

Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. 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. Ideally, the programming language best suited for the task at hand will be selected. Programmable devices have existed for centuries. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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. 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. Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic. Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display. Techniques like Code refactoring can enhance readability. 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. Also, specific user environment and usage history can make it difficult to reproduce the problem. 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). It is usually easier to code in "high-level" languages than in "low-level" ones. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Also, specific user environment and usage history can make it difficult to reproduce the problem. There are many approaches to the Software development process. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Ideally, the programming language best suited for the task at hand will be selected. Integrated development environments (IDEs) aim to integrate all such help. However, readability is more than just programming style. It is very difficult to determine what are the most popular modern programming languages.