Computer science

From Wikipedia, the free encyclopedia

Computer science or **computing science** (sometimes abbreviated **CS**) is the study of the theoretical foundations of information and computation, and of practical techniques for their implementation and application in computer systems.[1][2][3] It is frequently described as the systematic study of algorithmic processes that create, describe and transform information. According to Peter J. Denning, the fundamental question underlying computer science is, *"What can be (efficiently) automated?"*[4] Computer science has many sub-fields; some, such as computer graphics, emphasize the computation of specific results, while others, such as computational complexity theory, study the properties of computational problems. Still others focus on the challenges in implementing computations. For example, programming language theory studies approaches to describing computations, while computer programming applies specific programming languages to solve specific computational problems, and human-computer interaction focuses on the challenges in making computers and computations useful, usable, and universally accessible to people.

Computer science deals with the theoretical foundations of information and computation, and of practical techniques for their implementation and application.

The general public sometimes confuses computer science with vocational areas that deal with computers (such as information technology), or think that it relates to their own experience of computers, which typically involves activities such as gaming, web-browsing, and word-processing. However, the focus of computer science is more on understanding the properties of the programs used to implement software such as games and web-browsers, and using that understanding to create new programs or improve existing ones.[5]

Contents

challenge= چالش
vocational= کار، شغل، پیشه
predate = قبل از موقع بخصوص واقع شدن
abacus= چرتکه
antiquity=عهد عتیق ، روزگار باستان ، قدمت
Predecessor = اسبق ، سابق ، قبلي ، جد، اجداد،
Broaden پهن کردن ، وسیع کردن ، منتشر کردن
Commonplace پیش پا افتاده ، معمولي ، مبتذل ، همه جايي
Intractable خود سر، سرپیچ ، متمرد، خود سرانه ، لجوج ، خیره سر،ستیزه جو، لجوجانه ، رام نشدني
Allied پیوسته ، متحد
Liquidity قابلیت تبدیل به پول ، تسویه پذیري ، ابگون پذیري
Accredit اختیار دادن ، اطمینان کردن (به)، مورد اطمینان بودن یا شدن ، برسمیت شناختن (موسسات فرهنگي)، معتبر شناختن
Aspect نمود، سیما، منظر، صورت ، ظاهر، وضع ، جنبه
Continental اقلیمي ، قاره اي
Renown اوازه ، نام ، شهرت ، معروفیت ، اشتهار، صیت ، مشهور کردن
Contentious ستیزه جو، دعوایي ، متنازع فیه ، ستیزگر

History

Main article: History of computer science

The early 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. Wilhelm Schickard built the first mechanical calculator in 1623.[6] Charles Babbage designed a difference engine in Victorian times[7] helped by Ada Lovelace.[8] Around 1900, punch-card machines[9] were introduced. However, all of these machines were constrained to perform a single task, or at best some subset of all possible tasks.

During the 1940s, as newer and more powerful computing machines were developed, the term *computer* came to refer to the machines rather than their human predecessors. 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. Computer science began to be established as a distinct academic discipline in the 1950s and early 1960s.[4][10] The first computer science degree program in the United States was formed at Purdue University in 1962.[11] Since practical computers became available, many applications of computing have become distinct areas of study in their own right.

Although many initially believed it was impossible that computers themselves could actually be a scientific field of study, in the late fifties it gradually became accepted among the greater academic population.[12] It is the now well-known IBM brand that formed part of the computer science revolution during this time. IBM (short for International Business Machines) released the IBM 704 and later the IBM 709 computers, which were widely used during the exploration period of such devices. "Still, working with the IBM [computer] was frustrating...if you had misplaced as much as one letter in one instruction, the program would crash, and you would have to start the whole process over again".[12] During the late 1950s, the computer science discipline was very much in its developmental stages, and such issues were commonplace.

Time has seen significant improvements in the usability and effectiveness of computer science technology. Modern society has seen a significant shift from computers being used solely by experts or professionals to a more widespread user base.

