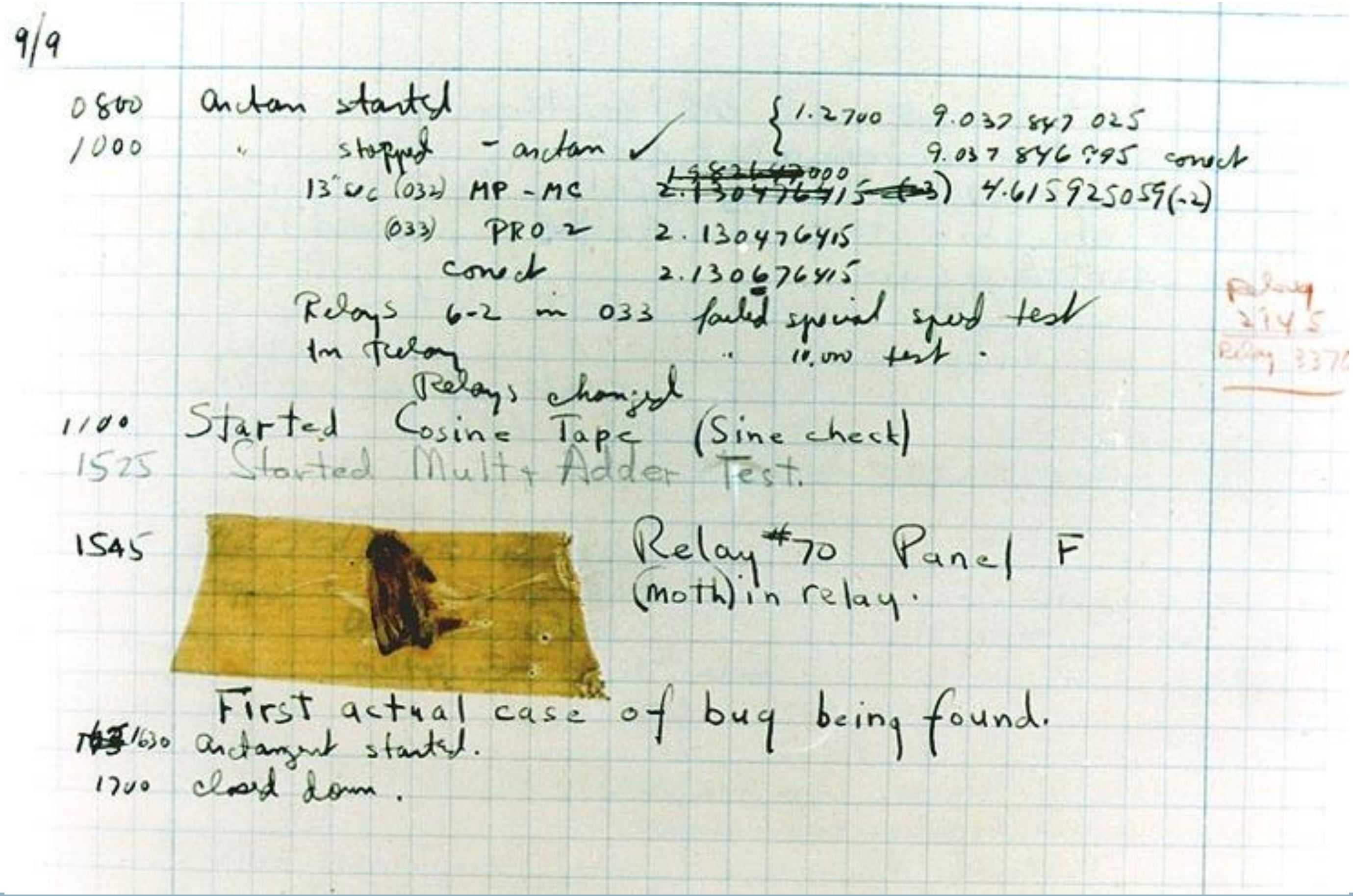
9/12/06 – 1/1/92

First Compiler
Hopper developed the first compiler for a programming language (targeting the Harvard Mark I computer).

