By the late 1960s, data storage devices and computer terminals became inexpensive enough that programs could be created by typing directly into the computers..  
 Different programming languages support different styles of programming (called programming paradigms).  
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).  
 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.  
For example, COBOL is still strong in corporate data centers often on large mainframe computers, Fortran in engineering applications, scripting languages in Web development, and C in embedded software.  
Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.  
Integrated development environments (IDEs) aim to integrate all such help.  
There exist a lot of different approaches for each of those tasks.  
 Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications.  
 Allen Downey, in his book How To Think Like A Computer Scientist, writes:  
 Many computer languages provide a mechanism to call functions provided by shared libraries.  
Techniques like Code refactoring can enhance readability.  
Also, specific user environment and usage history can make it difficult to reproduce the problem.  
It is usually easier to code in "high-level" languages than in "low-level" ones.  
 Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users.  
Use of a static code analysis tool can help detect some possible problems.