A.M. TURING AWARD

SPOTLIGHT on LAUREATES

Spotlight on ACM A.M. Turing Laureates

The ACM A.M. Turing Award, computing's most prestigious honor, acknowledges individuals who have made lasting and major contributions to the field. Here, we highlight the remarkable accomplishments of these men and women, as well as the far-reaching impact of their work. These Spotlights on ACM A.M. Turing Laureates are arranged in reverse chronological order, from 2016 to 1966.

More information about the Turing Laureates can be found at http://amturing.acm.org.

SIR TIM BERNERS-LEE
Inventor of the World Wide Web, the first web browser, and the protocols and algorithms allowing the Web to scale
acm
A.M. TURING AWARD
2016

Sir Tim Berners-Lee

Sir Tim Berners-Lee's vision was instrumental in the development and implementation of what is perhaps the most ubiquitous achievement in the entire history of human communication: The World Wide Web. Sir Tim envisioned a world connected by a global information space where documents and other resources could be identified with uniform names. To this end, he introduced a universal resource naming scheme based on URIs and URLs, the HTTP protocol, and the HTML markup language, as well as the first web browser and web serving software. Sir Tim also had the foresight to recognize the importance of creating this entire infrastructure in an open-source fashion, which catalyzed the Web's further development.

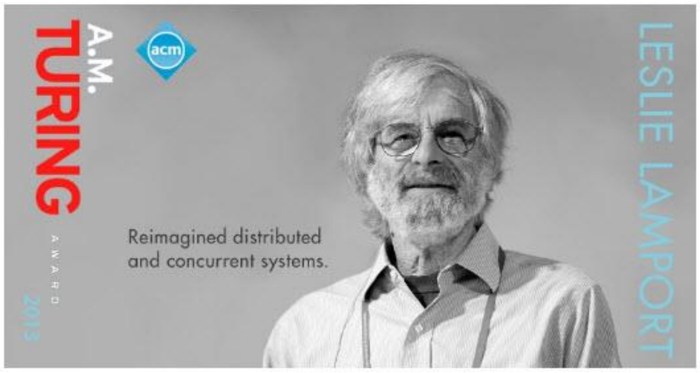
WHITFIELD DIFFIE & MARTIN HELLMAN
Invented public-key cryptography
acm
A.M. TURING AWARD
2015

Whitfield Diffie and Martin Hellman

In the fall of 1974, Whitfield Diffie and Martin Hellman met for what was intended to be a short meeting. What transpired was a rich discussion that lasted long into the night, and the forging of an intellectual partnership that would yield groundbreaking insights on which the modern field of cryptography still rests. In 1976, Diffie and Hellman published the revolutionary paper *New Directions in Cryptography* in *Communication of the ACM*, which laid the framework for the concepts of public-key cryptography and digital signatures, which today are the most widely used security protocols on the Internet, protecting trillions of dollars of web-based financial transactions each day.

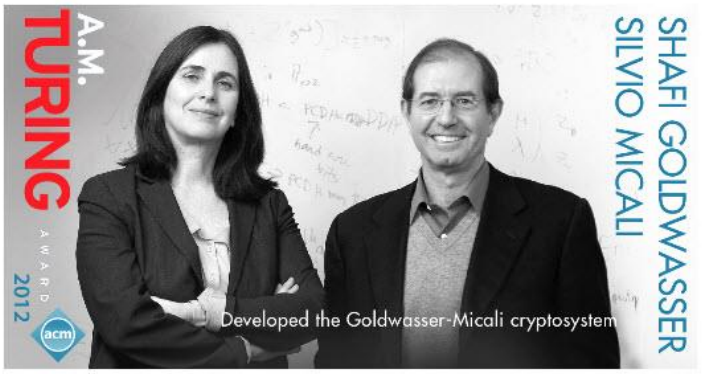
Michael Stonebraker

Edgar F. Codd's seminal papers on the relational model inspired 2014 ACM A.M. Turing Award recipient Michael Stonebraker to work on an efficient and practical implementation of the theories Codd laid out. The result, INGRES, built a new commercial market of relational database systems over the next decade. The relational database management system is one of computing's most important and widely used technologies, having replaced filing cabinets as the standard way of storing and retrieving information. Stonebraker's blend of technical brilliance with a keen sense of entrepreneurship makes him a one-of-a-kind technologist.



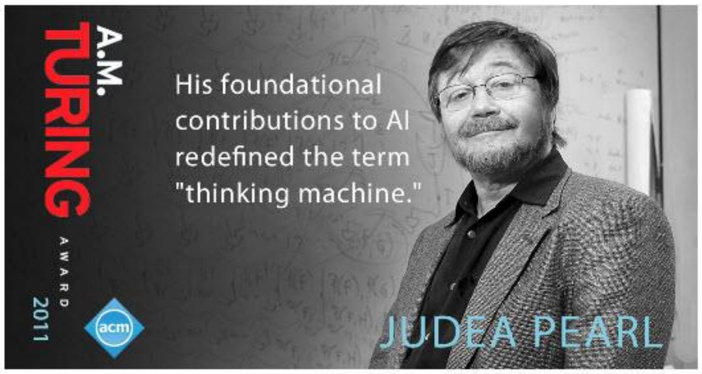
Leslie Lamport

Leslie Lamport's works shed light on fundamental issues of concurrent programs, for which there was no formal theory at the time. He grappled with fundamental concepts such as causality and logical time, atomic and regular shared registers, sequential consistency, state machine replication, Byzantine agreement and wait-freedom. Lamport developed a substantial body of work on the formal specification and verification of concurrent system, and has contributed to the development of automated tools applying these methods. For these and other seminal contributions to theory and practice of distributed and concurrent systems, Lamport received the 2013 ACM A.M. Turing Award.



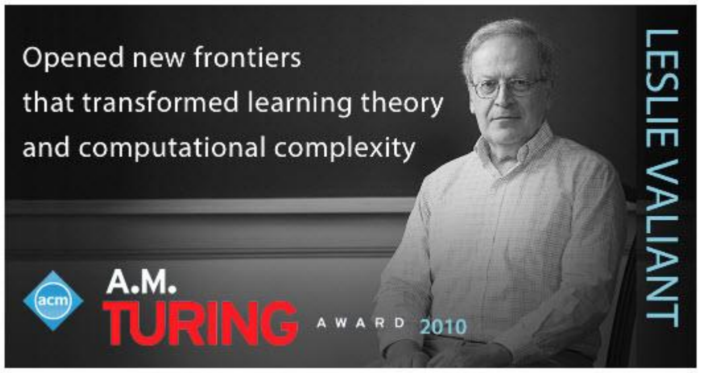
Shafi Goldwasser & Silvio Micali

Shafi Goldwasser and Silvio Micali received the 2012 ACM AM Turing Award for their work that helped lay the foundation of the science of #cryptography. Goldwasser and Micali met at the University of California, Berkeley, in the 1980s amidst an atmosphere of rapid progress in computing. Together, they explored how number theory might be applied to the concept of data privacy, and in 1982 they co-authored the paper "Probabilistic Encryption," the premise of which remains central to the theoretical framework of encryption. The partnership between these two remarkable innovators led to conceptual breakthroughs that continue to inform one of the most rapidly advancing areas of computing today.


