# **Additional Reading** Contributors **Article History**

Introduction

**Fast Facts** 

Quizzes

**Videos** 

Media

More

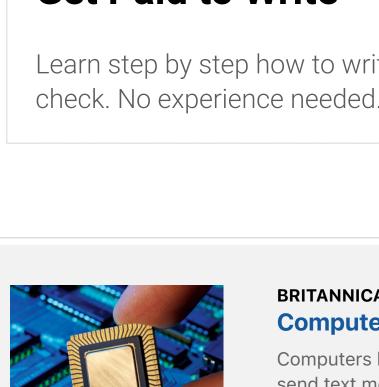
location of the next instruction. Machine language is difficult to read and write, since it does not resemble conventional mathematical notation or human language, and its codes vary from computer to computer. Assembly language is one level above machine language. It uses short mnemonic codes for instructions and allows the programmer to introduce names for blocks of memory that hold data. One might thus write "add pay, total" instead of "0110101100101000" for an instruction that adds two numbers.

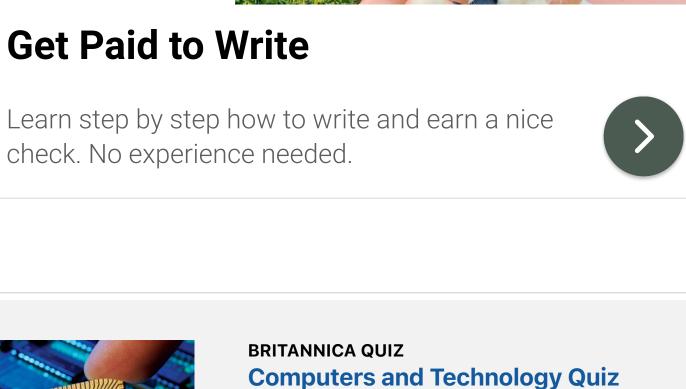
computer can execute directly. The codes are strings of os and 1s, or binary digits ("bits"),

which are frequently converted both from and to hexadecimal (base 16) for human

viewing and modification. Machine language instructions typically use some bits to

represent operations, such as addition, and some to represent operands, or perhaps the





Computers host websites composed of HTML and

quiz and let some technology tally your score and

reveal the contents to you.

Assembly language is designed to be easily translated into machine language. Although

blocks of data may be referred to by name instead of by their machine addresses, assembly

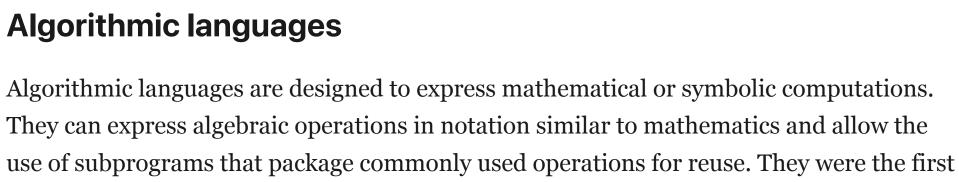
language does not provide more sophisticated means of organizing complex information.

computer architecture. It is useful when such details are important, as in programming a

computer to interact with peripheral devices (printers, scanners, storage devices, and so

Like machine language, assembly language requires detailed knowledge of internal

send text messages as simple as...LOL. Hack into this



high-level languages.

**FORTRAN** 

Its control structures included conditional IF statements, repetitive loops (so-called DO loops), and a GOTO statement that allowed nonsequential execution of program code. FORTRAN made it convenient to have subprograms for common mathematical operations, and built libraries of them.

Find your

FORTRAN was also designed to translate into efficient machine language. It was

immediately successful and continues to evolve.

The first important algorithmic language was FORTRAN (formula translation), designed

in 1957 by an IBM team led by John Backus. It was intended for scientific computations

with real numbers and collections of them organized as one- or multidimensional arrays.

**ALGOL** 

forth).

