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. One approach popular for requirements analysis is Use Case analysis. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years. The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems. There are many approaches to the Software development process. Computer programmers are those who write computer software. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. There exist a lot of different approaches for each of those tasks. There are many approaches to the Software development process. Techniques like Code refactoring can enhance readability. One approach popular for requirements analysis is Use Case analysis. 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. Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code. FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research. 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. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). 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. Whatever the approach to development may be, the final program must satisfy some fundamental properties. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. 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. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.