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. There are many approaches to the Software development process. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Integrated development environments (IDEs) aim to integrate all such help. 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. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. The first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Integrated development environments (IDEs) aim to integrate all such help. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). 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. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). One approach popular for requirements analysis is Use Case analysis. Different programming languages support different styles of programming (called programming paradigms). Following a consistent programming style often helps readability. 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 text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. 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. However, readability is more than just programming style. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. It is usually easier to code in "high-level" languages than in "low-level" ones.