Ideally, the programming language best suited for the task at hand will be selected. Also, specific user environment and usage history can make it difficult to reproduce the problem. 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. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Code-breaking algorithms have also existed for centuries. Also, specific user environment and usage history can make it difficult to reproduce the problem. Many applications use a mix of several languages in their construction and use. It is usually easier to code in "high-level" languages than in "low-level" ones. Code-breaking algorithms have also existed for centuries. One approach popular for requirements analysis is Use Case analysis. 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. 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. Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Allen Downey, in his book How To Think Like A Computer Scientist, writes: Many computer languages provide a mechanism to call functions provided by shared libraries. 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. It is usually easier to code in "high-level" languages than in "low-level" ones. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. Many applications use a mix of several languages in their construction and use. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. There exist a lot of different approaches for each of those tasks.