

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. 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. However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Techniques like Code refactoring can enhance readability. There are many approaches to the Software development process. Whatever the approach to development may be, the final program must satisfy some fundamental properties. However, with the concept of the stored-program computer introduced in 1949, both programs and data were stored and manipulated in the same way in computer memory. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages. For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. One approach popular for requirements analysis is Use Case analysis. 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. Their jobs usually involve: Although programming has been presented in the media as a somewhat mathematical subject, some research shows that good programmers have strong skills in natural human languages, and that learning to code is similar to learning a foreign language. The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems. There are many approaches to the Software development process. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. 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. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form.