

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. Scripting and breakpointing is also part of this 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. Use of a static code analysis tool can help detect some possible problems. 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. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Techniques like Code refactoring can enhance readability. 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. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. Normally the first step in debugging is to attempt to reproduce the problem. Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language. Integrated development environments (IDEs) aim to integrate all such help. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. 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. Integrated development environments (IDEs) aim to integrate all such help. 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. It affects the aspects of quality above, including portability, usability and most importantly maintainability. 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. 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*. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process.