

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. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. The following properties are among the most important: In computer programming, readability refers to the ease with which a human reader can comprehend the purpose, control flow, and operation of source code. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm. 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. Normally the first step in debugging is to attempt to reproduce the problem. 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. 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. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Following a consistent programming style often helps readability. However, readability is more than just programming style. 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. However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. 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. Different programming languages support different styles of programming (called programming paradigms). 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. 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. 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. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.