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.  
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).  
It affects the aspects of quality above, including portability, usability and most importantly maintainability.  
Integrated development environments (IDEs) aim to integrate all such help.  
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.  
Some languages are more prone to some kinds of faults because their specification does not require compilers to perform as much checking as other languages.  
The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA.  
This can be a non-trivial task, for example as with parallel processes or some unusual software bugs.  
Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards.  
However, readability is more than just programming style.  
 Debugging is often done with IDEs. Standalone debuggers like GDB are also used, and these often provide less of a visual environment, usually using a command line.  
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.  
 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.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.