Their jobs usually involve:  
 Although programming has been presented in the media as a somewhat mathematical subject, some research shows that good programmers have strong skills in natural human languages, and that learning to code is similar to learning a foreign language.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
Ideally, the programming language best suited for the task at hand will be selected.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
Scripting and breakpointing is also part of this process.  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
Ideally, the programming language best suited for the task at hand will be selected.  
One approach popular for requirements analysis is Use Case analysis.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
For this purpose, algorithms are classified into orders using so-called Big O notation, which expresses resource use, such as execution time or memory consumption, in terms of the size of an input.  
 The first computer program is generally dated to 1843, when mathematician Ada Lovelace published an algorithm to calculate a sequence of Bernoulli numbers, intended to be carried out by Charles Babbage's Analytical Engine.