A.M. TURING AWARD 2011
acm
His foundational contributions to AI redefined the term "thinking machine."
JUDEA PEARL

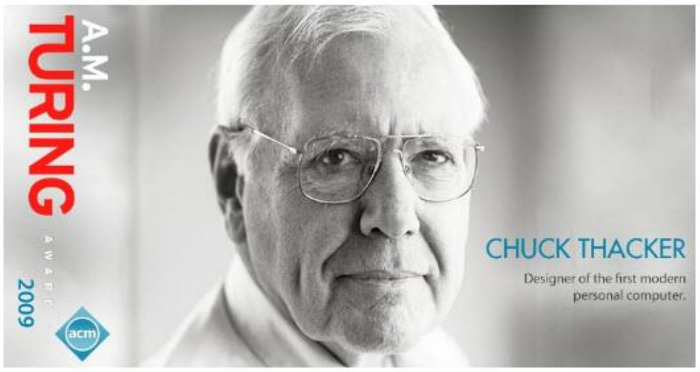
Judea Pearl

Judea Pearl's work enabled machines to process information under uncertainty and assesses cause-effect relationships from empirical findings, which are the foundational characteristics of artificially intelligent machines. His early work on heuristic searches—a trial-and-error method of problem-solving—propelled the evolution of AI into a mature field with sound scientific foundations. He then went on to develop the theoretical foundations for reasoning under uncertainty using a "Bayesian network," an extremely general and flexible modeling tool that mimics the neural activities of the human brain. The networks also changed the way causality is treated in the empirical sciences, which are based on experiment and observation.


Opened new frontiers that transformed learning theory and computational complexity
LESLIE VALIANT
acm
A.M. TURING AWARD 2010

Leslie Valiant

Leslie Valiant's work has been instrumental in leading to advances in artificial intelligence as well as computing practices including natural language processing, handwriting recognition, and computer vision. Valiant made great strides towards fundamentally understanding how the brain "computes" – a breakthrough which helped lay the framework for building modern forms of machine learning and communications like IBM's Watson computing system. Valiant's "Theory of the Learnable," published in 1984 in Communications of the ACM, established the research area known as Computational Learning Theory, which led to the development of algorithms that adapt their behavior in response to feedback from the environment, and had a major impact on mainstream research in AI.



Chuck Thacker

Chuck Thacker's colleagues call him an "engineer's engineer." While working at Xerox PARC alongside fellow ACM A.M. Turing Award recipient Butler Lampson, Thacker was the driving creative force behind 1973's "Alto," the first system recognized as a "personal computer." Thacker moved forward from this success to make other extraordinary contributions to fields including local area networks, multiprocessor workstations, snooping cache coherence protocols, laser printing, and tablet personal computers. In 1997, he joined Microsoft Research to help establish Microsoft Research Cambridge in the UK where he was instrumental to the development of the tablet PC.

Barbara Liskov

In 1971, ACM A.M. Turing Award Recipient Barbara Liskov joined the MIT faculty as a professor in the Laboratory for Computer Science. There, Liskov led the design and implementation of the CLU programming language, which emphasized the notions of modular programming, data abstraction, and polymorphism. These concepts are a foundation of object-oriented programming used in modern computer languages such as Java and C#. Her MIT group also created the Argus language, which extended the ideas of CLU to ease implementation of programs distributed over a network. Later, with Jeannette Wing, Liskov developed a new notion of subtyping known as the Liskov substitution principle, and her contributions in this area have influenced advanced system developments and set a standard for clarity and usefulness. Her subsequent work has covered many aspects of operating systems and computation, including important work on object-oriented database systems, caching, persistence, recovery, fault tolerance, security, and others.

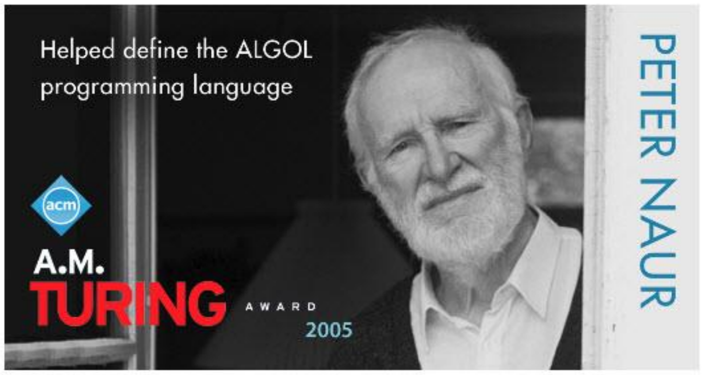
acm
A.M. TURING AWARD 2007
EDMUND M. CLARKE, E. ALLEN EMERSON, JOSEPH SIFAKIS
Model Checking: An Automated Quality Assurance Method

Edmund M. Clarke, E. Allen Emerson and Joseph Sifakis

In 1981,seminal papers by Edmund M. Clarke and E. Allen Emerson working in the USA, and by Joseph Sifakis working independently in France, founded an automatic quality assurance method, dubbed Model Checking. This method provides an algorithmic means of verifying whether or not an abstract model representing a system or design, satisfies a formal specification expressed in temporal logic. The progression of Model Checking technology to the point where it can be successfully used for very complex systems has required the development of sophisticated means of coping with extremely large state spaces, and substantive increases in expressiveness. The laureates work on these topics has further engendered a very large international research community investigating such topics. Consequently, many major hardware and software companies are now using Model Checking in practice. Applications include formal verification of VLSI circuits, communication protocols, and embedded systems.

Frances "Fran" Allen

Frances E. "Fran" Allen's work has led to remarkable advances in compiler design and machine architecture that are at the foundation of modern high-performance computing. Her contributions were foundational to the process of translating a user's problem-solving language statements into efficient sequences of computer instructions. Her work also made great advancements in automatic program parallelization, which enables programs to use multiple processors simultaneously in order to obtain faster results. These theoretical breakthroughs led to computing techniques that are still used in areas including weather forecasting, DNA matching, and national security. Allen became the first woman to receive the ACM A.M. Turing Award in 2007.

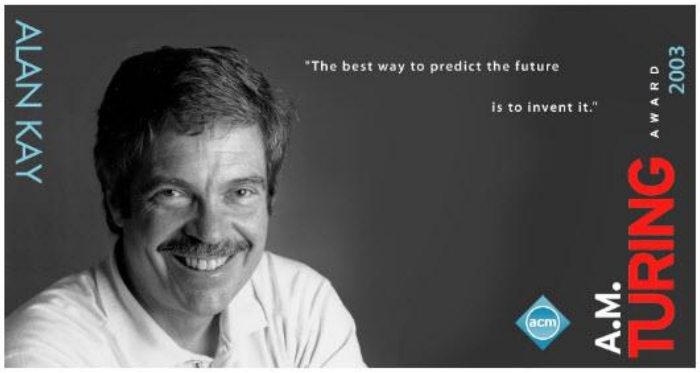


Peter Naur

Peter Naur came to computer science through an intellectually eclectic background that began in astronomy. After earning a degree in astronomy at Copenhagen University, Naur studied computer programming at Cambridge University where he Electronic Delay Storage Automatic Calculator (EDSAC) in order to solve a perturbation problem in astronomy. In the 1950s, Naur moved to the United States, where he would become instrumental to the development of the ALGOL programming language. For this pioneering contributions to the "art and science of computer programming," Naur received the 2005 ACM A.M. Turing Award.

Vint Cerf and Robert Kahn

ACM A.M. Turing Award recipients Vint Cerf and Robert Kahn first met at UCLA in 1969, where they worked on the nascent ARPANET project - the technology that later became the technical foundation of the Internet. In June 1973 Kahn and Cerf invited networking experts from around the world to a seminar to weigh in on their idea of developing a system for interconnecting networks, and following this meeting, Cerf and Kahn published the seminal 1974 paper, "A Protocol for Packet Network Intercommunication" - the paper that laid the groundwork for the TCP/IP protocol and the global spread of the Internet.



