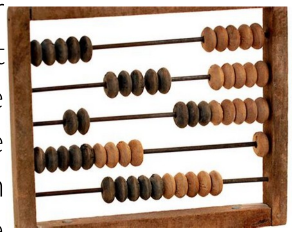
The History of Computing



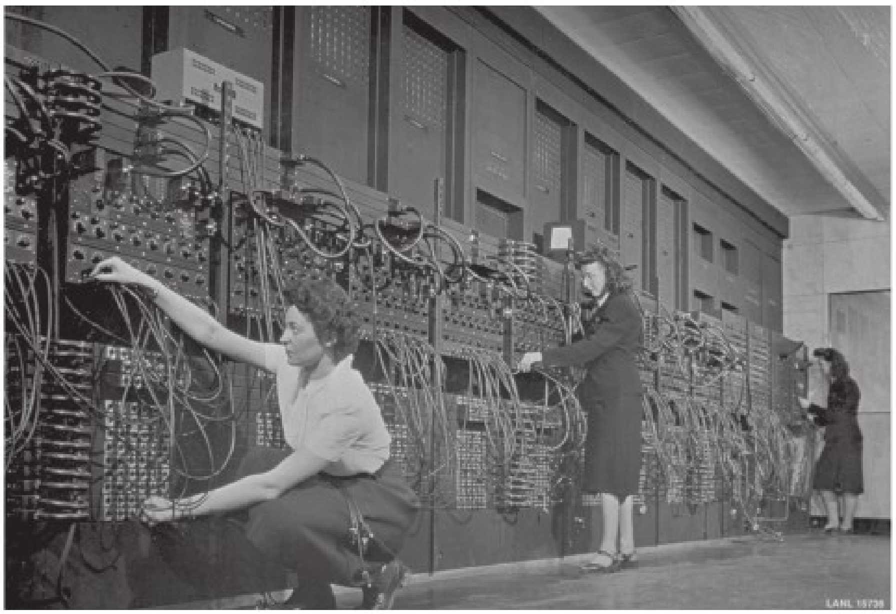
Today's computers have an extensive genealogy. One of the earlier computing devices was the **abacus**. It probably had its roots in ancient China and was used in the early Greek and Roman civilizations. The machine is quite simple, consisting of beads strung on rods that are mounted in a rectangular frame. As the beads are moved back and forth on the rods, their positions represent stored values. It is in the positions of the beads that this "computer" represents and stores data. For control of an algorithm's execution, the machine relies on the human operator. Thus the **abacus alone is merely a data storage system**; it must be combined with a human to create a complete computational machine.

In the time period after the Middle Ages and before the Modern Era, the quest for more **sophisticated computing machines** was seeded. A few inventors began to experiment with the technology of gears. Among these were Wilhelm Schickard (1592-1635), Blaise Pascal (1623–1662) of France, Gottfried Wilhelm Leibniz (1646–1716) of Germany, and Charles Babbage (1792–1871) of England. These machines r**epresented data through gear positioning.** Data was entered mechanically by establishing initial gear positions. Output from Pascal's and Leibniz's machines was achieved by observing the final gear positions. Babbage, on the other hand, envisioned machines that would print results of computations on paper to eliminate the possibility of transcription errors. As for the ability to follow an algorithm, we can see a progression of flexibility in these machines. Pascal's machine was built to perform only addition. Consequently, the appropriate sequence of steps was embedded into the structure of the machine itself. In a similar manner, Leibniz's machine had its **algorithms firmly embedded in its architecture**, although the operator could select from a variety of arithmetic operations it offered. Babbage's Difference Engine (of which only a demonstration model was constructed) could be modified to perform a variety of calculations, but his **Analytical Engine** (never funded for construction) was **designed to read instructions** in the form of holes in paper cards. Thus Babbage's Analytical Engine was programmable. In fact, Augusta Ada Byron (**Ada Lovelace**), who published a paper in which she **demonstrated how Babbage's Analytical Engine could be programmed to perform various computations**, is often identified today as the world's first programmer.

