

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. Normally the first step in debugging is to attempt to reproduce the problem. 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. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. 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. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. There are many approaches to the Software development process. There are many approaches to the Software development process. 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. Normally the first step in debugging is to attempt to reproduce the problem. 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. Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. 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. Normally the first step in debugging is to attempt to reproduce the problem. Use of a static code analysis tool can help detect some possible problems. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Integrated development environments (IDEs) aim to integrate all such help. 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. Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute. It affects the aspects of quality above, including portability, usability and most importantly maintainability.