Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Computer programmers are those who write computer software. Scripting and breakpointing is also part of this process. 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. 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. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. 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. 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. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Many applications use a mix of several languages in their construction and use. Whatever the approach to development may be, the final program must satisfy some fundamental properties. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. 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. 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. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. 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. Also, specific user environment and usage history can make it difficult to reproduce the problem. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).