The purpose of programming is to find a sequence of instructions that will automate the performance of a task (which can be as complex as an operating system) on a computer, often for solving a given problem.  
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.  
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.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers.  
Normally the first step in debugging is to attempt to reproduce the problem.  
However, while these might be considered part of the programming process, often the term software development is more likely used for this larger overall process – whereas the terms programming, implementation, and coding tend to be focused on the actual writing of code.  
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.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
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.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
 Different programming languages support different styles of programming (called programming paradigms).  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.