

Also, specific user environment and usage history can make it difficult to reproduce the problem. 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. 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. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. Ideally, the programming language best suited for the task at hand will be selected. 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. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. 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. 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. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). 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. New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation). In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. 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. Use of a static code analysis tool can help detect some possible problems. 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. Proficient programming usually requires expertise in several different subjects, including knowledge of the application domain, details of programming languages and generic code libraries, specialized algorithms, and formal logic. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.