He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
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.  
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.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
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.  
One approach popular for requirements analysis is Use Case analysis.  
There are many approaches to the Software development process.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
Relatedly, software engineering combines engineering techniques and principles with software development.  
Ideally, the programming language best suited for the task at hand will be selected.  
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.