Alan Kay

2003 ACM A.M. Turing Award recipient Alan Kay was one of the driving creative forces at XEROX PARC, where he conceived of and developed the Dynabook - a powerful and portable device with a keyboard for entering information. Today's myriad portable computing devices all have roots in Kay's Dynabook, and it is for this that he is sometimes referred to as the "father of personal computers." Kay and his research team also developed SmallTalk, the first dynamic object-oriented programming language.

View Kay's lecture at 2012's ACM Turing Centenary Celebration here.

RON RIVEST, ADI SHAMIR & LEN ADLEMAN
RSA public-key cryptography
A.M. TURING AWARD 2002
acm

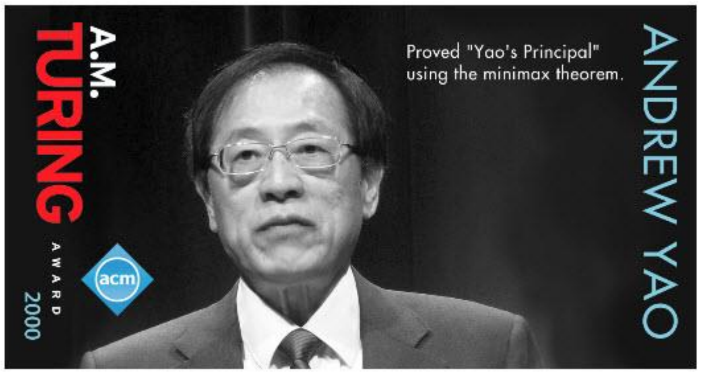
Len Adleman, Ron Rivest, and Adi Shamir

In 1976, Len Adleman, Ron Rivest & Adi Shamir, then graduate students at MIT, read a paper by cryptographers Whitfield Diffie and Martin Hellman that described potential ways to send private messages in which the sender and receiver did not need a shared secret key. Inspired by this idea, the three young computer scientists went to work on developing a way to implement it, and in 1977, they published a paper demonstrating how a message could easily be encoded, sent to a recipient, and decoded with little chance of it being decoded by a third party. The "RSA" method, as it was first called, is now known as Public Key Cryptography, and its practical application cannot be understated; it is now used in almost all Internet-based commercial transactions.

Designed the Simula languages, fundamental to object-oriented programming
KRISTEN NYGAARD & OLE-JOHAN DAHL
A.M. TURING AWARD 2001
acm

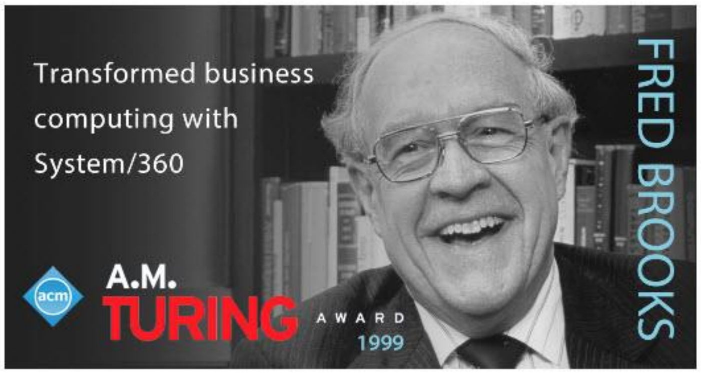
Kristen Nygaard and Ole-Johan Dahl

Ole-Johan Dahl and Kristen Nygaard played an instrumental role in the development of object-oriented programming, the most commonly used framework for programming languages used today. Between 1961 and 1967, Dahl and Nygaard created Simula I and Simula 67 at the Norwegian Computing Center. These two languages allowed developers to create software systems in which each layer relies on a platform implemented by its lower layers, which eventually led the way to a method of programming that is both accessible and available to the entire research community. The object oriented programming model is the basis of the most widely applied programming languages in use today.



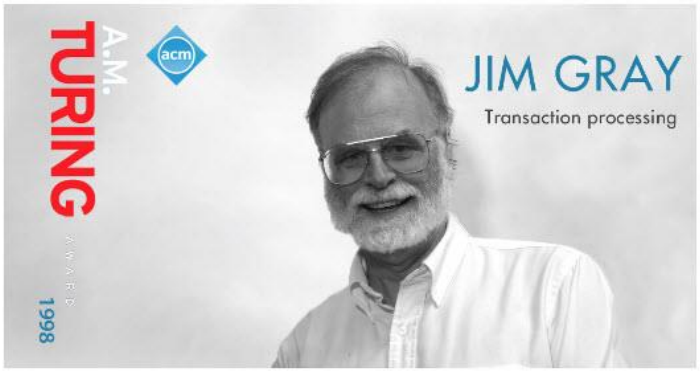
Andrew Yao

2000 ACM A.M. Turing Award recipient Andrew Chi-Chih Yao was born in Shanghai, China in 1946. He received his Ph.D. from Harvard University in 1972 under the supervision of Nobel Laureate Sheldon Glashow. Yao joined the Computer Science Department at Stanford University in 1976, where he made numerous fundamental contributions to the theory of algorithms. Yao introduced Yao's Minimax Principal, a fundamental technique for reasoning about randomized algorithms and complexity. In the 1980s, Yao's interests turned to #cryptograpghy, which continues to be his central research focus. Among many other groundbreaking contributions to this area, Yao introduced a definition of pseudorandom number generator based on computational complexity.



Fred Brooks

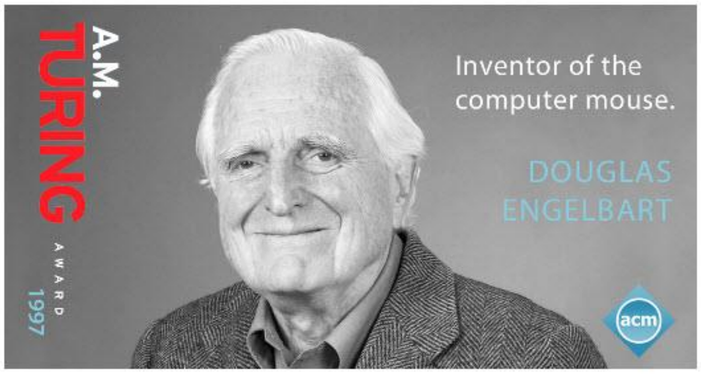
Frederick Brooks joined IBM after earning his PhD. He helped design the IBM 7090 "Stretch" computer, which pioneered a number of advanced techniques including instruction look-ahead, overlapping and pipelining of instruction execution, error checking and correction, and the 8-bit addressable character. He later worked on the IBM 8000 and the System/360, which was announced in 1964. The importance of the System/360 cannot be understated: it was a widely successful computer that transformed the face of business computing and reshaped the landscape of the computer companies throughout the world. For these enduring contributions to the computing field, Brooks received the 1999 ACM A.M. Turing Award.



Jim Gray

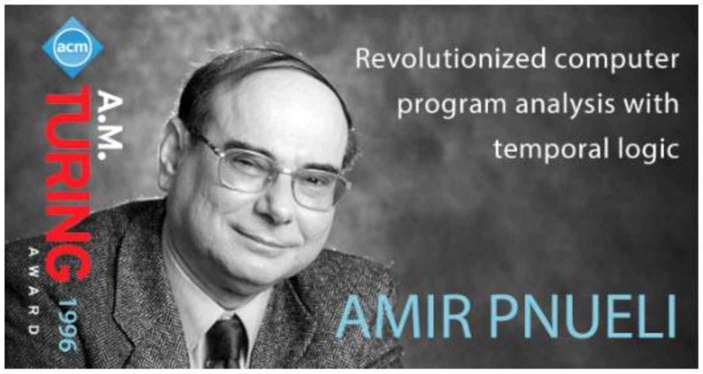
In 1973, Jim Gray joined the newly formed System R project, where he developed techniques that allowed concurrent execution of many transactions, as well as restart after crashes, while maintaining the consistency of the database. This work formed the foundation of his 1998 ACM A.M. Turing Award.

