He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Techniques like Code refactoring can enhance readability. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. There exist a lot of different approaches for each of those tasks. It is usually easier to code in "high-level" languages than in "low-level" ones. Different programming languages support different styles of programming (called programming paradigms). Code-breaking algorithms have also existed for centuries. Their jobs usually involve: Although programming has been presented in the media as a somewhat mathematical subject, some research shows that good programmers have strong skills in natural human languages, and that learning to code is similar to learning a foreign language. 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. 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. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Following a consistent programming style often helps readability. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. 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. Programmable devices have existed for centuries. 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. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Scripting and breakpointing is also part of this process. 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. 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. Whatever the approach to development may be, the final program must satisfy some fundamental properties. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.