

The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Techniques like Code refactoring can enhance readability. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. 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. FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research. 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. 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. Unreadable code often leads to bugs, inefficiencies, and duplicated code. 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. 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. 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. After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug. Programming languages are essential for software development. Computer programmers are those who write computer software. 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. 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. 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. A similar technique used for database design is Entity-Relationship Modeling (ER Modeling). The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging). Computer programmers are those who write computer software. Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.