

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. One approach popular for requirements analysis is Use Case analysis. Some of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. It is very difficult to determine what are the most popular modern programming languages. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. There exist a lot of different approaches for each of those tasks. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Following a consistent programming style often helps readability. Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages. Many applications use a mix of several languages in their construction and use. 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. Some of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. 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. 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. 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. Scripting and breakpointing is also part of this process. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Normally the first step in debugging is to attempt to reproduce the problem. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. Use of a static code analysis tool can help detect some possible problems. 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*. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.