It affects the aspects of quality above, including portability, usability and most importantly maintainability. Unreadable code often leads to bugs, inefficiencies, and duplicated code. 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. 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. Different programming languages support different styles of programming (called programming paradigms). It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Programmable devices have existed for centuries. Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA). 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. Whatever the approach to development may be, the final program must satisfy some fundamental properties. 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. 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. Many programmers use forms of Agile software development where the various stages of formal software development are more integrated together into short cycles that take a few weeks rather than years. By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input. 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. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Computer programmers are those who write computer software. Different programming languages support different styles of programming (called programming paradigms). Programmable devices have existed for centuries. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. Techniques like Code refactoring can enhance readability.