

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 academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems. Integrated development environments (IDEs) aim to integrate all such help. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. 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. 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. The choice of language used is subject to many considerations, such as company policy, suitability to task, availability of third-party packages, or individual preference. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Code-breaking algorithms have also existed for centuries. 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. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. 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. Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. 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. The first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine. 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. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. There are many approaches to the Software development process. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in *A Manuscript on Deciphering Cryptographic Messages*.