Major achievements

Despite its short history as a formal academic discipline, computer science has made a number of fundamental contributions to science and society. These include:

- The start of the "digital revolution," which includes the current Information Age and the Internet.[14]
- A formal definition of computation and computability, and proof that there are computationally unsolvable and intractable problems.[15]
- The concept of a programming language, a tool for the precise expression of methodological information at various levels of abstraction.[16]
- In cryptography, breaking the Enigma machine was an important factor contributing to the Allied victory in World War II.[13]
- Scientific computing enabled advanced study of the mind, and mapping the human genome became possible with Human Genome Project.[14] Distributed computing projects such as Folding@home explore protein folding.
- Algorithmic trading has increased the efficiency and liquidity of financial markets by using artificial intelligence, machine learning, and other statistical and numerical techniques on a large scale.[17]

The German military used the Enigma machine during World War II for communication they thought to be secret. The large-scale decryption of Enigma traffic at Bletchley Park was an important factor that contributed to Allied victory in WWII.[13]

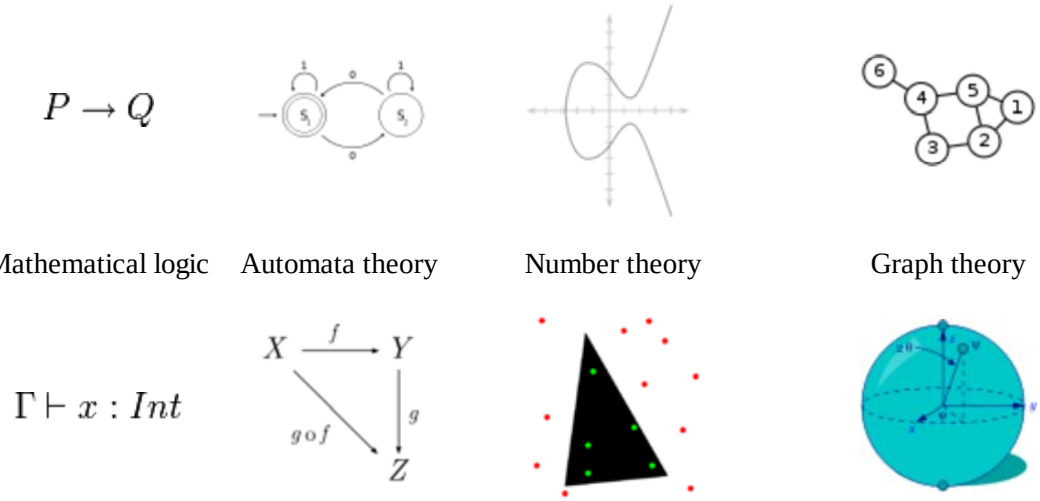
Areas of computer science

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.[18][19] The Computer Sciences Accreditation Board (CSAB) [20] – which is made up of representatives of the Association for Computing Machinery (ACM), the Institute of Electrical and Electronics Engineers Computer Society, and the Association for Information Systems – 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, computer-human interaction, computer graphics, operating systems, and numerical and symbolic computation as being important areas of computer science.[18]

Theoretical computer science

The broader field of theoretical computer science encompasses both the classical theory of computation and a wide range of other topics that focus on the more abstract, logical, and mathematical aspects of computing.

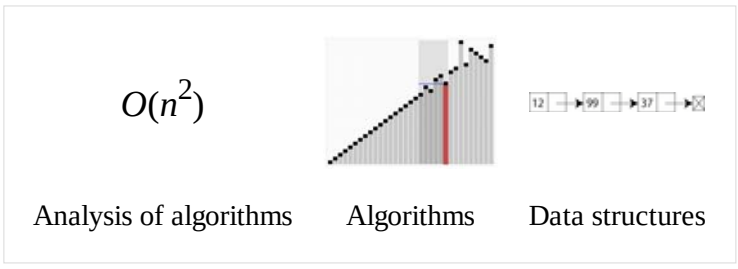


Theory of computation

The study of the 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 computational problem.

The famous "P=NP?" problem, one of the Millennium Prize Problems,[21] is an open problem in the theory of computation.

Algorithms and data structures



Programming methodology and languages

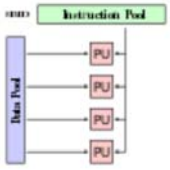
Compilers

Programming languages

Computer elements and architecture

Digital logic

Microarchitecture

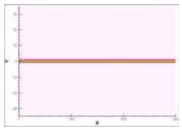


Multiprocessing

Numerical and symbolic computation

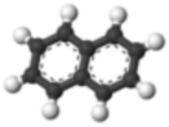
$y = \sin(x) + c$

Symbolic computation



Numerical analysis

Computational physics

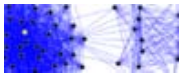


Computational chemistry

Bioinformatics

Computational neuroscience

Cognitive Science



Computational sociology

Computational economics

Applied Computer Science

The following areas are often studied from a more theoretical, computer science viewpoint, as well as from a more practical, engineering perspective.

