

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. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Programming languages are essential for software development. Whatever the approach to development may be, the final program must satisfy some fundamental properties. 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. Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. 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. 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. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in *A Manuscript on Deciphering Cryptographic Messages*. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. However, readability is more than just programming style. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams. 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. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory. 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. 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. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging).