Computer science is the study of computation, information, and automation.[1][2][3] Computer science spans theoretical disciplines (such as algorithms, theory of computation, and information theory) to applied disciplines (including the design and implementation of hardware and software).[4][5][6] Though more often considered an academic discipline, computer science is closely related to computer programming.[7]

Algorithms and data structures are central to computer science.[8] The theory of computation concerns abstract models of computation and general classes of problems that can be solved using them. The fields of cryptography and computer security involve studying the means for secure communication and for preventing security vulnerabilities. Computer graphics and computational geometry address the generation of images. Programming language theory considers different ways to describe computational processes, and database theory concerns the management of repositories of data. Human–computer interaction investigates the interfaces through which humans and computers interact, and software engineering focuses on the design and principles behind developing software. Areas such as operating systems, networks and embedded systems investigate the principles and design behind complex systems. Computer architecture describes the construction of computer components and computer-operated equipment. Artificial intelligence and machine learning aim to synthesize goal-orientated processes such as problem-solving, decision-making, environmental adaptation, planning and learning found in humans and animals. Within artificial intelligence, computer vision aims to understand and process image and video data, while natural language processing aims to understand and process textual and linguistic data.

The fundamental concern of computer science is determining what can and cannot be automated.[2][9][3][10][11] The Turing Award is generally recognized as the highest distinction in computer science.[12][13]

History

Main article: History of computer science

History of computing

Hardware

Hardware before 1960Hardware 1960s to present

Software

SoftwareSoftware configuration managementUnixFree software and open-source software

Computer science

Artificial intelligenceCompiler constructionEarly computer scienceOperating systemsProgramming languagesProminent pioneersSoftware engineering

Modern concepts

General-purpose CPUsGraphical user interfaceInternetLaptopsPersonal computersVideo gamesWorld Wide WebCloud

By country

BulgariaEastern BlocPolandRomaniaSouth AmericaSoviet UnionYugoslavia

Timeline of computing

before 19501950–19791980–19891990–19992000–20092010–20192020–presentmore timelines ...

Glossary of computer science

Category

Gottfried Wilhelm Leibniz (1646–1716) developed logic in a binary number system and has been called the "founder of computer science".[14]

Charles Babbage is sometimes referred to as the "father of computing".[15]

Ada Lovelace published the first algorithm intended for processing on a computer.[16]

The earliest foundations of what would become computer science predate the invention of the modern digital computer. Machines for calculating fixed numerical tasks such as the abacus have existed since antiquity, aiding in computations such as multiplication and division. Algorithms for performing computations have existed since antiquity, even before the development of sophisticated computing equipment.[17]

Wilhelm Schickard designed and constructed the first working mechanical calculator in 1623.[18] In 1673, Gottfried Leibniz demonstrated a digital mechanical calculator, called the Stepped Reckoner.[19] Leibniz may be considered the first computer scientist and information theorist, because of various reasons, including the fact that he documented the binary number system. In 1820, Thomas de Colmar launched the mechanical calculator industry[note 1] when he invented his simplified arithmometer, the first calculating machine strong enough and reliable enough to be used daily in an office environment. Charles Babbage started the design of the first automatic mechanical calculator, his Difference Engine, in 1822, which eventually gave him the idea of the first programmable mechanical calculator, his Analytical Engine.[20] He started developing this machine in 1834, and "in less than two years, he had sketched out many of the salient features of the modern computer".[21] "A crucial step was the adoption of a punched card system derived from the Jacquard loom"[21] making it infinitely programmable.[note 2] In 1843, during the translation of a French article on the Analytical Engine, Ada Lovelace wrote, in one of the many notes she included, an algorithm to compute the Bernoulli numbers, which is considered to be the first published algorithm ever specifically tailored for implementation on a computer.[22] Around 1885, Herman Hollerith invented the tabulator, which used punched cards to process statistical information; eventually his company became part of IBM. Following Babbage, although unaware of his earlier work, Percy Ludgate in 1909 published[23] the 2nd of the only two designs for mechanical analytical engines in history. In 1914, the Spanish engineer Leonardo Torres Quevedo published his Essays on Automatics,[24] and designed, inspired by Babbage, a theoretical electromechanical calculating machine which was to be controlled by a read-only program. The paper also introduced the idea of floating-point arithmetic.[25][26] In 1920, to celebrate the 100th anniversary of the invention of the arithmometer, Torres presented in Paris the Electromechanical Arithmometer, a prototype that demonstrated the feasibility of an electromechanical analytical engine,[27] on which commands could be typed and the results printed automatically.[28] In 1937, one hundred years after Babbage's impossible dream, Howard Aiken convinced IBM, which was making all kinds of punched card equipment and was also in the calculator business[29] to develop his giant programmable calculator, the ASCC/Harvard Mark I, based on Babbage's Analytical Engine, which itself used cards and a central computing unit. When the machine was finished, some hailed it as "Babbage's dream come true".[30]

