

It affects the aspects of quality above, including portability, usability and most importantly maintainability. One approach popular for requirements analysis is Use Case analysis. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. It is very difficult to determine what are the most popular modern programming languages. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display. FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research. However, readability is more than just programming style. Also, specific user environment and usage history can make it difficult to reproduce the problem. Integrated development environments (IDEs) aim to integrate all such help. Programming languages are essential for software development. It is very difficult to determine what are the most popular modern programming languages. Ideally, the programming language best suited for the task at hand will be selected. Normally the first step in debugging is to attempt to reproduce the problem. 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. There are many approaches to the Software development process. 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. 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. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses. Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances. 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. Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.