The idea of communicating an algorithm via holes in paper was not originated by Babbage. He got the idea from Joseph Jacquard (1752–1834), who, in 1801, had developed a weaving loom in which the steps to be performed during the weaving process were determined by patterns of holes in large thick cards made of wood (or cardboard). In this manner, the algorithm followed by the loom could be changed easily to produce different woven designs. Another beneficiary of

Jacquard's idea was Herman Hollerith (1860–1929), who applied the concept of representing information as holes in paper cards to speed up the tabulation process in the 1890 U.S. census. (It was this work by Hollerith that led to the creation of IBM.) Such cards ultimately came to be known as punched cards and survived as a popular means of communicating with computers well into the 1970s.

Figure 0.4 Three women operating the ENIAC's (Electronic Numerical Integrator And Computer) main control panel while the machine was at the Moore School. The machine was later moved to the U.S. Army's Ballistics Research Laboratory. (Courtesy U.S. Army.)



Nineteenth-century technology was unable to produce the complex gear-driven machines of Pascal, Leibniz, and Babbage cost-effectively. But with the **advances in electronics in the early 1900s**, this barrier was overcome. Examples of this progress include the electromechanical **Z3** completed in 1941 by Konrad Zuse and the **Mark I**, completed in 1944 at Harvard University by Howard Aiken and a group of IBM engineers. These machines made heavy use of electronically controlled mechanical relays. In this sense they were obsolete almost as soon as they were built, because other researchers were applying the technology of vacuum tubes to construct totally electronic computers. The first of these vacuum tube machines was apparently the Atanasoff-Berry machine, constructed during the period from 1937 to 1941 at Iowa State College (now Iowa State University) by John Atanasoff and his assistant, Clifford Berry. Another was a machine called **Colossus**, built under the direction of Tommy Flowers in England to decode German messages during the latter part of World War II. Other, more flexible machines, such as the **ENIAC** (electronic numerical integrator and calculator) developed by John Mauchly and J. Presper

Eckert, School of Electrical Engineering, University of Pennsylvania, soon followed (Figure 0.4). From that point on, the history of computing machines has been closely linked to advancing technology, including the invention of transistors (for which physicists William Shockley, John Bardeen, and Walter Brattain were awarded a Nobel Prize) and the subsequent development of complete circuits constructed as single units, called integrated circuits (for which Jack Kilby also won a Nobel Prize in physics). With these developments, the **room-sized machines of the 1940s were reduced over the decades to the size of single cabinets.** At the same time, the processing power of computing machines began to double every two years (a trend that has continued to this day). As work on integrated circuitry progressed, many of the components within a computer became readily available on the open market as integrated circuits encased in toy-sized blocks of plastic called chips.

A major step toward popularizing computing was the development of **desktop computers**. The origins of these machines can be traced to the computer hobbyists who built homemade computers from combinations of chips. It was within this "underground" of hobby activity that Steve Jobs and Stephen Wozniak built a commercially viable home computer and, in 1976, established Apple Computer, Inc. (now Apple Inc.) to manufacture and market their products. Other companies that marketed similar products were Commodore, Heathkit, and Radio Shack. Although these products were popular among computer hobbyists, they were not widely accepted by the business community, which continued to look to the well-established IBM and its large mainframe computers for the majority of its computing needs.

In 1981, IBM introduced its first desktop computer, called the personal computer, or PC, whose underlying software was developed by a newly formed company known as Microsoft. The PC was an instant success and legitimized the desktop computer as an established commodity in the minds of the business community. Today, the term PC is widely used to refer to all those machines (from various manufacturers) whose design has evolved from IBM's initial desktop computer, most of which continue to be marketed with software from Microsoft. At times, however, the term PC is used interchangeably with the generic terms desktop or laptop.

