

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. 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. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. In 1206, the Arab engineer Al-Jazari invented a programmable drum machine where a musical mechanical automaton could be made to play different rhythms and drum patterns, via pegs and cams. Unreadable code often leads to bugs, inefficiencies, and duplicated code. 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. 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. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). 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. Unreadable code often leads to bugs, inefficiencies, and duplicated code. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. It is usually easier to code in "high-level" languages than in "low-level" ones. 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. Unreadable code often leads to bugs, inefficiencies, and duplicated code. However, readability is more than just programming style. Scripting and breakpointing is also part of this process. Many applications use a mix of several languages in their construction and use. Unreadable code often leads to bugs, inefficiencies, and duplicated code. 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. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Following a consistent programming style often helps readability. Use of a static code analysis tool can help detect some possible problems. 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.