COBOL
She led the development of COBOL through her ideas of machine-independent programming languages that were closer to human languages than to machine code.

Bugs
Hopper is said to have coined the term '*bug*' after finding a moth stuck in an electrical relay which had caused the computer to malfunction.

For things that we have to learn to do, we learn by doing them. Aristotle


9/9
0800 arctan started
1000 stopped - arctan
1.2700 9.037 847 025
9.037 846 995 correct
(033) PRO 2 2.130476415
correct 2.130676415
Relays 6-2 in 033 failed special speed test
Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Mult+ Adder Test.
1545 Relay #70 Panel F
(moth) in relay.
First actual case of bug being found.
1630 arctangent started.
1700 closed down.

io9.com

John von Neumann

28/12/03 – 8/2/57

The von Neumann Architecture

In 1945 Neumann described the first computer in which the program and data were stored in the same address space, based on Eckert and Mauchly's work on the EDVAC.

Random Numbers

Von Neumann co-developed the Monte Carlo method and pioneered the generation of pseudorandom numbers.

Merge Sort

Knuth credits von Neumann with developing the merge sort algorithm in 1945.

For things that we have to learn to do, we learn by doing them. Aristotle

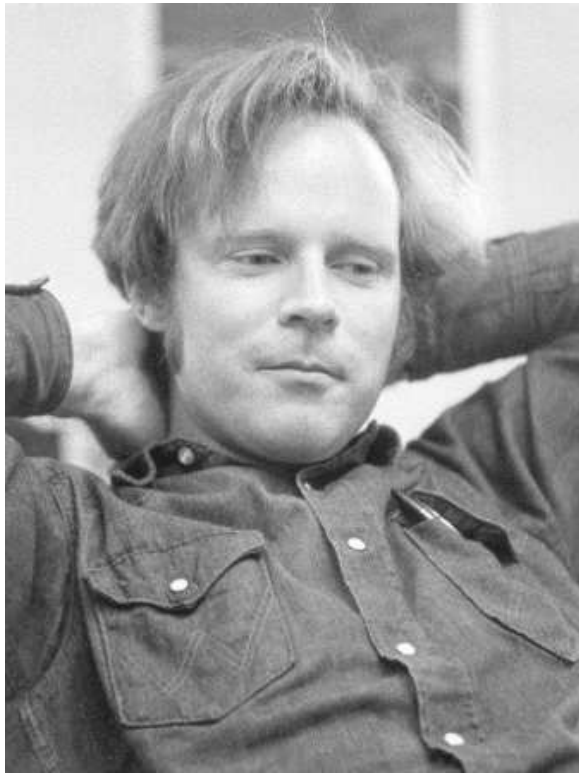
Carnegie Mellon University

Jeanette Wing

Professor, Columbia University
Director of Data Sciences Institute
Microsoft VP, MSR International
Since 2012, Wing has been vice president at Microsoft, head of Microsoft Research International.
CMU CSD Head
From 2010-2012, Wing was head of computer science at Carnegie Mellon University. She was also head from 2004-2007.
NSF Assistant Director
From 2007-2010, Wing was at the NSF as Assistant Director, responsible for Computer Information Science and Engineering (CISE).

Famous for:
- Formal methods
- OO programming
- computational thinking

For things that we have to learn to do, we learn by doing them. Aristotle



IEEE Computer Society

Robert Floyd

8/6/36 – 25/9/2001

Program correctness
Floyd was a pioneer of the idea of applying mathematics to the problem of program correctness. His work contributed significantly Hoare logic

Software Engineering
Floyd is said to have been one of the first advocates of refactoring and the rewriting of working programs from scratch.

