Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. Techniques like Code refactoring can enhance readability. Methods of measuring programming language popularity include: counting the number of job advertisements that mention the language, the number of books sold and courses teaching the language (this overestimates the importance of newer languages), and estimates of the number of existing lines of code written in the language (this underestimates the number of users of business languages such as COBOL). Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). 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. One approach popular for requirements analysis is Use Case analysis. Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists. Scripting and breakpointing is also part of this process. Integrated development environments (IDEs) aim to integrate all such help. 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. A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. Programming languages are essential for software development. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. It is very difficult to determine what are the most popular modern programming languages. In 1801, the Jacquard loom could produce entirely different weaves by changing the "program" – a series of pasteboard cards with holes punched in them. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. It is very difficult to determine what are the most popular modern programming languages. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute. This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.