During the 1940s, with the development of new and more powerful computing machines such as the Atanasoff–Berry computer and ENIAC, the term computer came to refer to the machines rather than their human predecessors.[31] As it became clear that computers could be used for more than just mathematical calculations, the field of computer science broadened to study computation in general. In 1945, IBM founded the Watson Scientific Computing Laboratory at Columbia University in New York City. The renovated fraternity house on Manhattan's West Side was IBM's first laboratory devoted to pure science. The lab is the forerunner of IBM's Research Division, which today operates research facilities around the world.[32] Ultimately, the close relationship between IBM and Columbia University was instrumental in the emergence of a new scientific discipline, with Columbia offering one of the first academic-credit courses in computer science in 1946.[33] Computer science began to be established as a distinct academic discipline in the 1950s and early 1960s.[7][34] The world's first computer science degree program, the Cambridge Diploma in Computer Science, began at the University of Cambridge Computer Laboratory in 1953. The first computer science department in the United States was formed at Purdue University in 1962.[35] Since practical computers became available, many applications of computing have become distinct areas of study in their own rights.

See also: History of computing and History of informatics

Etymology and scope

See also: Informatics § Etymology

Although first proposed in 1956,[36] the term "computer science" appears in a 1959 article in Communications of the ACM,[37] in which Louis Fein argues for the creation of a Graduate School in Computer Sciences analogous to the creation of Harvard Business School in 1921.[38] Louis justifies the name by arguing that, like management science, the subject is applied and interdisciplinary in nature, while having the characteristics typical of an academic discipline.[37] His efforts, and those of others such as numerical analyst George Forsythe, were rewarded: universities went on to create such departments, starting with Purdue in 1962.[39] Despite its name, a significant amount of computer science does not involve the study of computers themselves. Because of this, several alternative names have been proposed.[40] Certain departments of major universities prefer the term computing science, to emphasize precisely that difference. Danish scientist Peter Naur suggested the term datalogy,[41] to reflect the fact that the scientific discipline revolves around data and data treatment, while not necessarily involving computers. The first scientific institution to use the term was the Department of Datalogy at the University of Copenhagen, founded in 1969, with Peter Naur being the first professor in datalogy. The term is used mainly in the Scandinavian countries. An alternative term, also proposed by Naur, is data science; this is now used for a multi-disciplinary field of data analysis, including statistics and databases.

In the early days of computing, a number of terms for the practitioners of the field of computing were suggested in the Communications of the ACM—turingineer, turologist, flow-charts-man, applied meta-mathematician, and applied epistemologist.[42] Three months later in the same journal, comptologist was suggested, followed next year by hypologist.[43] The term computics has also been suggested.[44] In Europe, terms derived from contracted translations of the expression "automatic information" (e.g. "informazione automatica" in Italian) or "information and mathematics" are often used, e.g. informatique (French), Informatik (German), informatica (Italian, Dutch), informática (Spanish, Portuguese), informatika (Slavic languages and Hungarian) or pliroforiki (πληροφορική, which means informatics) in Greek. Similar words have also been adopted in the UK (as in the School of Informatics, University of Edinburgh).[45] "In the U.S., however, informatics is linked with applied computing, or computing in the context of another domain."[46]

A folkloric quotation, often attributed to—but almost certainly not first formulated by—Edsger Dijkstra, states that "computer science is no more about computers than astronomy is about telescopes."[note 3] The design and deployment of computers and computer systems is generally considered the province of disciplines other than computer science. For example, the study of computer hardware is usually considered part of computer engineering, while the study of commercial computer systems and their deployment is often called information technology or information systems. However, there has been exchange of ideas between the various computer-related disciplines. Computer science research also often intersects other disciplines, such as cognitive science, linguistics, mathematics, physics, biology, Earth science, statistics, philosophy, and logic.

