

After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Normally the first step in debugging is to attempt to reproduce the problem. These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics. Code-breaking algorithms have also existed for centuries. 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. 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. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. 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. Computer programmers are those who write computer software. 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. It is usually easier to code in "high-level" languages than in "low-level" ones. 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. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Ideally, the programming language best suited for the task at hand will be selected. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Scripting and breakpointing is also part of this process. 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. Different programming languages support different styles of programming (called programming paradigms). Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). Normally the first step in debugging is to attempt to reproduce the problem. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.