Gray spent much of the 1980s at Tandem Computers, where the scope of his work expanded to extensive involvement in product development and activities involving the entire field of data processing. He played a key role in developing the NonStop SQL relational database management system, designed end user-oriented performance benchmarks, and helped establish the Transaction Processing Performance Council.



Douglas Engelbart

Douglas Engelbart is perhaps best known for inventing the computer mouse, though his many other contributions during the early age of computing were just as significant. Engelbart's experiences during the Second World War solidified his lifelong view of technology as a means for improving the condition of humanity, and of computers as powerful tools that could help solve the world's increasingly challenging problems. Despite his reverence for the power and potential of technology, Engelbart always maintained that the capacity for improving on improvements, or "getting better at getting better," is a uniquely human capability. Engelbart's Law, which encapsulates this principal, is named after him.


Revolutionized computer program analysis with temporal logic
AMIR PNUELI
A.M. TURING AWARD 1996

Amir Pnueli

While working at the department of computer science at Tel Aviv University, Amir Pnueli became deeply involved in logics and deductive methods, and particularly in the work of Arthur Prior. Pnueli became the first to realize the potential implications of applying Prior's "tense logic" to computer programs. Pneuli's 1977 seminal paper, "The Temporal Logic of Programs," revolutionized the way computer programs are analyzed. At the time, practical program verification was widely considered to be hopeless, but Pnueli's paper introduced the notion of reasoning about programs as execution paths, breathing new life into the field of program verification. For this visionary work combining temporal logic with computing science, Pnueli received the 1996 ACM A.M. Turing Award.

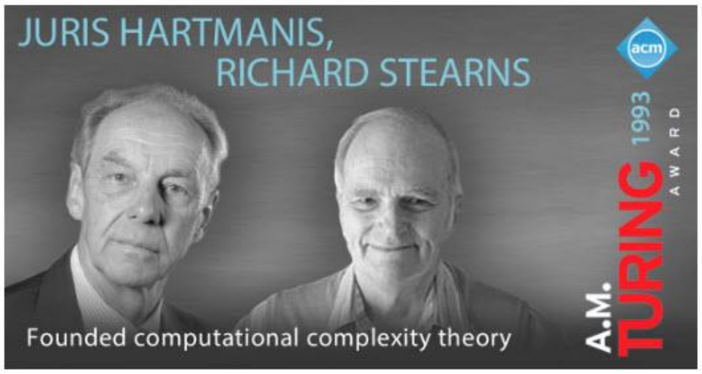
Manuel Blum

In the early 1970s, most students working on complexity sought to either tame the beast or establish that all such efforts are futile. Manuel Blum took a different approach to the subject: he decided to make friends with complexity, to turn the tables on it, and use it to accomplish useful things. In 1986, Blum and Shafi Goldwasser (who received the 2012 ACM Turing Award with Silvio Micali) came up with a public key cryptography which, unlike RSA, had been mathematically proven to be as hard to break as factoring. In the 1990s, along with Luis von Ahn and others, Blum helped develop CAPTCHA , now a ubiquitous device used to determine whether a webpage visitor is human.

ED FEIGENBAUM & RAJ REDDY
Defined and demonstrated the significance of artificial intelligence.
A.M. TURING AWARD 1994

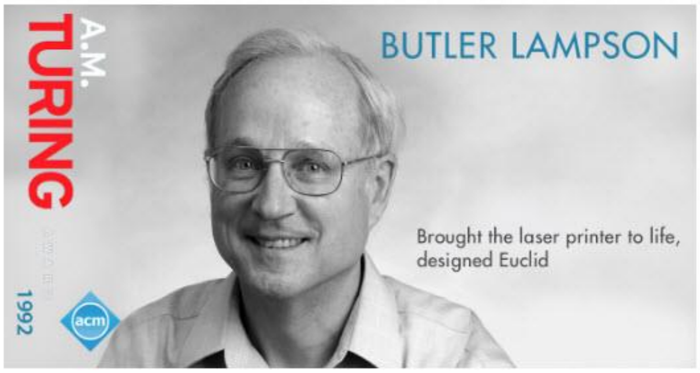
Ed Feigenbaum and Raj Reddy

Ed Feigenbaum and Raj Reddy's concurrent work in artificial intelligence has been instrumental in both defining the emerging field of applied artificial intelligence and in demonstrating its technological significance. Feigenbaum's work developing DENDRAL, a computer program that could guess the geometrical structure of complex chemical compounds, was an early step in defining the enormous potential of endowing computers with "knowledge" rather than imbuing them with "reasoning methods" -- a crucial pivot in AI research. Raj Reddy developed groundbreaking continuous-speech recognition systems which laid the framework for most speech recognition technologies in use today. He later developed voice control of robots, large-vocabulary connected speech recognition, speaker-independent speech recognition, and unrestricted-vocabulary dictation that went on to become historic demonstrations of AI's potential. Many of these ideas now underlie modern commercial speech-recognition products.


JURIS HARTMANIS, RICHARD STEARNS
Founded computational complexity theory
A.M. TURING AWARD 1993

Juris Hartmanis and Richard Stearns

The collaborative work of Juris Hartmanis and Richard Stearns explored how much time and memory are needed to perform different computations -- a field they named "computational complexity." Their seminal 1965 paper, "On the Computational Complexity of Algorithms," used a Turing Machine in an analysis of how to measure the resources needed when algorithms run on specific machines. In contrast to other approaches to complexity theory, this analysis measured the resources that algorithms use when run on specific machines. This work provided the basis for the development of complexity theory as it is now studied, and helped establish computer science as a formal discipline distinct from mathematics, physics and electrical engineering. It also expanded Turing's work in defining the limitations of "what is computable" to a model for "what is efficiently computable."



Butler Lampson

In 1969, Butler Lampson joined the Xerox Palo Alto Research Center (PARC), where he became one of the core members of its Computer Science Laboratory (CSL). In his time at PARC, Lampson was instrumental in the development of the Smalltalk and Mesa programming languages and the Alto and Dorado personal computers, among other innovative feats to come out of the laboratory.

Throughout his illustrious career, Lampson has made contributions to computer architecture, local area networks, raster printers, page description languages, operating systems, remote procedure call, programming languages and their semantics, programming-in-the-large, fault-tolerant computing, transaction processing, computer security, WYSIWYG editors, and tablet computers.

Made key breakthroughs in disparate areas: proofs, languages, concurrency
A.M.
TURING 1991
AWARD
ROBIN MILNER

Robin Milner

At the onset of his career, Robin Milner's work alongside Whitfield Diffie, Richard Weyhrauch, and Malcolm Newey helped lead to the mechanization of Dana Scott's Logic of Computable Functions, representing the first tool for machine assisted proof construction that is both theoretically based and practical. His later development of the ML programming language merged polymorphic type inference together with a type-safe exception-handling mechanism. Later, in 1980, Milner published the book Calculus of Communicating Systems, which lays out the general theory of concurrency. Milner received the 1991 ACM Turing Award for these three distinct achievements, each of which has had a marked, important, and widespread impact.