Computer science is considered by some to have a much closer relationship with mathematics than many scientific disciplines, with some observers saying that computing is a mathematical science.[7] Early computer science was strongly influenced by the work of mathematicians such as Kurt Gödel, Alan Turing, John von Neumann, Rózsa Péter and Alonzo Church and there continues to be a useful interchange of ideas between the two fields in areas such as mathematical logic, category theory, domain theory, and algebra.[36]

The relationship between computer science and software engineering is a contentious issue, which is further muddied by disputes over what the term "software engineering" means, and how computer science is defined.[47] David Parnas, taking a cue from the relationship between other engineering and science disciplines, has claimed that the principal focus of computer science is studying the properties of computation in general, while the principal focus of software engineering is the design of specific computations to achieve practical goals, making the two separate but complementary disciplines.[48]

The academic, political, and funding aspects of computer science tend to depend on whether a department is formed with a mathematical emphasis or with an engineering emphasis. Computer science departments with a mathematics emphasis and with a numerical orientation consider alignment with computational science. Both types of departments tend to make efforts to bridge the field educationally if not across all research.

Philosophy

Main article: Philosophy of computer science

Epistemology of computer science

Despite the word "science" in its name, there is debate over whether or not computer science is a discipline of science,[49] mathematics,[50] or engineering.[51] Allen Newell and Herbert A. Simon argued in 1975,

Computer science is an empirical discipline. We would have called it an experimental science, but like astronomy, economics, and geology, some of its unique forms of observation and experience do not fit a narrow stereotype of the experimental method. Nonetheless, they are experiments. Each new machine that is built is an experiment. Actually constructing the machine poses a question to nature; and we listen for the answer by observing the machine in operation and analyzing it by all analytical and measurement means available.[51]

It has since been argued that computer science can be classified as an empirical science since it makes use of empirical testing to evaluate the correctness of programs, but a problem remains in defining the laws and theorems of computer science (if any exist) and defining the nature of experiments in computer science.[51] Proponents of classifying computer science as an engineering discipline argue that the reliability of computational systems is investigated in the same way as bridges in civil engineering and airplanes in aerospace engineering.[51] They also argue that while empirical sciences observe what presently exists, computer science observes what is possible to exist and while scientists discover laws from observation, no proper laws have been found in computer science and it is instead concerned with creating phenomena.[51]

Proponents of classifying computer science as a mathematical discipline argue that computer programs are physical realizations of mathematical entities and programs can be deductively reasoned through mathematical formal methods.[51] Computer scientists Edsger W. Dijkstra and Tony Hoare regard instructions for computer programs as mathematical sentences and interpret formal semantics for programming languages as mathematical axiomatic systems.[51]

Paradigms of computer science

A number of computer scientists have argued for the distinction of three separate paradigms in computer science. Peter Wegner argued that those paradigms are science, technology, and mathematics.[52] Peter Denning's working group argued that they are theory, abstraction (modeling), and design.[7] Amnon H. Eden described them as the "rationalist paradigm" (which treats computer science as a branch of mathematics, which is prevalent in theoretical computer science, and mainly employs deductive reasoning), the "technocratic paradigm" (which might be found in engineering approaches, most prominently in software engineering), and the "scientific paradigm" (which approaches computer-related artifacts from the empirical perspective of natural sciences,[53] identifiable in some branches of artificial intelligence).[54] Computer science focuses on methods involved in design, specification, programming, verification, implementation and testing of human-made computing systems.[55]

Fields

This is a dynamic list and may never be able to satisfy particular standards for completeness. You can help by adding missing items with reliable sources.

Further information: Outline of computer science

As a discipline, computer science spans a range of topics from theoretical studies of algorithms and the limits of computation to the practical issues of implementing computing systems in hardware and software.[56][57] CSAB, formerly called Computing Sciences Accreditation Board—which is made up of representatives of the Association for Computing Machinery (ACM), and the IEEE Computer Society (IEEE CS)[58]—identifies four areas that it considers crucial to the discipline of computer science: theory of computation, algorithms and data structures, programming methodology and languages, and computer elements and architecture. In addition to these four areas, CSAB also identifies fields such as software engineering, artificial intelligence, computer networking and communication, database systems, parallel computation, distributed computation, human–computer interaction, computer graphics, operating systems, and numerical and symbolic computation as being important areas of computer science.[56]

Theoretical computer science

Main article: Theoretical computer science

Theoretical Computer Science is mathematical and abstract in spirit, but it derives its motivation from the practical and everyday computation. Its aim is to understand the nature of computation and, as a consequence of this understanding, provide more efficient methodologies.

Theory of computation

Main article: Theory of computation

According to Peter Denning, the fundamental question underlying computer science is, "What can be automated?"[3] Theory of computation is focused on answering fundamental questions about what can be computed and what amount of resources are required to perform those computations. In an effort to answer the first question, computability theory examines which computational problems are solvable on various theoretical models of computation. The second question is addressed by computational complexity theory, which studies the time and space costs associated with different approaches to solving a multitude of computational problems.