He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.  
FORTRAN, the first widely used high-level language to have a functional implementation, came out in 1957, and many other languages were soon developed—in particular, COBOL aimed at commercial data processing, and Lisp for computer research.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Also, those involved with software development may at times engage in reverse engineering, which is the practice of seeking to understand an existing program so as to re-implement its function in some way.  
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).  
Ideally, the programming language best suited for the task at hand will be selected.  
Provided the functions in a library follow the appropriate run-time conventions (e.g., method of passing arguments), then these functions may be written in any other language.  
Use of a static code analysis tool can help detect some possible problems.  
Techniques like Code refactoring can enhance readability.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
To produce machine code, the source code must either be compiled or transpiled.  
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.  
 Popular modeling techniques include Object-Oriented Analysis and Design (OOAD) and Model-Driven Architecture (MDA).  
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.