

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. 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. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Many applications use a mix of several languages in their construction and use. Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. 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 Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in *A Manuscript on Deciphering Cryptographic Messages*. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Programs were mostly entered using punched cards or paper tape. Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Use of a static code analysis tool can help detect some possible problems. One approach popular for requirements analysis is Use Case analysis. 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. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. Techniques like Code refactoring can enhance readability. For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash. One approach popular for requirements analysis is Use Case analysis.