Use of a static code analysis tool can help detect some possible problems..  
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.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.  
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.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
 It is very difficult to determine what are the most popular modern programming languages.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
Integrated development environments (IDEs) aim to integrate all such help.  
 Programs were mostly entered using punched cards or paper tape.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
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).