

The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.

Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Many applications use a mix of several languages in their construction and use. Programming languages are essential for software development. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. 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. Techniques like Code refactoring can enhance readability. 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. 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. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Unreadable code often leads to bugs, inefficiencies, and duplicated code. These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics. Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute. However, readability is more than just programming style. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. 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. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Whatever the approach to development may be, the final program must satisfy some fundamental properties.