Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute. 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. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. Different programming languages support different styles of programming (called programming paradigms). Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. There are many approaches to the Software development process. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. It affects the aspects of quality above, including portability, usability and most importantly maintainability. 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. Some of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. Techniques like Code refactoring can enhance readability. Some of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. Integrated development environments (IDEs) aim to integrate all such help. 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. Programs were mostly entered using punched cards or paper tape. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. 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. In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Techniques like Code refactoring can enhance readability. Whatever the approach to development may be, the final program must satisfy some fundamental properties.