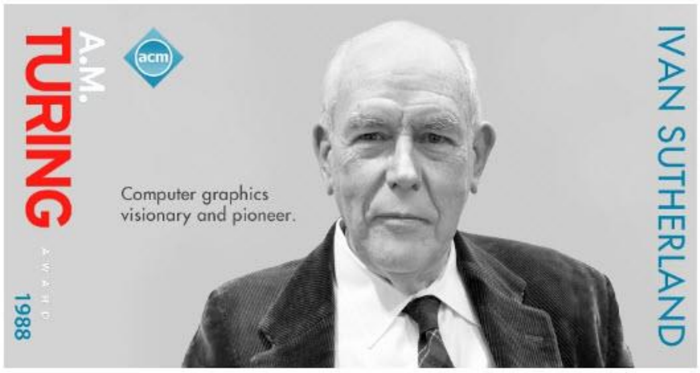
FERNANDO CORBATO
Laid the framework for Compatible Time-Sharing System and Multics
A.M.
TURING 1990
AWARD

Fernando J. "Corby" Corbato

The concept of "time-sharing," in which a computer would switch rapidly between numerous programs, was proposed by several scholars at the end of the 1950s as a way to meet access management challenges. In 1961, Fernando J. "Corby" Corbato built the initial version of the Compatible Time-Sharing System (CTSS) and demonstrated it in November of that year. Later, working in an interdepartmental initiative at MIT known as Project MAC, Corbato was instrumental in developing a second-generation replacement for CTSS called Multics: Multiplexed Information and Computing Services. This went on to become a prototype of a "computer utility" that provides computing and storage service to a large user community.

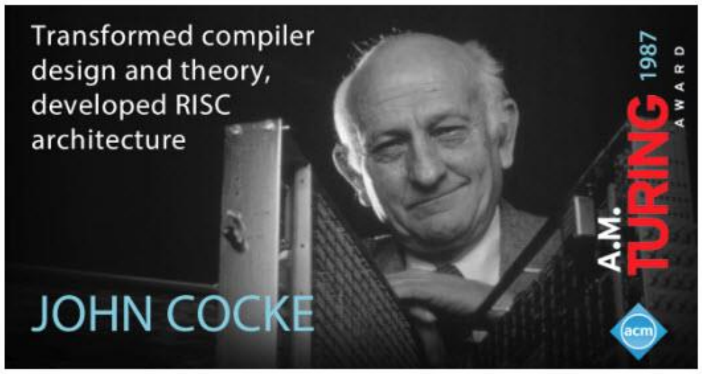
William Kahan

William Kahan is recognized as a researcher and theorist of exceptional talent within numerical analysis – a field that lies at the intersection of computing and applied mathematics. Kahan contributed to several widely used algorithms, including the Golub-Kahan variant of the QR algorithm for singular value decomposition and a Compensated Summation algorithm to offset rounding errors in trajectory computations. Kahan's work at UC Berkeley involved the creation of practical tools as well as papers, algorithms and theorems. Together with his students he produced the widely used fdlibm (Freely Distributable LIBM) mathematics library distributed with BSD Unix and used for machines supporting the new IEEE 754 floating-point standard.



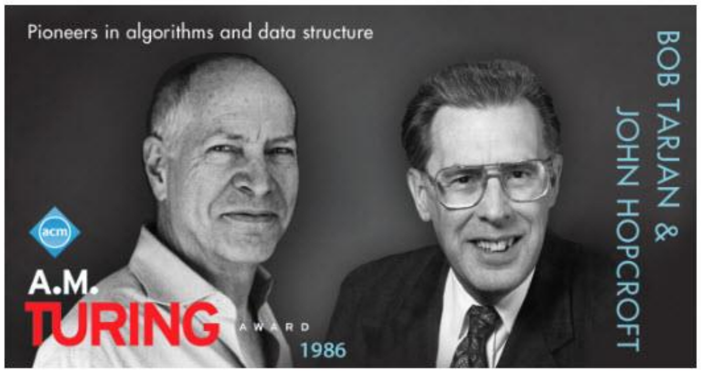
Ivan Sutherland

For his Ph.D thesis at MIT, Ivan Sutherland developed and described "Sketchpad," the first computer graphical user interface. The primitive TX-2 computer Sutherland used to develop Sketchpad ran "batches" of jobs and had virtually no software, but it did have a light pen which Sutherland used in a revolutionary way: to allow the user to draw directly on the computer display. In addition to being the first example of a true GUI, Sketchpad's underlying technologies represented other breakthroughs, including hierarchical drawings and constraint-satisfaction methods. When asked how he could have done this all in the space of one year, Sutherland replied, "Well, I didn't know it was hard."


Transformed compiler design and theory, developed RISC architecture
JOHN COCKE
A.M.
TURING
1987
AWARD
acm

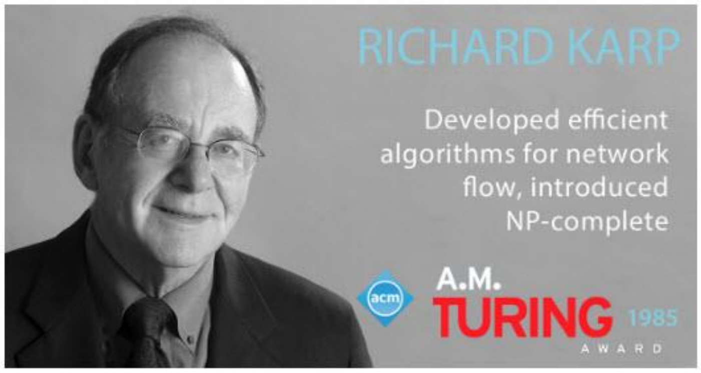
John Cocke

John Cocke made fundamental contributions to the architecture of high performance computers and to the design of optimizing compilers. Cocke led the 801 Minicomputer project at IBM, which adopted an innovative design philosophy based on a tight coupling between—and simultaneous development of—the hardware and the compiler. The result was a product with hardware whose instructions were vastly streamlined compared to previous models. The approach Cocke took in developing the "801" had a broad impact on computer architecture research, and was later called the Reduced Instruction Set Computer (RISC) approach. Cocke's pioneering research on computer architecture and optimizing compilers had a lasting impact on computing systems.



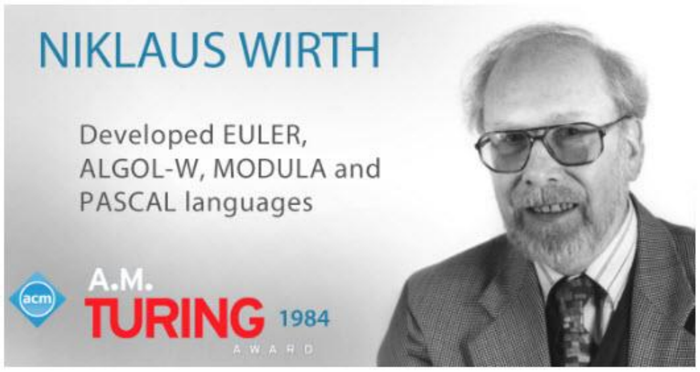
Robert Tarjan and John Hopcroft

When Robert Tarjan read fellow ACM A.M. Turing Award recipient Donald Knuth's book *The Art of Programming* he was hooked on studying the analysis of algorithms. While completing his PhD at Stanford, Tarjan, in collaboration with John Hopcroft, developed the first linear time algorithm for planarity. At the time there was no commonly-used model for measuring efficiency analytically. Hopcroft and Tarjan decided that the model of computation would be a hypothetical computer in which the goal of the algorithm design was to minimize the worst-case running time. These techniques are now covered in most undergraduate courses in algorithm design. Tarjan and Hopcroft jointly received the Turing award for this and related work in 1986.


