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.  
Use of a static code analysis tool can help detect some possible problems.  
One approach popular for requirements analysis is Use Case analysis.  
As early as the 9th century, a programmable music sequencer was invented by the Persian Banu Musa brothers, who described an automated mechanical flute player in the Book of Ingenious Devices.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
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.  
Use of a static code analysis tool can help detect some possible problems.  
Many applications use a mix of several languages in their construction and use.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
This is interpreted into machine code.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Programming involves tasks such as analysis, generating algorithms, profiling algorithms' accuracy and resource consumption, and the implementation of algorithms (usually in a particular programming language, commonly referred to as coding).  
There exist a lot of different approaches for each of those tasks.  
 Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation.