However, readability is more than just programming style. For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software. Scripting and breakpointing is also part of this process. 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. It affects the aspects of quality above, including portability, usability and most importantly maintainability. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software. There exist a lot of different approaches for each of those tasks. Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. Following a consistent programming style often helps readability. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Programs were mostly entered using punched cards or paper tape. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Ideally, the programming language best suited for the task at hand will be selected. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. 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. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Scripting and breakpointing is also part of this process. One approach popular for requirements analysis is Use Case analysis.