

The following properties are among the most important: In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code. Unreadable code often leads to bugs, inefficiencies, and duplicated code. 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. 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. 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. 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. 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). 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. Programming languages are essential for software development. Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers. Programmable devices have existed for centuries. 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. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. 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. Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Computer programmers are those who write computer software. Integrated development environments (IDEs) aim to integrate all such help. 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. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. Unreadable code often leads to bugs, inefficiencies, and duplicated code. It affects the aspects of quality above, including portability, usability and most importantly maintainability. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). 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. There exist a lot of different approaches for each of those tasks.