It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. However, readability is more than just programming style. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. Use of a static code analysis tool can help detect some possible problems. Integrated development environments (IDEs) aim to integrate all such help. Ideally, the programming language best suited for the task at hand will be selected. Code-breaking algorithms have also existed for centuries. There exist a lot of different approaches for each of those tasks. Programming languages are essential for software development. While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se. 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. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. 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. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. 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. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses. Code-breaking algorithms have also existed for centuries. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL).