

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. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Use of a static code analysis tool can help detect some possible problems. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Programmable devices have existed for centuries. Their jobs usually involve: Although programming has been presented in the media as a somewhat mathematical subject, some research shows that good programmers have strong skills in natural human languages, and that learning to code is similar to learning a foreign language. Normally the first step in debugging is to attempt to reproduce the problem. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). 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. 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. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language. 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. Techniques like Code refactoring can enhance readability. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Scripting and breakpointing is also part of this process. Many applications use a mix of several languages in their construction and use. 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. There are many approaches to the Software development process. Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display. 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. 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.