

Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. Techniques like Code refactoring can enhance readability. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. Many applications use a mix of several languages in their construction and use. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Following a consistent programming style often helps readability. 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. 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. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. 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. 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. 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. 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. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). 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 the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. 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. However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory. For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software. However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages. Also, specific user environment and usage history can make it difficult to reproduce the problem. 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. 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.