A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. There exist a lot of different approaches for each of those tasks. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. 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. Some of these factors include: The presentation aspects of this (such as indents, line breaks, color highlighting, and so on) are often handled by the source code editor, but the content aspects reflect the programmer's talent and skills. Also, specific user environment and usage history can make it difficult to reproduce the problem. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Techniques like Code refactoring can enhance readability. Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years. Programs were mostly entered using punched cards or paper tape. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Many applications use a mix of several languages in their construction and use. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Computer programmers are those who write computer software. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. 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. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Following a consistent programming style often helps readability. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. 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. 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.