The Role of Algorithms

We begin with the most fundamental concept of computer science—that of an algorithm. Informally, an algorithm is a set of steps that defines how a task is performed. For example, there are algorithms for cooking (called recipes), for finding your way through a strange city (more commonly called directions), for operating washing machines (usually displayed on the inside of the washer's lid or perhaps on the wall of a laundromat), for playing music (expressed in the form of sheet music), and for performing magic tricks. Before a machine such as a computer can perform a task, an algorithm for performing that task must be discovered and represented in a form that is compatible with the machine. Such a **representation of an algorithm is called a program**. For the convenience of humans, computer programs are usually printed on paper or displayed on computer screens. For the convenience of machines, programs are encoded in a manner compatible with the technology of the machine. The process of developing a program, encoding it in machine-compatible form, and inserting it into a machine is called programming. Programs, and the algorithms they represent, are collectively referred to as **software**, in contrast to the machinery itself, which is known as **hardware**.

The study of algorithms began as a subject in mathematics. Indeed, the search for algorithms was a significant activity of mathematicians long before the development of today's computers. The goal was to find a single set of directions that described how all problems of a particular type could be solved. One of the best known examples of this early research is the long division algorithm for finding the quotient of two multiple-digit numbers. Another example is the Euclidean algorithm, discovered by the ancient Greek mathematician Euclid, for finding the greatest common divisor of two positive integers. Once an algorithm for performing a task has been found, the performance of that task no longer requires an understanding of the principles on which the algorithm is based. Instead, the performance of the task is reduced to the process of merely following directions. (We can follow the Euclidean algorithm to find a greatest common divisor without understanding why the algorithm works.) In a sense, the intelligence required to solve the problem at hand is encoded in the algorithm. Capturing and conveying intelligence (or at least intelligent behavior) by means of algorithms allows us to build machines that perform useful tasks. Consequently, the level of intelligence displayed by machines is limited by the intelligence that can be conveyed through algorithms. We can construct a machine to perform a task only if an algorithm exists for performing that task. In turn, if no algorithm exists for solving a problem, then the solution of that problem lies beyond the capabilities of machines. Identifying the limitations of algorithmic capabilities solidified as a subject in mathematics in the 1930s with the publication of **Kurt Gödel's incompleteness theorem**. This theorem essentially states that *in any mathematical theory encompassing our traditional arithmetic system, there are* *statements whose truth or falseness cannot be established by algorithmic means*. In short, any complete study of our arithmetic system lies beyond the capabilities of algorithmic activities. This realization shook the foundations of mathematics, and **the study of algorithmic capabilities** that ensued was the beginning of the field known today as computer science. Indeed, it is the study of algorithms that forms the core of computer science.

Augusta Ada Byron

Augusta Ada Byron, Countess of Lovelace, has been the subject of much commentary in the computing community. She lived a somewhat tragic life of less than 37 years (1815–1852) that was complicated by poor health and the fact that she was a nonconformist in a society that limited the professional role of women. Although she was interested in a wide range of science, she concentrated her studies in mathematics. Her interest in "compute science" began when she became fascinated by the machines of Charles Babbage at a demonstration of a prototype of his Difference Engine in 1833. Her contribution to computer science stems from her translation from French into English of a paper discussing Babbage's designs for the Analytical Engine. To this translation, Babbage encouraged her to attach an addendum describing applications of the engine and containing examples of how the engine could be programmed to perform various tasks. Babbage's enthusiasm for Ada Byron's work was apparently motivated by his hope that its publication would lead to financial backing for the construction of his Analytical Engine. (As the daughter of Lord Byron, Ada Byron held celebrity status with potentially significant financial connections.) This backing never materialized, but Ada Byron's addendum has survived and is considered to contain the first examples of computer programs. The degree to which Babbage influenced Ada Byron's work is debated by historians. Some argue that Babbage made major contributions, whereas others contend that he was more of an obstacle than an aid. Nonetheless, Augusta Ada Byron is recognized today as the world's first programmer, a status that was certified by the U.S. Department of Defense when it named a prominent programming language (Ada) in her honor.