

Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. 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. Ideally, the programming language best suited for the task at hand will be selected. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). 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. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. It affects the aspects of quality above, including portability, usability and most importantly maintainability. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Code-breaking algorithms have also existed for centuries. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Scripting and breakpointing is also part of this process. Computer programmers are those who write computer software. 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. Use of a static code analysis tool can help detect some possible problems. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Following a consistent programming style often helps readability. The following properties are among the most important: In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code. 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. 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. 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. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. 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. 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.