Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. Computer programmers are those who write computer software. It is usually easier to code in "high-level" languages than in "low-level" ones. Integrated development environments (IDEs) aim to integrate all such help. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Later a control panel (plug board) added to his 1906 Type I Tabulator allowed it to be programmed for different jobs, and by the late 1940s, unit record equipment such as the IBM 602 and IBM 604, were programmed by control panels in a similar way, as were the first electronic computers. Normally the first step in debugging is to attempt to reproduce the problem. 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. 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. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Integrated development environments (IDEs) aim to integrate all such help. Normally the first step in debugging is to attempt to reproduce the problem. 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. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. However, readability is more than just programming style. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design 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). High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. 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.