

However, readability is more than just programming style. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Also, specific user environment and usage history can make it difficult to reproduce the problem. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Ideally, the programming language best suited for the task at hand will be selected. 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. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Also, specific user environment and usage history can make it difficult to reproduce the problem. However, readability is more than just programming style. 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. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). 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. There are many approaches to the Software development process. There exist a lot of different approaches for each of those tasks. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Many applications use a mix of several languages in their construction and use. Integrated development environments (IDEs) aim to integrate all such help. Techniques like Code refactoring can enhance readability. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Different programming languages support different styles of programming (called programming paradigms). Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL). Different programming languages support different styles of programming (called programming paradigms). Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. 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.