High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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. Normally the first step in debugging is to attempt to reproduce the problem. 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. 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. Techniques like Code refactoring can enhance readability. 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. Normally the first step in debugging is to attempt to reproduce the problem. It affects the aspects of quality above, including portability, usability and most importantly maintainability. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. There are many approaches to the Software development process. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Integrated development environments (IDEs) aim to integrate all such help. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Integrated development environments (IDEs) aim to integrate all such help. Following a consistent programming style often helps readability. Code-breaking algorithms have also existed for centuries. There exist a lot of different approaches for each of those tasks. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers. There are many approaches to the Software development process. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Different programming languages support different styles of programming (called programming paradigms).