However, Charles Babbage had already written his first program for the Analytical Engine in 1837..  
Normally the first step in debugging is to attempt to reproduce the problem.  
 A similar technique used for database design is Entity-Relationship Modeling (ER Modeling).  
Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability.  
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).  
 Computer programmers are those who write computer software.  
 Auxiliary tasks accompanying and related to programming include analyzing requirements, testing, debugging (investigating and fixing problems), implementation of build systems, and management of derived artifacts, such as programs' machine code.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
 Following a consistent programming style often helps readability.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
Ideally, the programming language best suited for the task at hand will be selected.  
 Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.  
It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages.  
 Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code.  
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.