Operating systems

Computer networks

Computer graphics

Computer vision

Databases

Information retrieval

Computer security

Artificial intelligence

Robotics

Human–computer interaction

Ubiquitous computing

Software engineering

Relationship with other fields

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. Certain departments of major universities prefer the term *computing science*, to emphasize precisely that difference. Danish scientist Peter Naur suggested the term *datalogy*, 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. Also, 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*.[22] Three months later in the same journal, *comptologist* was suggested, followed next year by *hypologist*.[23] The term *computics* has also been suggested.[24] In continental Europe, names such as *informatique* (French), *Informatik* (German) or *informatica* (Dutch), derived from information and possibly mathematics or automatic, are more common than names derived from computer/computation.

The renowned computer scientist Edsger Dijkstra stated, "Computer science is no more about computers than astronomy is about telescopes." 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 much cross-fertilization of ideas between the various computer-related disciplines. Computer science research has also often crossed into other disciplines, such as philosophy, cognitive science, linguistics, mathematics, physics, and economics.

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.[4] Early computer science was strongly influenced by the work of mathematicians such as Kurt Gödel and Alan Turing, 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.

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. 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.[25]

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

Computer science education

Some universities teach computer science as a theoretical study of computation and algorithmic reasoning. These programs often feature the theory of computation, analysis of algorithms, formal methods, concurrency theory, databases, computer graphics and systems analysis, among others. They typically also teach computer programming, but treat it as a vessel for the support of other fields of computer science rather than a central focus of high-level study.

Other colleges and universities, as well as secondary schools and vocational programs that teach computer science, emphasize the practice of advanced programming rather than the theory of algorithms and computation in their computer science curricula. Such curricula tend to focus on those skills that are important to workers entering the software industry. The practical aspects of computer programming are often referred to as software engineering. However, there is a lot of disagreement over the meaning of the term, and whether or not it is the same thing as programming.

See also

Main article: Outline of computer science

- Career domains in computer science
- Computer scientist
- Computing
- Didactics of informatics
- English in computer science
- History of computer science
- Informatics
- Information and communication technologies for development
- List of academic computer science departments
- List of computer science conferences
- List of computer scientists
- List of open problems in computer science
- List of publications in computer science
- List of pioneers in computer science
- List of software engineering topics
- Philosophy of computer science
- Software engineering
- Women in computing

References

1. ^ "*Computer science is the study of information*" New Jersey Institute of Technology (http://www.njit.edu/v2/archivecatalog/undergraduate/91/19-und.html) , Gutenberg Information Technologies
2. ^ "*Computer science is the study of computation.*" Computer Science Department, College of Saint Benedict (http://www.csbsju.edu/computerscience/curriculum) , Saint John's University
3. ^ "*Computer Science is the study of all aspects of computer systems, from the theoretical foundations to the very practical aspects of managing large software projects.*" Massey University (http://study.massey.ac.nz/major.asp?major_code=2010&prog_code=93068)
4. ^ ***a*** ***b*** ***c*** Denning, P.J. (2000). "Computer Science: The Discipline" (http://web.archive.org/web/20060525195404/http://www.idi.ntnu.no/emner/dif8916/denning.pdf) (PDF). *Encyclopedia of Computer Science*. http://web.archive.org/web/20060525195404/http://www.idi.ntnu.no/emner/dif8916/denning.pdf.
5. ^ "*Common myths and preconceptions about Cambridge Computer Science*" Computer Science Department (http://www.cl.cam.ac.uk/admissions/undergraduate/myths/) , University of Cambridge
6. ^ Nigel Tout (2006). "Calculator Timeline" (http://www.vintagecalculators.com/html/calculator_time-line.html) . *Vintage Calculator Web Museum*. http://www.vintagecalculators.com/html/calculator_time-line.html. Retrieved 2006-09-18.
7. ^ "Science Museum - Introduction to Babbage" (http://www.sciencemuseum.org.uk/on-line/babbage/index.asp) . http://www.sciencemuseum.org.uk/on-line/babbage/index.asp. Retrieved 2006-09-24.
8. ^ "A Selection and Adaptation From Ada's Notes found in "Ada, The Enchantress of Numbers," by Betty Alexandra Toole Ed.D. Strawberry Press, Mill Valley, CA" (http://www.scottlan.edu/Lriddle/women/ada-love.htm) . http://www.scottlan.edu