RICHARD KARP
Developed efficient algorithms for network flow, introduced NP-complete
acm
A.M.
TURING
1985
AWARD

Richard Karp

In 1972, Richard Karp published the paper "Reducibility among Combinatorial Problems," which provided a convincing explanation for the intrinsic difficulty of the Traveling Salesmen Problem. In that seminal paper, Karp expanded on the concept (first introduced by Stephen Cook in 1971, and independently by Leonid Levin) of NP-completeness and proved that if any of 21 well-known difficult problems (including TSP) could be solved by an efficient algorithm, then all of the problems could be efficiently solved. Karp's work on NP-completeness substantially motivated the discussion among computer scientists and mathematicians of the famous unsolved P = NP question, which is considered the most important open question in both fields. Karp also made significant contributions to the field of computational biology later in his career.

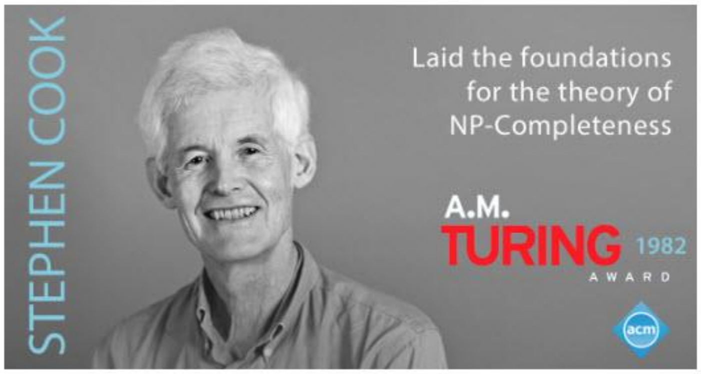


Niklaus Wirth

Early in his career, Niklaus Wirth created the Euler and PL360 programming languages. These broke new ground in formal separation of syntax and semantics; in novel implementation techniques, and in careful language design for efficient implementation with specific parsing methods. Later, working with Tony Hoare, Wirth used the Euler language as the basis of Algol-W, which then became the basis for Pascal. This simple, flexible language retained Algol's code structures, logical completeness, and support for recursion, but stripped away some of its complexity and added support for complex and user-defined data types. Because of its versatility, Pascal provided a foundation for future computer languages, systems, and architectural research for years to come.

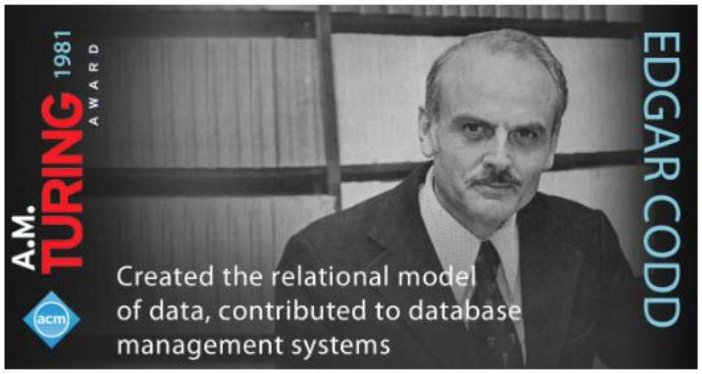
Ken Thompson and Dennis Ritchie

1993 ACM A.M. Turing Award co-recipients Ken Thompson and Dennis Ritchie met at the Bell Laboratories in 1966. It was there that in the span of a month, Thompson and Richie wrote the first version of the Unix operating system for a Digital Equipment Corporation PDP-7 using a cross-assembler that ran on GECOS. In 1973, at the second ACM Symposium on Operating Systems Principles, Thompson and Ritchie presented an elegant, simple, and well-written paper that described Unix, and by the end of that year there were over 20 Unix systems running. Unix went on have a far-reaching impact on its contemporary and subsequent operating systems.



Stephen Cook

In 1970, Stephen Cook published his seminal paper, "The complexity of theorem proving procedures," while on faculty at the University of Toronto, and presented it a year later at the 3rd Annual ACM Symposium on Theory of Computing (STOC '71). This paper marked the introduction of the theory of NP-completeness, which henceforth occupied a central place in theoretical computer science. The theory of NP-completeness provides a way to characterize the difficulty of computational problems with respect to the time -- that is, the maximum number of computational steps required to solve a problem -- as a function of input size. Cook received the 1982 ACM Turing Award for this profound contribution to computing.


EDGAR CODD
Created the relational model of data, contributed to database management systems
A.M. TURING AWARD 1981

Edgar Codd

In 1957, while at IBM, Edgar Codd worked on the design of STRETCH (the IBM 7030), leading the team that developed the world's first multiprogramming system. Later, turning his attention to database issues, Codd recognized the potential improvement that a solid theoretical foundation would provide to the database systems at the time. Codd applied his knowledge of mathematical logic to create the invention with which his name will forever be associated: the relational model of data. Now widely recognized as one of the great technical achievements of the 20th century, this visionary breakthrough revolutionized the way databases were perceived, and transformed the entire database field into a respectable scientific discipline.

ANTONY HOARE
Made enduring contributions to programming language design and definition
A.M. TURING AWARD 1980

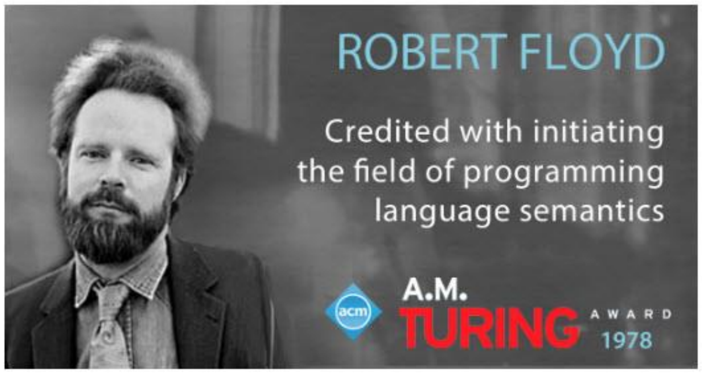
Antony Hoare

In the early 1960s, while working at a small British computer company called Elliott Brothers, Anthony ("Tony") Hoare led the team that produced the ALGOL 60 compiler for the Elliott 503 computer. This experience gave Hoare and appreciation for the crucial role of programming languages that would shape the focus of much of his later work. His paper "An Axiomatic Basis for Computer Programming," published in Communications of the ACM, became one of the most influential works on the theory of programming. In it, Hoare developed the logical system known as Hoare triples, and used this to reframe the semantics of programming languages. In 1978, Hoare's book Communicating Sequential Processes proposed the CSP language, where the interaction between programs was limited to pre-planned communications.

KENNETH IVERSON
Developed APL notation and its implementation for different use cases
A.M. TURING AWARD 1979

Kenneth Iverson

In the late 1950s, while on faculty at Harvard, Kenneth Iverson began work on a mathematical notation for expressing algorithms based on an extension of the higher dimensional arrays of tensor algebra and the concept of operators. In 1960, Ken joined the IBM Research Center, where he continued his work toward developing this new notation. In 1962 Ken published the book A Programming Language, from which came the acronym "APL." The first implementation of APL was written in 1965 in Fortran for an IBM 7090 computer, and was used in batch mode. Iverson continued to develop implementations of APL for both commercial and educational purposes throughout the 1970s and 80s.


ROBERT FLOYD
Credited with initiating the field of programming language semantics
A.M. TURING AWARD 1978

Robert Floyd

In the early part of his career, Robert Floyd published influential papers on compilers that described a new notation for symbol manipulation systems, and introduced a new method of scanning arithmetic expressions to produce more efficient machine code. Floyd later developed a notation that assigned conditions at each branch and entry point in a program, and it was his influential paper on this topic which ultimately inspired Tony Hoare to develop Hoare triples. Throughout his career, Floyd also invented important practical algorithms that have uses including finding the shortest paths through networks, computing the median of data, and rendering grayscale images with binary pixels using error diffusion (the Floyd-Steinberg algorithm). A longstanding collaborator with Donald Knuth, Floyd was also the main proof reader and critic for Knuth's famous The Art of Computer Programming.

