

Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Use of a static code analysis tool can help detect some possible problems. Techniques like Code refactoring can enhance readability. 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. Different programming languages support different styles of programming (called programming paradigms). Programmable devices have existed for centuries. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. 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. 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. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Ideally, the programming language best suited for the task at hand will be selected. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Many applications use a mix of several languages in their construction and use. 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. It is usually easier to code in "high-level" languages than in "low-level" ones. 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. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Programming languages are essential for software development. Computer programmers are those who write computer software. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.