Activity 209. Read the article to expand on the interdisciplinary connection and mutual influence that exists between mathematics and computer science.

The roots of modern computer science lie in an interest in rapid computation. Simple mechanical calculators may date back to ancient times; however, it is the work of mathematicians Blaise Pascal (1623–1662) and Gottfried Leibniz (1646–1716) that gave rise to the first practical mechanical calculators. By the mid-19th century, Charles Babbage (1791–1871) had conceptualized and designed mechanical computers that included the essential features (programs, processor, memory, input/output) of the modern digital computer. His motivation was the need for rapid, accurate calculation of statistical tables made necessary by the manufacturing economy of the Industrial Revolution. By the end of the century, the volume of such data had increased to the point where mechanical calculators and tabulators had become the only practical way to keep up.

Mathematically, a computer can be seen as a way to rapidly and automatically execute procedures that have been proven to lead to reliable solutions to a problem. Once computers came on the scene, mathematical principles for verifying or proving algorithms would acquire new practical importance.

By the early 20th century, however, mathematicians were beginning to examine the problem of determining what propositions were provable, and in 1931 Kurt Gödel published a proof that any mathematical system necessarily allowed for the formation of propositions that could not be proven using the axioms of that system. An analogous question was determining what problems were computable. Working independently, two researchers formulated models that could be used to test for computability. Turing's model, in particular, provided a theoretical construct that could, using combinations of a few simple operations, calculate anything that was computable.

By the 1940s, electromechanical (relays) or electronic (tube) switching elements made it possible to build practical high-speed computers. Computer circuit designers could draw upon the advances in symbolic logic in the 19th century. Boolean logic, with its true/false values, would prove ideal for operating computers constructed from on/off switched elements.

The mathematical tools of the previous 150 years could now be used to design systems that could not only calculate but also manipulate symbols and achieve results in higher mathematics.

A variety of mathematical disciplines bear upon the design and use of modern computers. Simple or complex algebra using variables in formulas is at the heart of many programs ranging from financial software to flight simulators.

Geometry, particularly the analytical geometry based upon the coordinate system devised by René Descartes (1596–1650) is fundamental to computer graphics displays, where the screen is divided into X (vertical) and Y (horizontal) axes. Modern graphics systems have added 3D depiction and sophisticated algorithms to allow the rapid display of complex objects. Beyond graphics, the Cartesian insight that converted geometry into algebra makes a variety of geometrical problems accessible to computation, including the finding of optimum paths for circuit design. Design of computer and network architectures also involves the related field of topology. The fascinating field of fractal geometry has found use in computer graphics and data storage techniques.

Aspects of number theory, often considered the most abstract branch of mathematics, have found surprising relevance in computer applications. These include randomization (random number generation) and the factoring of large numbers, which is crucial for cryptography.

Mathematics as a discipline is thus essential to its younger sibling, computer science. In turn, however, computer science and technology have enriched the pursuit of mathematical truth in surprising ways. As early as 1956, a program called Logic Theorist, written by Herbert Simon (1916–2001) and Allen Newell (1927–1992) demonstrated how a program (that is, a collection of algorithms) could prove mathematical propositions given axioms and rules. While these early programs worked on a somewhat hit-or-miss basis, later theorem-solving programs produced solutions different from the standard ones known to mathematicians, and sometimes more elegant. Thus, the computer, which began as an aid to calculation, became an aid to symbol manipulation and to some extent an independent creative source.

(by Harry Henderson, from Encyclopedia of Computer Science and Technology)