

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. One approach popular for requirements analysis is Use Case analysis. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Scripting and breakpointing is also part of this process. Ideally, the programming language best suited for the task at hand will be selected. 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. 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. Different programming languages support different styles of programming (called programming paradigms). As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). 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. 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. Ideally, the programming language best suited for the task at hand will be selected. 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. Following a consistent programming style often helps readability. Programmable devices have 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. 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. Techniques like Code refactoring can enhance readability. 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 the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. Scripting and breakpointing is also part of this process.