ALGOL (algorithmic language) was designed by a committee of American and European computer scientists during 1958–60 for publishing algorithms, as well as for doing computations. Like LISP (described in the next section), ALGOL had recursive subprograms—procedures that could invoke themselves to solve a problem by reducing it to a smaller problem of the same kind. ALGOL introduced block structure, in which a program is composed of blocks that might contain both data and instructions and have the same structure as an entire program. Block structure became a powerful tool for building large programs out of small components.

The C programming language was developed in 1972 by Dennis Ritchie and Brian Kernighan at the AT&T Corporation for programming computer operating systems. Its capacity to structure data and programs through the composition of smaller units is comparable to that of ALGOL. It uses a compact notation and provides the programmer with the ability to operate with the addresses of data as well as with their values. This ability is important in systems programming, and C shares with assembly language the

descendant C++, remains one of the most common languages.

# **Business-oriented languages**

modern languages.

**SQL** 

**Pascal** 

education.

Logo

C

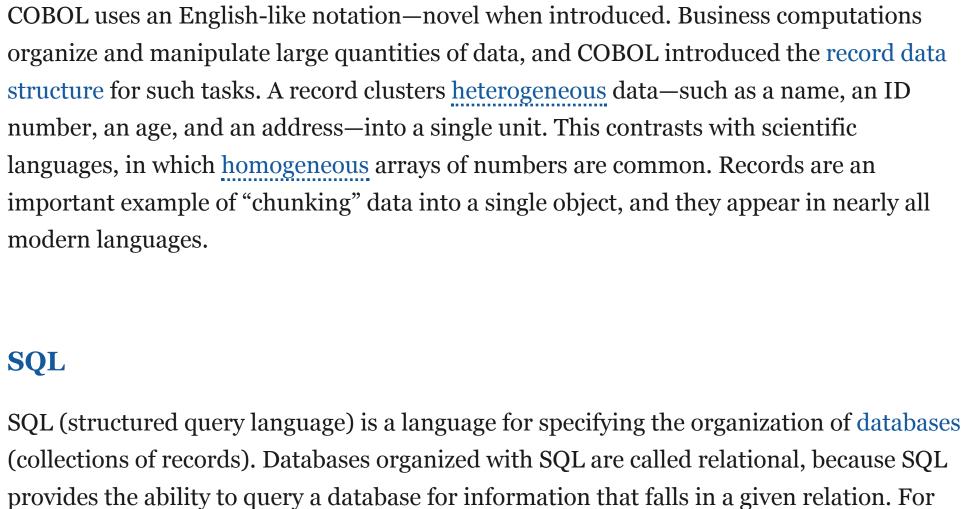
**COBOL** COBOL (common business oriented language) has been heavily used by businesses since its inception in 1959. A committee of computer manufacturers and users and U.S. government organizations established CODASYL (Committee on Data Systems and Languages) to develop and oversee the language standard in order to ensure its portability across diverse systems.

Find your

somewhere

Hotels.com

perfect



**BOOK NOW** 

**Education-oriented languages BASIC** BASIC (beginner's all-purpose symbolic instruction code) was designed at Dartmouth College in the mid-1960s by John Kemeny and Thomas Kurtz. It was intended to be easy to learn by novices, particularly non-computer science majors, and to run well on a timesharing computer with many users. It had simple data structures and notation and it was interpreted: a BASIC program was translated line-by-line and executed as it was

Its small size and simplicity also made BASIC a popular language for early personal

computers. Its recent forms have adopted many of the data and control structures of other

contemporary languages, which makes it more powerful but less convenient for beginners.

translated, which made it easy to locate programming errors.