Turing Award 1978

For things that we have to learn to do, we learn by doing them. Aristotle

Fran Allen

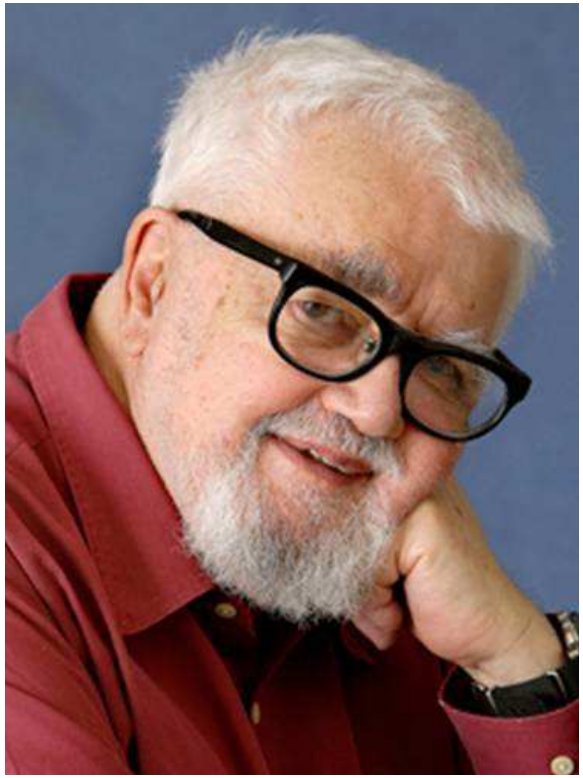
4/8/32 – 4/8/2020

Optimizing Compilers
Allen pioneered the field of optimizing compilers. She introduced the application of graph theory to program optimization.

Parallelizing Compilers
Allen pioneered the development of automatic parallelizing compilers, with the development of PTRAN, a parallel compiler for FORTRAN.

Turing Award 2006

For things that we have to learn to do, we learn by doing them. Aristotle



saildart.org/jcm2012.html

John McCarthy

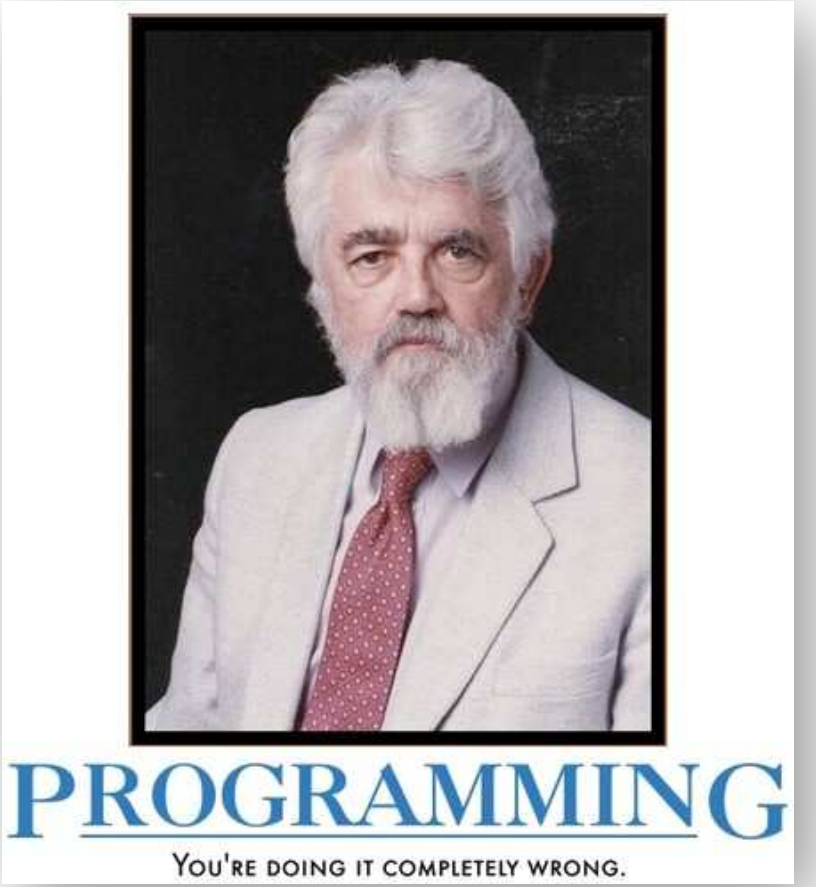
4/9/27 – 24/10/11

Artificial Intelligence
McCarthy was a pioneer of AI, and coined the term ‘artificial intelligence’.

