He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
This is interpreted into machine code.  
Expert programmers are familiar with a variety of well-established algorithms and their respective complexities and use this knowledge to choose algorithms that are best suited to the circumstances.  
Languages form an approximate spectrum from "low-level" to "high-level"; "low-level" languages are typically more machine-oriented and faster to execute, whereas "high-level" languages are more abstract and easier to use but execute less quickly.  
Proficient programming thus usually requires expertise in several different subjects, including knowledge of the application domain, specialized algorithms, and formal logic.  
Techniques like Code refactoring can enhance readability.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
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.  
Many applications use a mix of several languages in their construction and use.  
 Different programming languages support different styles of programming (called programming paradigms).  
This is interpreted into machine code.  
 These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics.  
 Tasks accompanying and related to programming include testing, debugging, source code maintenance, implementation of build systems, and management of derived artifacts, such as the machine code of computer programs.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
Many applications use a mix of several languages in their construction and use.