JOHN BACKUS
Profoundly influenced high-level programming system design through work on FORTRAN.
A.M. TURING AWARD 1977

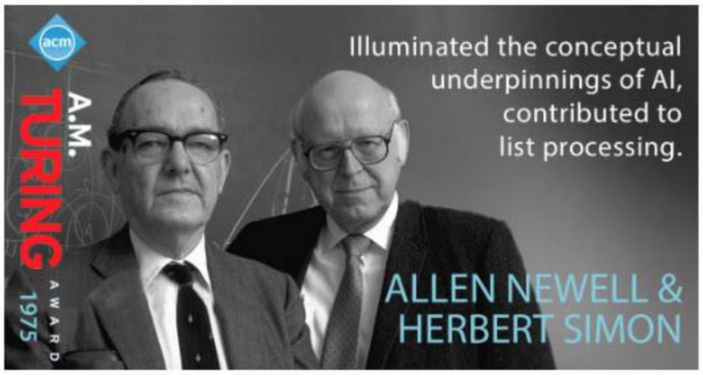
John Backus

John Backus's entry into computing was serendipitous. While touring the IBM Computing Center in New York City, Backus saw the Selective Sequence Electronic Calculator (SSEC), and made a passing comment to his tour guide about his interest in working on it. He was then taken upstairs to meet the SSEC project director, and hired on the spot as a programmer. While working at IBM, Backus invented a program called Speedcoding to facilitate programming. He later proposed the creation of a new language based on Speedcoding that would facilitate the development of the IBM 704. After two years of work, Backus and his team released FORTRAN, which quickly gained considerable traction in the scientific community and went on to become the dominant programming language for scientific applications for many decades.

MICHAEL RABIN & DANA SCOTT
Introduced nondeterministic machines by publishing "Finite Automata and Their Decision Problem."
acm
A.M. TURING AWARD 1976

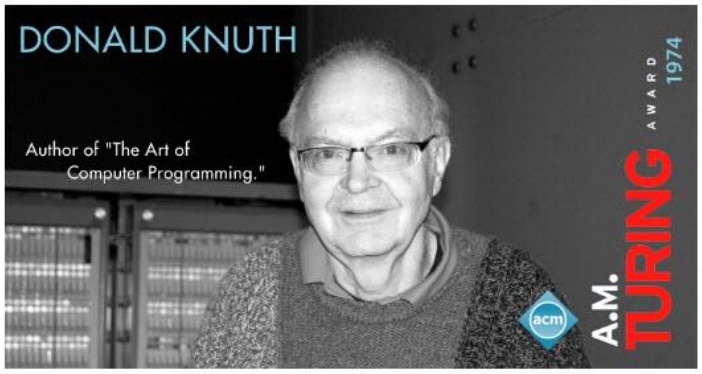
Michael Rabin and Dana Scott

Michael Rabin and Dana Scott met at an IBM summer research workshop for a select group of young scientists. There, he and Scott collaborated on the famous paper, "Finite Automata and Their Decision Problem," which led to their joint Turing Award in 1976. In this paper, Rabin and Scott moved away from artificial neural networks that earlier thinkers on the subject had focused on, and instead used a computational model known as a finite state machine. These had been investigated before, but Rabin and Scott considered a nondeterministic machine that had not one but several possible transitions out of each state. This concept has proven to be extremely valuable in the theoretical investigation of many problems, and continues to be an inspiration for new work.


acm
A.M. TURING AWARD 1975
Illuminated the conceptual underpinnings of AI, contributed to list processing.
ALLEN NEWELL & HERBERT SIMON

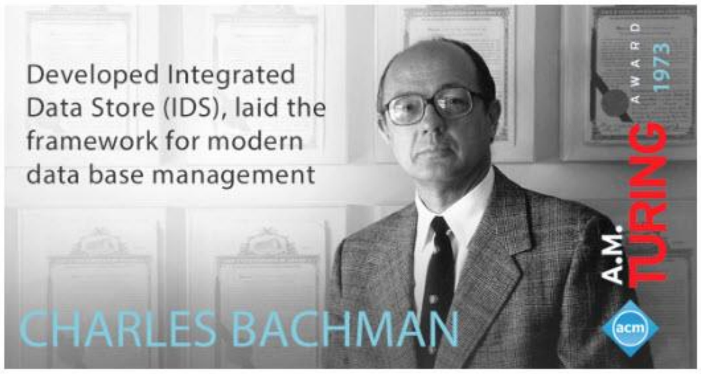
Allen Newell & Herbert Simon

Allen Newell and Herbert Simon's work – both individually and in their transformative collaborations on artificial intelligence theory and programs – was always driven by the goal of understanding the cognitive architecture of the human mind, and how computation could augment the human capacity for problem solving. Drawing on his background in psychology, Herbert Simon laid out the framework for heuristic programming techniques. Later, Simon collaborated with Allen Newell, as well as with J.C. Shaw, to expand heuristic programming into the first successful AI program, the Logic Theorist (LT), in 1956. LT became a remarkable success, and Simon and Newell elaborated on this to develop the first list processing language, IPL (Information Processing Language).



Donald Knuth

Donald Knuth noticed something about the existing literature on computer science when he first surveyed it in 1962: it wasn't very good. An especially adept writer with an instinct for organization, Knuth took on the task of writing a concise, comprehensive text book on compilers, which later turned into an encompassing tome on computer programming. In June, 1965, the book had reached 3,000 hand-written pages, and by 1973, Knuth's work had evolved into volumes 1-3 of *The Art of Computer Programming* ("TAOCP"), the work that went on to become a perennial staple in computer science education. http://ow.ly/YIxPK


Developed Integrated Data Store (IDS), laid the framework for modern data base management
CHARLES BACHMAN
A.M. TURING AWARD 1973
acm

Charles Bachman

During his experience working for Dow Chemical and General Electric in the 1950s and 60s, Charles Bachman realized that in order to justify the sizable overhead expense of a computer, companies needed machines that could tie together business processes such as sales, accounting, and inventory, and provide managers with integrated, up-to-date information. In 1963, Bachman invented the Integrated Data Store, or IDS, which maintained a single set of shared files on disk, together with the tools to structure and maintain them. Because of the boldness and remarkable efficiency of this design, the "data base management system," as the IDS was being called, was one of the most important areas of business computing research and development.

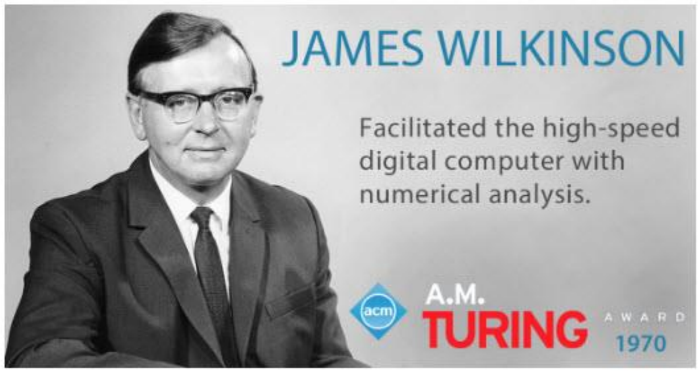
EDSGER W. DIJKSTRA
Fundamental contributions to programming as a high, intellectual challenge.
A.M. TURING AWARD 1972

Edsger Dijkstra

Edsger Dijkstra is known for championing computer programming as a high-level intellectual challenge and for his insistence and practical demonstration that programs be composed correctly. In 1968, Dijkstra published a letter to the editor in "Communications of the ACM" titled "Go To Statement Considered Harmful," arguing that the GO TO statement, found in many high-level programming languages, is a major source of errors, and should therefore be eliminated. This triggered a longstanding debate in the computing community that ultimately resulted in programming languages providing alternatives to GO TO. Around this time, Dijkstra began to rigorously advocate for the recognition of software development as its own scientific discipline. His book, "Notes on Structured Programming," published in 1970, has since had a far-reaching impact on all areas of computer science, and has been critical in establishing mathematical analyses of program design and specifications as central activities in computer science research.