ALGOL and Lisp
McCarthy worked on the committee that developed ALGOL. He then developed Lisp, the language of choice for AI.

Garbage Collection
McCarthy invented garbage collection (automatic memory management), published in CACM in 1960.

Turing Award 1971


PROGRAMMING
YOU'RE DOING IT COMPLETELY WRONG.

ACM

Barbara Liskov

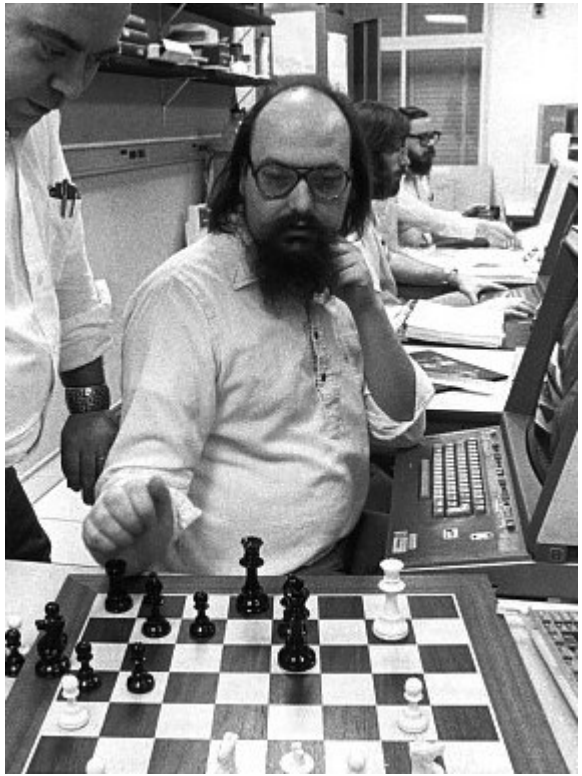
7/11/39 –

CLU
Liskov and her students at MIT developed CLU in the 1970's. CLU extended ALGOL with data types that had code that operated on them; an important step in the development of object-oriented languages.

Argus
Liskov lead the development of Argus in the 1980's. Argus was the first high-level language to support distributed programs.

Turing Award 2008

For things that we have to learn to do, we learn by doing them. Aristotle



Bell Labs

Ken Thompson

4/2/43 –

Multics and Unix

At Bell Labs, Thompson worked on Multics and upon leaving the project was instrumental in creating Unix with Dennis Ritchie.

B

Thompson created the B programming language, a typeless imperative programming language which was the precursor to C.

Regular Expressions, UTF-8

Thompson created an editor which included regular expressions. He also developed the UTF-8 character set.

