

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. 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). Programmable devices have existed for centuries. Techniques like Code refactoring can enhance readability. 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. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Use of a static code analysis tool can help detect some possible problems. While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Programmable devices have existed for centuries. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic. However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se. 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. 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. Normally the first step in debugging is to attempt to reproduce the problem. 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. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Programs were mostly entered using punched cards or paper tape. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Scripting and breakpointing is also part of this process. Also, specific user environment and usage history can make it difficult to reproduce the problem. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process.