/Lriddle/women/ada-love.htm. Retrieved 2006-05-04.
9. ^ "IBM Punch Cards in the U.S. Army" (http://www.pattonhq.com/ibm.html) . http://www.pattonhq.com/ibm.html. Retrieved 2006-09-24.
10. ^ http://www.cl.cam.ac.uk/conference/EDSAC99/statistics.html
11. ^ Computer science pioneer Samuel D. Conte dies at 85 (http://www.cs.purdue.edu/feature/conte.html) July 1, 2002
12. ^ **a** **b** Levy, Steven (1984). *Hackers: Heroes of the Computer Revolution*. Doubleday. ISBN 0-385-19195-2.
13. ^ **a** **b** David Kahn, The Codebreakers, 1967, ISBN 0-684-83130-9.
14. ^ **a** **b** http://www.cis.cornell.edu/Dean/Presentations/Slides/bgu.pdf
15. ^ Constable, R.L. (March 2000) (PDF). *Computer Science: Achievements and Challenges circa 2000* (http://www.cs.cornell.edu/cis-dean/bgu.pdf) . http://www.cs.cornell.edu/cis-dean/bgu.pdf.
16. ^ Abelson, H.; G.J. Sussman with J. Sussman (1996). *Structure and Interpretation of Computer Programs* (2nd ed.). MIT Press. ISBN 0-262-01153-0. "The computer revolution is a revolution in the way we think and in the way we express what we think. The essence of this change is the emergence of what might best be called *procedural epistemology* — the study of the structure of knowledge from an imperative point of view, as opposed to the more declarative point of view taken by classical mathematical subjects."
17. ^ Black box traders are on the march (http://www.telegraph.co.uk/money/main.jhtml?xml=/money/2006/08/27/ccsoft27.xml) The Telegraph, August 26, 2006
18. ^ **a** **b** Computing Sciences Accreditation Board (28 May 1997). "Computer Science as a Profession" (http://www.csab.org/comp_sci_profession.html) . http://www.csab.org/comp_sci_profession.html. Retrieved 2008-09-01.
19. ^ Committee on the Fundamentals of Computer Science: Challenges and Opportunities, National Research Council (2004). *Computer Science: Reflections on the Field, Reflections from the Field* (http://www.nap.edu/catalog.php?record_id=11106#toc) . National Academies Press. ISBN 978-0-309-09301-9. http://www.nap.edu/catalog.php?record_id=11106#toc.
20. ^ Computer Sciences Accreditation Board (http://www.csab.org/)
21. ^ Clay Mathematics Institute (http://www.claymath.org/millennium/P_vs_NP/) P=NP
22. ^ Communications of the ACM 1(4):p.6
23. ^ Communications of the ACM 2(1):p.4
24. ^ IEEE Computer 28(12):p.136
25. ^ Parnas, David L. (1998). "Software Engineering Programmes are not Computer Science Programmes (http://citeseer.ist.psu.edu/parnas98software.html) ". *Annals of Software Engineering* **6**: 19–37. doi:10.1023/A:1018949113292 (http://dx.doi.org/10.1023%2FA%3A1018949113292) ., p. 19: "Rather than treat software engineering as a subfield of computer science, I treat it as an element of the set, Civil Engineering, Mechanical Engineering, Chemical Engineering, Electrical Engineering, .."

Further reading

- Association for Computing Machinery. 1998 ACM Computing Classification System (http://www.acm.org/class/1998/overview.html) . 1998.
- IEEE Computer Society and the Association for Computing Machinery. Computing Curricula 2001: Computer Science (http://www.computer.org/portal/cms_docs_ieeecs/ieeecs/education/cc2001/cc2001.pdf) . December 15, 2001.
- Peter J. Denning. *Is computer science science? (http://portal.acm.org/citation.cfm?id=1053309&coll=&dl=ACM&CFID=15151515&CFTOKEN=6184618)* , Communications of the ACM, April 2005.
- Donald E. Knuth. *Selected Papers on Computer Science*, CSLI Publications, Cambridge Univ. Press, 1996.
- Peter J. Denning, *Great principles in computing curricula (http://portal.acm.org/citation.cfm?id=971303&dl=ACM&coll=&CFID=15151515&CFTOKEN=6184618)* , Technical Symposium on Computer Science Education, 2004.

External links

- Computer science (http://www.dmoz.org/Computers/Computer_Science//) at the Open Directory Project
- Directory of free university lectures in Computer Science (http://www.lecturefox.com/computerscience/)
- Collection of computer science lectures (http://videolectures.net/Top/Computer_Science/)
- bibliography/ Collection of Computer Science Bibliographies (http://liinwww.ira.uka.de/)
- CS Directory and resources (http://csdir.org/)
- Photographs of computer scientists (http://se.ethz.ch/~meyer/gallery/) (Bertrand Meyer's gallery)

Webcasts

- UCLA Computer Science 1 Freshman Computer Science Seminar Section 1 (http://www.oid.ucla.edu/webcasts/courses/2006-2007/2006fall/cs1)
- Berkeley Introduction to Computers (http://webcast.berkeley.edu/course_details.php?seriesid=1906978395)

Retrieved from "http://en.wikipedia.org/wiki/Computer_science"
Categories: Computer science