Turing Award 1983

For things that we have to learn to do, we learn by doing them. Aristotle

computerhistory.org

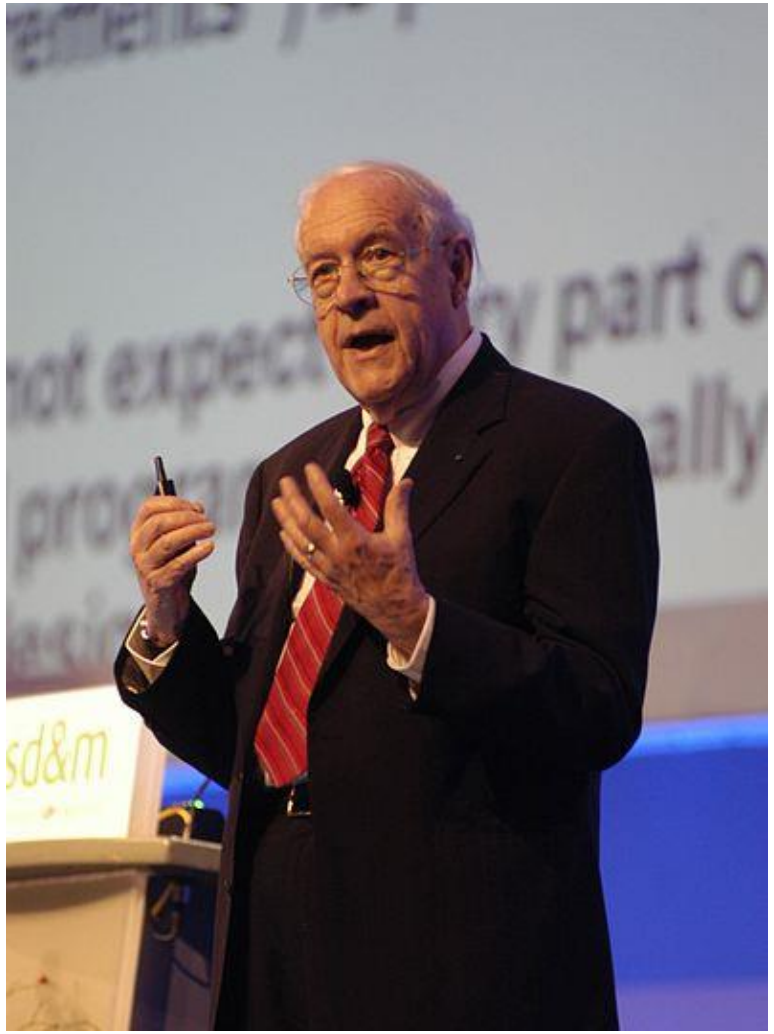
Ada Lovelace

10/12/1815 – 27/11/1852

Analytical Engine

Lovelace is credited with writing the first algorithm designed for processing by a machine, and as such is often regarded as the first computer programmer. The algorithm allowed the Analytical Engine to calculate Bernoulli numbers. Lovelace corresponded with Charles Babbage and was fascinated by his Difference Engine. One of her ambitions was to create a model for the brain “a calculus for the nervous system”.

For things that we have to learn to do, we learn by doing them. Aristotle



www.sdm.de

Fred Brooks

19/4/31 –

The Mythical Man Month

Brooks' reflections on the development of IBM's OS/360, and the errors he made as manager of that large project. Brooks called the book "*the Bible of Software Engineering*", because "*everybody quotes it, some people read it, and a few people go by it.*"

"Adding manpower to a late software project makes it later"

No Silver Bullet

The *essential* complexity of software suggests that there will be no once-and-for-all fix to the problem of efficient software construction.

"… this irreducible essence of modern software systems: complexity, conformity, changeability, and invisibility."

Turing Award 1999

For things that we have to learn to do, we learn by doing them. Aristotle

MIT

Shafi Goldwasser

1958 –

Zero-knowledge Proofs

Used in cryptography to prove the correctness of a statement without revealing anything other than the correctness of the statement.

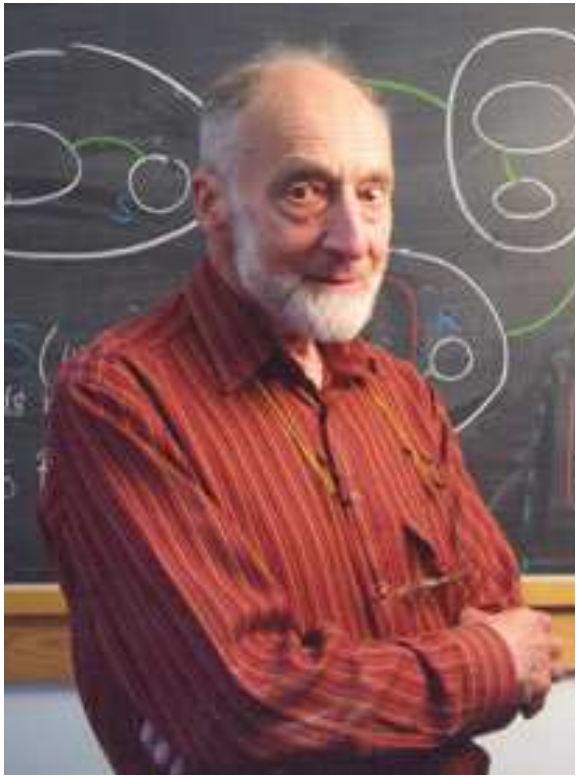
Other Cryptography

Goldwasser has made many other contributions to cryptography, was awarded the RSA Award in Mathematics, and has the RSA Professorship at MIT

Gödel Prize 1993, 2001
Turing Award, 2012

Source:Dake, https://en.wikipedia.org/wiki/Zero-knowledge_proof

For things that we have to learn to do, we learn by doing them. Aristotle



ACM

Robin Milner

13/1/34 – 20/3/10

Automated Theorem Proving

Milner lead the development of LCF, one of the first human-assisted theorem provers.

