One approach popular for requirements analysis is Use Case analysis. 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. The academic field and the engineering practice of computer programming are both largely concerned with discovering and implementing the most efficient algorithms for a given class of problems. Computer programmers are those who write computer software. Different programming languages support different styles of programming (called programming paradigms). 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. High-level languages made the process of developing a program simpler and more understandable, and less bound to the underlying hardware. They are the building blocks for all software, from the simplest applications to the most sophisticated ones. In the 1880s, Herman Hollerith invented the concept of storing data in machine-readable form. Whatever the approach to development may be, the final program must satisfy some fundamental properties. 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. 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). Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. 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. Text editors were also developed that allowed changes and corrections to be made much more easily than with punched cards. Unreadable code often leads to bugs, inefficiencies, and duplicated code. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. There are many approaches to the Software development process. Various visual programming languages have also been developed with the intent to resolve readability concerns by adopting non-traditional approaches to code structure and display. Ideally, the programming language best suited for the task at hand will be selected. Programming languages are essential for software development. 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.