Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Ideally, the programming language best suited for the task at hand will be selected. 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). 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. There are many approaches to the Software development process. Programmable devices have existed for centuries. Programs were mostly entered using punched cards or paper tape. It affects the aspects of quality above, including portability, usability and most importantly maintainability. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. Many applications use a mix of several languages in their construction and use. 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. 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. One approach popular for requirements analysis is Use Case analysis. 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. For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Programmable devices have existed for centuries. For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the 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. 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. Also, specific user environment and usage history can make it difficult to reproduce the problem. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.