

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. 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. Use of a static code analysis tool can help detect some possible problems. Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.

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. 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. However, readability is more than just programming style. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. There exist a lot of different approaches for each of those tasks. One approach popular for requirements analysis is Use Case analysis. Code-breaking algorithms have also existed for centuries. 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. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. There are many approaches to the Software development process. 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. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them.