all segments of the population are taken into account, so that no one is disenfranchised in terms of access to technology.

CPSR is organized in local chapters and in working groups: Cyber Rights, Civil Liberties, Internet Governance, Education, Ethics, Computers and the Law, and others. Working groups consist of people who discuss a certain topic (usually by e-mail), then produce reports and policy recommendations. CPSR has a total membership of about 2000 persons, 84 percent of whom are under 35 years old and two-thirds of whom are male. Most members are computer professionals who belong to other societies, such as the **Association for Computing Machinery** (ACM) and the **Institute of Electrical and Electronics Engineers** (IEEE).

Similar organizations to CPSR exist in other countries, such as IFIP in Germany (*Informatiker für Frieden und Verantwortung*—Computer Scientists for Peace and Responsibility). Like CPSR, IFIP is composed of regional chapters and working groups and holds a national meeting every year to discuss political issues related to computer technology.

FURTHER READING

Bellin, David, and Gary Chapman. *Computers in Battle: Will They Work?* Boston: Harcourt Brace Jovanovich, 1987.

Valovic, Thomas. *Digital Mythologies: The Hidden Complexities of the Internet.* New Brunswick, N.J.: Rutgers University Press, 2000.

—Raúl Rojas

Computer Science

Although it would be tempting to call computer science (CS) the science of building computers and making them work, there are actually several possible definitions. *Computing science* would be a better name, since it stresses that this is the science that studies computation in all its possible manifestations, the algorithms thereof, and its possible applications. In other languages CS has a more informative name: Informatik (in German), Informática (in Spanish), and Informatique (in French), stressing the fact that it is the science of automatic information processing.

Computer science became a discipline only after the advent of computers. The first professional organization in the field was the **Association for Computing Machinery** (ACM), founded in 1947, only two years after the official presentation of the **ENIAC**. The first programmers and computer scientists came from the fields of mathematics, physics, and other branches of the natural sciences. Although the first CS departments were not started officially until the 1960s, some universities, including Harvard and Princeton, started offering programming courses very early. Especially the Harvard Computation Laboratory, directed by **Howard Aiken** (1900–73), could be considered as the precursor of later CS programs in the United States.

It has been said that the term *computer science* was coined by George Forsythe from Stanford University in the 1960s. The first CS department was established at Purdue University in 1962, and the first person to receive a Ph.D. in CS was Richard Wexelblatt at the University of Pennsylvania in 1965. Over many years, the CS departments at the Massachusetts Institute of Technology, Stanford, Carnegie Mellon University, and the University of California–Berkeley, have consistently been ranked as the best in the United States.

Some authors consider CS to be a type of applied branch of mathematics, others consider it a branch of engineering. Although the discussion might seem trivial, it actually goes to the heart of the matter. In the former case, CS is considered to be a science of abstract computable structures; computer scientists abstract a representation in terms of data structures from a practical problem and develop algorithms to solve the problem. In the latter case, CS as engineering, the emphasis is on the practical approach to the solution of computational problems.

There are four major subdivisions of CS and several subdisciplines in each group. CS is divided into theoretical, technical, practical, and applied CS. Theoretical CS deals with all theoretical aspects related to computational devices. *Computability theory*, for example, examines different hierarchies of machines and its relative computational power. A universal device, such as the **Turing machine** is able to perform any computation that could be performed by a human being provided with paper and pencil (and much patience). But

there are alternatives, such as the lambda calculus or recursive functions, which are also capable of implementing universal computation. Whereas computability theory deals with the possibility of computing something, *complexity theory* deals with the efficiency with which the computation is done. If the computation can be performed in a number of steps that increases only polynomially with the size of the input, the computation is said to be feasible. If the number of steps increases exponentially, we are faced with a difficult computational problem. Complexity theory gives guarantees for the efficiency of algorithms and reduces entire problem classes to other types of easier problems.

Other subdisciplines within theoretical CS are automata theory, computational geometry, graph theory, formal languages, and circuit complexity. *Theoretical CS* is in many cases just another name for traditional topics that previously were investigated in mathematics departments.

Technical CS deals with the construction of computers, computers networks, and computing infrastructure in general. Building modern **very large scale integration** (VLSI) chips is no longer an art, but requires computer automated tools and rigorous standards. A chip containing 10 million transistors can only be designed using large CAD systems that profit from the latest results and methodologies developed at universities.

Practical CS is a more heterogeneous branch of CS, in which practical solutions to many different types of problems are developed. For example, writing large programs requires a special discipline studied in software engineering. This subfield of CS deals with measuring the complexity of programming projects and explores how to subdivide them in modular portions that can be given to a small group or to a single programmer.

Database theory and practice is another area in which computer scientists are active. A database is any computerized repository of information, and there are several strategies to access this information. Relational databases, one alternative for the organization of the information, can be thought of as tables of data accessible through a number of primitive operations.

Other areas of practical CS are **artificial intelligence** (AI) and **robotics**. AI, also called *machine learning*, deals with the means and methods to make computers simulate intelligent behavior. After much hype in the 1980s, the field has now consolidated and its results can be used in the emergent field of mobile autonomous robots.

Finally, applied CS covers research typically done in CS departments, but also outside them or in strong collaboration with specialists from other disciplines. Some examples are geographical information systems, medical informatics, information systems for businesses and the emerging field of bioinformatics, very related to theoretical CS.

In the United States the number of Ph.D.s awarded in CS has been around 1000 every year at the time of this writing. The yearly enrollment of undergraduates rose from 10,000 in 1995 to 20,000 in 1999, reflecting the increased need for CS professionals in industry. Even so, CS departments in the United States were not able to provide all the specialists required by industry, and many of them had to be recruited from abroad.

FURTHER READING

Brookshear, J. Glenn. *Computer Science: An Overview*, 5th ed. Reading, Mass.: Addison-Wesley, 1997.

Dunne, Paul E. *Computability Theory: Concepts and Applications*. New York: Ellis Horwood, 1991.

Fitting, Melvin. *Computability Theory, Semantics, and Logic Programming*. New York: Oxford University Press; Oxford: Clarendon Press, 1987.

Freeman, Peter, and William Aspray. *The Supply of Information Technology Workers in the U.S.* Washington, D.C.: Computing Research Association, 1999.

Irwin, Mary Jane, and Frank Friedman. "1998–1999 Taulbee Survey." *Computer Research News*, Mar. 2000.

Wang, Yingxu, and Graham King. *Software Engineering Processes: Principles and Applications*. Boca Raton, Fla.: CRC Press, 2000.

—Raúl Rojas

Computer Sciences Corporation

Computer Sciences Corporation (CSC) was one of the first companies founded with software development as its primary business and the only computer services company founded prior to 1960 that survived massive changes in the industry during the last four decades of the twentieth century. A computer services