

However, readability is more than just programming style. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. 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. Scripting and breakpointing is also part of this process. Normally the first step in debugging is to attempt to reproduce the problem. 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. There exist a lot of different approaches for each of those tasks. 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. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. 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. Ideally, the programming language best suited for the task at hand will be selected. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. 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. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. Following a consistent programming style often helps readability. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. It is very difficult to determine what are the most popular modern programming languages. 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. It is usually easier to code in "high-level" languages than in "low-level" ones.