There exist a lot of different approaches for each of those tasks..  
Use of a static code analysis tool can help detect some possible problems.  
 Programmable devices have existed for centuries.  
Assembly languages were soon developed that let the programmer specify instruction in a text format (e.g., ADD X, TOTAL), with abbreviations for each operation code and meaningful names for specifying addresses.  
Trade-offs from this ideal involve finding enough programmers who know the language to build a team, the availability of compilers for that language, and the efficiency with which programs written in a given language execute.  
 After the bug is reproduced, the input of the program may need to be simplified to make it easier to debug.  
 Programs were mostly entered using punched cards or paper tape.  
Techniques like Code refactoring can enhance readability.  
A study found that a few simple readability transformations made code shorter and drastically reduced the time to understand it.  
 Whatever the approach to development may be, the final program must satisfy some fundamental properties.  
Unreadable code often leads to bugs, inefficiencies, and duplicated code.  
However, readability is more than just programming style.  
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.  
Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation.  
He gave the first description of cryptanalysis by frequency analysis, the earliest code-breaking algorithm.