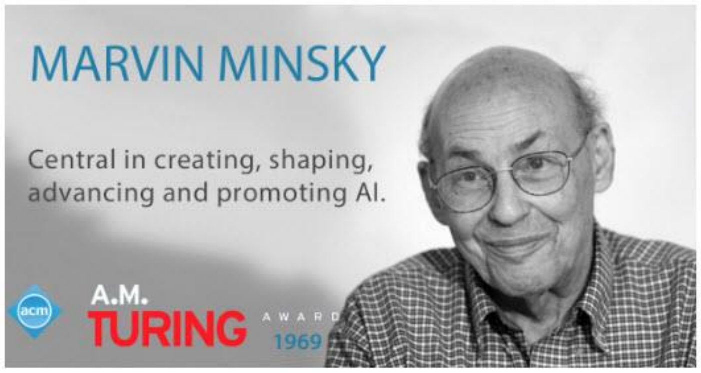
John McCarthy

In 1958, John McCarthy and fellow Turing Laureate Marvin Minsky formed the Artificial Intelligence Project at MIT, where pioneering work took place in a wide range of fields from robotics, the theory of computation and common- sense reasoning, to human-computer interfaces. At MIT, McCarthy and his colleagues made numerous critical contributions to the early advancement of artificial intelligence. It was here that McCarthy developed LISP, which ultimately became the standard artificial intelligence programming language, had a profound impact on computing writ large, and still continues to inform new languages. Also during this period, inspired by advancements made by the SAGE air defense system, McCarthy then came up with a scheme for creating general-purpose timesharing, the conceptual precursor to computer networking. McCarthy's work in this area was instrumental in the development of the ARPANET.


JAMES WILKINSON
Facilitated the high-speed digital computer with numerical analysis.
A.M. TURING AWARD 1970

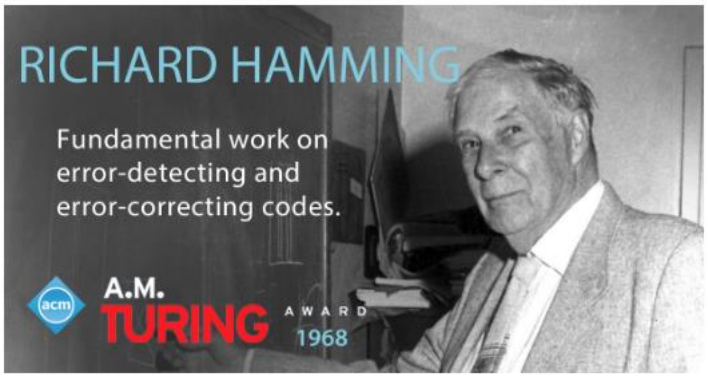
James Wilkinson

After World War II, James Wilkinson worked with Alan Turing himself at the National Physical Laboratory on problems associated with Turing's proposal to build an electronic computer. When Turing left the NPL, Wilkinson (with others) took over the development of Turing's computer—the Pilot ACE. Wilkinson used the Pilot ACE to experiment with various methods for solving matrix problems, and this work led to multiple discoveries. One of the most notable of these was his development of "backward" error analysis, which helped understand the precise role of roundoff error in matrix computations. In doing so, Wilkinson pioneered successful error analyses of all the matrix algorithms of his day, and this helped illuminate the benefits to technology that digital computers with the ability to solve systems of equations would have. Along with breakthrough work in linear algebraic computations, Wilkinson's work played an integral role in the development of high-speed digital computers.



Marvin Minsky

Marvin Minsky's work on artificial intelligence using symbol manipulation dates from the field's earliest days in the 1950s and 1960s. Many consider his 1960 paper, "Steps toward Artificial Intelligence," to be the call-to-arms for a generation of researchers. That paper established symbol manipulation to be at the center of any attempt at understanding intelligence. Later, in the early 1960s, Minsky founded the MIT Artificial Intelligence Laboratory with John McCarthy, which quickly became an epicenter of understanding machine intelligence. In his later work on "perceptrons" -- simple computational devices that capture some of the characteristics of neural behavior -- Minsky and Seymour Papert clearly defined the boundaries of preceptrons, thus raising the sophistication of research on neurally-inspired mechanisms to a new level. Taken together, Minsky's own work in the field and his founding of the MIT Artificial Intelligence Laboratory firmly establish him as one of the founders of the field of artificial intelligence.


RICHARD HAMMING
Fundamental work on error-detecting and error-correcting codes.
acm
A.M. TURING AWARD
1968

Richard Hamming

In 1945, Richard Hamming began work on the Manhattan Project, where he was put in charge of the IBM calculating machines that played a vital role in the project. His work during this period helped form his view of mathematics in which computation was of major importance. In 1946, Hamming accepted a position in the mathematics department at the Bell Telephone Laboratories in New Jersey. Here, he conducted his seminal work on error-detecting and error-correcting codes, and wrote a fundamental paper on this topic which created an entirely new field within information theory. "Hamming codes," "Hamming distance" and "Hamming metric," standard terms used today in coding theory and other areas of mathematics, all originated in Hamming's classic paper, "Error Detecting and Error Correcting Codes," and are of ongoing practical use in computer design. Hamming received the ACM Turing Award in 1968 for his foundational work in these fields.

MAURICE WILKES
Built and designed EDSAC, the first computer with an internally stored program.
acm
A.M. TURING AWARD
1967

Maurice Wilkes

In 1946, while sailing across the Atlantic on the Queen Mary, Maurice Wilkes began the design of a machine he called the Electronic Delay Storage Automatic Calculator, or EDSAC. He initiated work on the EDSAC in early 1947 at Cambridge University, and it sprang to life on May 6, 1949 as the world's first practical stored-program electronic computer capable of running realistic programs. When EDSAC became loaded to its capacity, Wilkes laid plans for EDSAC's successor, EDSAC 2. It was in designing the EDSAC 2 that Wilkes devised the design principle of microprogramming, which went on to have broad and enduring importance in computing. In the early 1960s, IBM based its pioneering System/360 computers around this idea, and it remains a cornerstone of computer architecture today.

ALAN PERLIS
Fundamental contributions to advanced programming techniques and compiler construction.
acm
A.M. TURING AWARD
1966

Alan Perlis

In 1955, Alan Perlis began work on the design of a "mathematical language compiler" for the Datatron 205 at Perdue University, which he and his team later named "IT" (for Internal Translator). IT was a "compiler" in that it automatically translated programs written in mathematical notation into machine code. IT became adopted by many university computing installations as a cheaper and more practical alternative to Fortran. In this way, IT contributed greatly to stimulating research on programming techniques. Perlis's IT spurred the evaluation (led by ACM) of the possibility of a "universal programming language." In May 1958, a meeting between a small group of ACM representatives in Zurich resulted in the definition of the International Algorithmic Language (IAL), which was renamed ALGOL the following year.

AWARDS & RECOGNITION

Spotlight on Turing Laureates

The ACM A.M. Turing Award, computing's most prestigious honor, acknowledges individuals who have made lasting and major contributions to the field of computing. Here, we look back at some of these technologies and breakthroughs that continue to impact our lives, and the remarkable innovators who helped shape them.