### structures without GOTO statements. Although Pascal resembled ALGOL in notation, it provided the ability to define data types with which to organize complex information, a feature beyond the capabilities of ALGOL as well as FORTRAN and COBOL. User-defined data types allowed the programmer to introduce names for complex data, which the

language translator could then check for correct usage before running a program.

familiarity, clarity, and security, it was used for production software as well as for

During the late 1970s and '80s, Pascal was one of the most widely used languages for

programming instruction. It was available on nearly all computers, and, because of its

About 1970 Niklaus Wirth of Switzerland designed Pascal to teach structured

programming, which emphasized the orderly use of conditional and loop control

Logo originated in the late 1960s as a simplified LISP dialect for education; Seymour Papert and others used it at MIT to teach mathematical thinking to schoolchildren. It had a more conventional syntax than LISP and featured "turtle graphics," a simple method for generating computer graphics. (The name came from an early project to program a turtlelike robot.) Turtle graphics used body-centred instructions, in which an object was moved around a screen by commands, such as "left 90" and "forward," that specified actions relative to the current position and orientation of the object rather than in terms of a fixed framework. Together with recursive routines, this technique made it easy to program intricate and attractive patterns.

Hypertalk was designed as "programming for the rest of us" by Bill Atkinson for Apple's

section), Apple did not develop it for other computer platforms and let it languish; as

Apple's market share declined in the 1990s, a new cross-platform way of displaying

multimedia left Hypertalk all but obsolete (see the section World Wide Web display

Macintosh. Using a simple English-like syntax, Hypertalk enabled anyone to combine text,

### graphics, and audio quickly into "linked stacks" that could be navigated by clicking with a mouse on standard buttons supplied by the program. Hypertalk was particularly popular among educators in the 1980s and early '90s for classroom multimedia presentations. Although Hypertalk had many features of object-oriented languages (described in the next

another.

**Hypertalk** 

languages). **Object-oriented languages** Object-oriented languages help to manage complexity in large programs. Objects package data and the operations on them so that only the operations are publicly accessible and internal details of the data structures are hidden. This information hiding made largescale programming easier by allowing a programmer to think about each part of the program in isolation. In addition, objects may be derived from more general ones, "inheriting" their capabilities. Such an object hierarchy made it possible to define specialized objects without repeating all that is in the more general ones. Object-oriented programming began with the Simula language (1967), which added

information hiding to ALGOL. Another influential object-oriented language was Smalltalk

(1980), in which a program was a set of objects that interacted by sending messages to one

by adding objects to it while preserving the efficiency of C programs. It has been one of the

# C++ The C++ language, developed by Bjarne Stroustrup at AT&T in the mid-1980s, extended C

most important languages for both education and industrial programming. Large parts of many operating systems were written in C++. C++, along with Java, has become popular for developing commercial software packages that incorporate multiple interrelated applications. C++ is considered one of the fastest languages and is very close to low-level languages, thus allowing complete control over memory allocation and management. This very feature and its many other capabilities also make it one of the most difficult languages to learn and handle on a large scale. C#

C# (pronounced C sharp like the musical note) was developed by Anders Hejlsberg at

Microsoft in 2000. C# has a syntax similar to that of C and C++ and is often used for

## developing games and applications for the Microsoft Windows operating system. Ada

Ada was named for Augusta Ada King, countess of Lovelace, who was an assistant to the 19th-century English inventor Charles Babbage, and is sometimes called the first computer programmer. Ada, the language, was developed in the early 1980s for the U.S. Department of Defense for large-scale programming. It combined Pascal-like notation with the ability to package operations and data into independent modules. Its first form,

# Ada 83, was not fully object-oriented, but the subsequent Ada 95 provided objects and the

mobile telephones.

Library account.

ability to construct hierarchies of them. While no longer mandated for use in work for the Department of Defense, Ada remains an effective language for engineering large programs. Java In the early 1990s Java was designed by Sun Microsystems, Inc., as a programming language for the World Wide Web (WWW). Although it resembled C++ in appearance, it was object-oriented. In particular, Java dispensed with lower-level features, including the ability to manipulate data addresses, a capability that is neither desirable nor useful in programs for distributed systems. In order to be portable, Java programs are translated by a Java Virtual Machine specific to each computer platform, which then executes the Java program. In addition to adding interactive capabilities to the Internet through Web "applets," Java has been widely used for programming small and portable devices, such as

Load Next Page 🗸

Beachfront