There exist a lot of different approaches for each of those tasks..  
One approach popular for requirements analysis is Use Case analysis.  
 Programs were mostly entered using punched cards or paper tape.  
Trial-and-error/divide-and-conquer is needed: the programmer will try to remove some parts of the original test case and check if the problem still exists.  
Programming languages are essential for software development.  
 The first step in most formal software development processes is requirements analysis, followed by testing to determine value modeling, implementation, and failure elimination (debugging).  
Normally the first step in debugging is to attempt to reproduce the problem.  
However, because an assembly language is little more than a different notation for a machine language, two machines with different instruction sets also have different assembly languages.  
In the 9th century, the Arab mathematician Al-Kindi described a cryptographic algorithm for deciphering encrypted code, in A Manuscript on Deciphering Cryptographic Messages.  
 Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages.  
They are the building blocks for all software, from the simplest applications to the most sophisticated ones.  
 High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
However, Charles Babbage had already written his first program for the Analytical Engine in 1837.