ML

Milner developed ML (*metalanguage*), an early non-pure functional programming language. ML included the first parametric polymorphisim (father of today's generics).

Pi-Calculus

Milner was one of the developers of the pi-calculus, a mathematical formalism for describing the properties of concurrent computation.

Turing Award 1991

For things that we have to learn to do, we learn by doing them. Aristotle

salon

Anita Borg

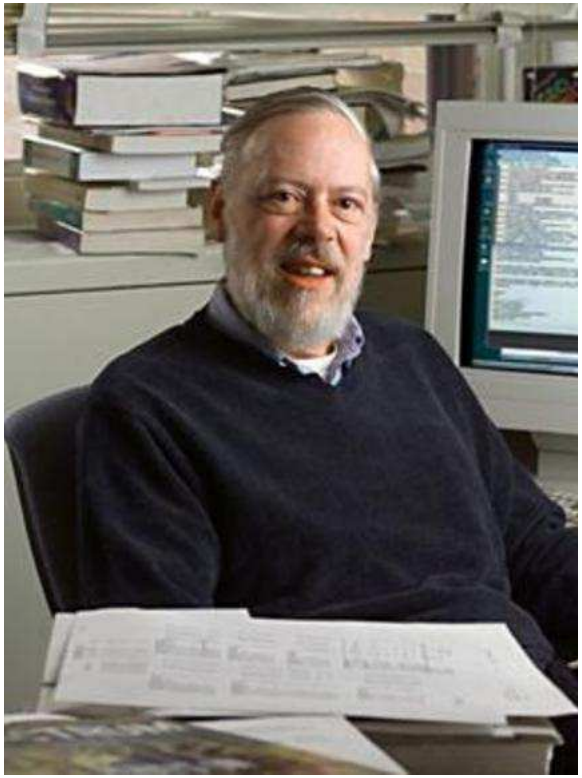
17/1/49 – 6/4/2003

Borg worked for DEC and XEROX PARC, PhD from NYU, focus on operating systems and memory performance.

Advocacy for Women in Technology

Borg founded the Institute for Women and Technology, the Grace Hopper Celebration of Women in Computing, and the Systers email network. Strong legacy today, including multiple awards named in her honor.

For things that we have to learn to do, we learn by doing them. Aristotle



io9.com

Dennis Ritchie

9/9/41 – 10/12/11

Unix

With Ken Thompson, Ritchie was instrumental in the development of the Unix operating system.

"UNIX is very simple, it just needs a genius to understand its simplicity" – Dennis Ritchie

C

Ritchie created the C programming language to use with the Unix operating system (it followed Ken Thompson's B language).

"[C has] the power of assembly language and the convenience of … assembly language." – DR

Turing Award 1983

For things that we have to learn to do, we learn by doing them. Aristotle

salon

Anita Borg

17/1/49 – 6/4/2003

Borg worked for DEC and XEROX PARC, PhD from NYU, focus on operating systems and memory performance.

Advocacy for Women in Technology

Borg founded the Institute for Women and Technology, the Grace Hopper Celebration of Women in Computing, and the Systers email network. Strong legacy today, including multiple awards named in her honor.

For things that we have to learn to do, we learn by doing them. Aristotle

Robert McClure

Niklaus Wirth

15/2/33 –

Programming Languages
Chief designer of Euler, Algol W, Pascal, Modula, Modula-2, Oberon, Oberon-2 and Oberon-07

Wirth's Law
'Software is getting slower more rapidly than hardware becomes faster' (Reiser)

Extended Backus-Naur Form Wirth developed the first EBNF. Now an ISO standard (ISO/IEC 14977).

'Reliable and transparent programs are usually not in the interest of the designer.'

