These compiled languages allow the programmer to write programs in terms that are syntactically richer, and more capable of abstracting the code, making it easy to target varying machine instruction sets via compilation declarations and heuristics. There exist a lot of different approaches for each of those tasks. Many applications use a mix of several languages in their construction and use. However, readability is more than just programming style. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. The first compiler related tool, the A-0 System, was developed in 1952 by Grace Hopper, who also coined the term 'compiler'. One approach popular for requirements analysis is Use Case analysis. Integrated development environments (IDEs) aim to integrate all such help. The Unified Modeling Language (UML) is a notation used for both the OOAD and MDA. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. When debugging the problem in a GUI, the programmer can try to skip some user interaction from the original problem description and check if remaining actions are sufficient for bugs to appear. Programmers typically use high-level programming languages that are more easily intelligible to humans than machine code, which is directly executed by the central processing unit. Computer programmers are those who write computer software. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. There exist a lot of different approaches for each of those tasks. Computer programmers are those who write computer software. Unreadable code often leads to bugs, inefficiencies, and duplicated code. 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. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. Techniques like Code refactoring can enhance readability. Normally the first step in debugging is to attempt to reproduce the problem. Readability is important because programmers spend the majority of their time reading, trying to understand, reusing and modifying existing source code, rather than writing new source code. Programming languages are essential for software development. 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.