

However, Charles Babbage had already written his first program for the Analytical Engine in 1837. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. One approach popular for requirements analysis is Use Case analysis. In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in *A Manuscript on Deciphering Cryptographic Messages*. 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. For example, when a bug in a compiler can make it crash when parsing some large source file, a simplification of the test case that results in only few lines from the original source file can be sufficient to reproduce the same crash. It is very difficult to determine what are the most popular modern programming languages. It is very difficult to determine what are the most popular modern programming languages. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code. 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. 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. 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, 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. Scripting and breakpointing is also part of this process. Normally the first step in debugging is to attempt to reproduce the problem. Computer programmers are those who write computer software. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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. 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. 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. 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. Different programming languages support different styles of programming (called programming paradigms). 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.