

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. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Also, specific user environment and usage history can make it difficult to reproduce the problem. 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. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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. One approach popular for requirements analysis is Use Case analysis. 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. Programming languages are essential for software development. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. However, readability is more than just programming style. Some of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research. 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. 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. 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. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. 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. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Integrated development environments (IDEs) aim to integrate all such help. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. Scripting and breakpointing is also part of this process.