

Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Allen Downey, in his book *How To Think Like A Computer Scientist*, writes: Many computer languages provide a mechanism to call functions provided by shared libraries. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages. FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research. Whatever the approach to development may be, the final program must satisfy some fundamental properties. It is very difficult to determine what are the most popular modern programming languages. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display. Programmable devices have existed for centuries. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Different programming languages support different styles of programming (called programming paradigms). While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se. Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. Computer programmers are those who write computer software. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. It is usually easier to code in "high-level" languages than in "low-level" ones. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in *A Manuscript on Deciphering Cryptographic Messages*.