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. 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. 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. 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. Use of a static code analysis tool can help detect some possible problems. 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. Use of a static code analysis tool can help detect some possible problems. Integrated development environments (IDEs) aim to integrate all such help. 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. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more 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. 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. There are many approaches to the Software development process. Unreadable code often leads to bugs, inefficiencies, and duplicated code. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Techniques like Code refactoring can enhance readability. 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. Use of a static code analysis tool can help detect some possible problems. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Scripting and breakpointing is also part of this process. 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. 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. Many applications use a mix of several languages in their construction and use. Ideally, the programming language best suited for the task at hand will be selected.