

It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. 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. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Integrated development environments (IDEs) aim to integrate all such help. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Integrated development environments (IDEs) aim to integrate all such help. 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. 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. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. Programmable devices have existed for centuries. For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash. Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages. Programs were mostly entered using punched cards or paper tape. Normally the first step in debugging is to attempt to reproduce the problem. It is usually easier to code in "high-level" languages than in "low-level" ones. Computer programmers are those who write computer software. 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. It affects the aspects of quality above, including portability, usability and most importantly maintainability. 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.