Turing Award 1984

For things that we have to learn to do, we learn by doing them. Aristotle

Esquire

Carol Bartz

28/8/48 –

CEO Autodesk

From 1992-2006 Bartz was CEO of Autodesk. Credited with transforming the company into the leading vendor of design software.

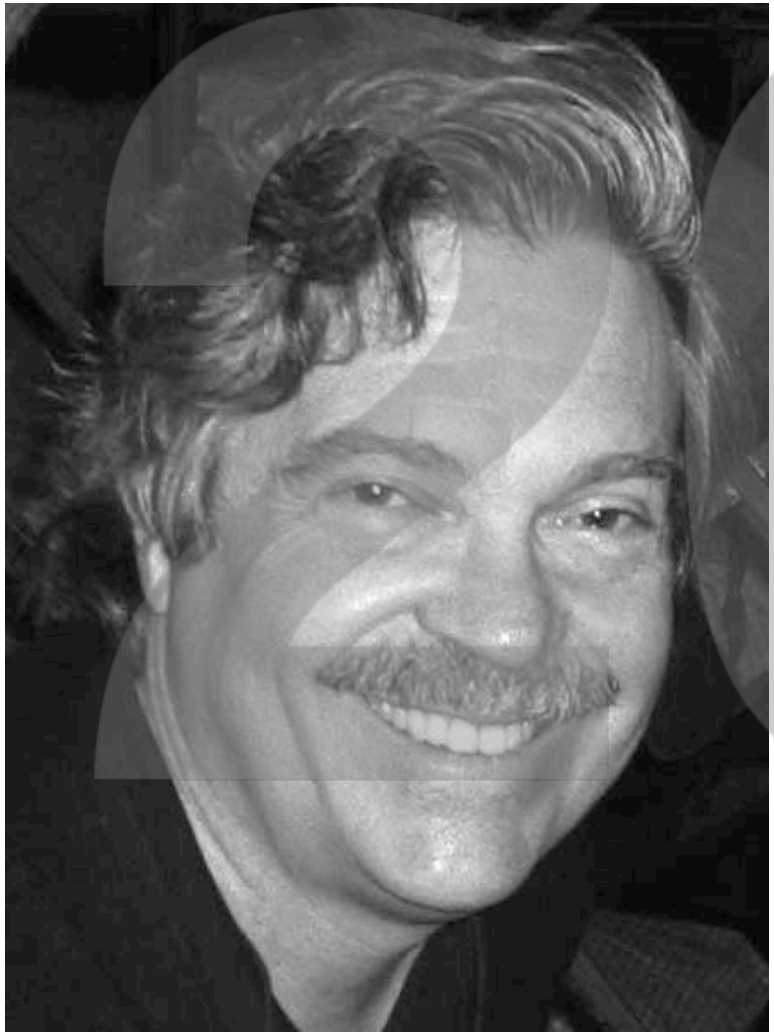
CEO Yahoo!

From 2009-2011 was CEO of Yahoo! She was tasked with turning around a failing company.

Computer Science degree from University of Wisconsin, Madison

"I have a belief that life isn't about balance, because balance is perfection ... Rather, it's about catching the ball before it hits the floor."

For things that we have to learn to do, we learn by doing them. Aristotle



vpri.org

Alan Kay

17/5/40 –

Smalltalk

Kay lead the development of Smalltalk while at Xerox PARC. Smalltalk remains an important example of object oriented programming.

Apple Macintosh

Kay's work at Xerox PARC on networked workstations and graphical user interfaces were commercialized by Apple Computer and lead to the development of the Lisa and Macintosh computers.

"The best way to predict the future is to invent it."

- Alan Kay

Turing Award 2003

For things that we have to learn to do, we learn by doing them. Aristotle

Goldberg B22

mac-history.net

22/7/45 –

Smalltalk-80

Goldberg managed Xerox PARC's System Concepts Laboratory, who developed Smalltalk-80 (lead by Alan Kay). This introduced the key ideas of object-oriented programming. The system also had a user interface based on windows, which lead to mainstream modern user interface design.

Co-Founded ParcPlace

Goldberg co-founded ParcPlace, a company that sold Smalltalk systems.

For things that we have to learn to do, we learn by doing them. Aristotle