

Integrated development environments (IDEs) aim to integrate all such help. 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. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. It is very difficult to determine what are the most popular modern programming languages. Integrated development environments (IDEs) aim to integrate all such help. Computer programmers are those who write computer software. Programs were mostly entered using punched cards or paper tape. 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. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. 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. Following a consistent programming style often helps readability. Normally the first step in debugging is to attempt to reproduce the problem. Allen Downey, in his book *How To Think Like A Computer Scientist*, writes: Many computer languages provide a mechanism to call functions provided by shared libraries. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. 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. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Also, specific user environment and usage history can make it difficult to reproduce the problem. 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. Different programming languages support different styles of programming (called programming paradigms). There exist a lot of different approaches for each of those tasks. 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. 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.