

Whatever the approach to development may be, the final program must satisfy some fundamental properties. 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. Techniques like Code refactoring can enhance readability. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. 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. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. Scripting and breakpointing is also part of this process. The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). They are the building blocks for all software, from the simplest applications to the most sophisticated ones. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software. Programming languages are essential for software development. Normally the first step in debugging is to attempt to reproduce the problem. 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. 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. However, Charles Babbage had already written his first program for the Analytical Engine in 1837. 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. Following a consistent programming style often helps readability. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. It affects the aspects of quality above, including portability, usability and most importantly maintainability. 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.