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. Computer programmers are those who write computer software. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. 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. Unreadable code often leads to bugs, inefficiencies, and duplicated code. It affects the aspects of quality above, including portability, usability and most importantly maintainability. Programmable devices have existed for centuries. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment. 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). Whatever the approach to development may be, the final program must satisfy some fundamental properties. Integrated development environments (IDEs) aim to integrate all such help. 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. One approach popular for requirements analysis is Use Case analysis. One approach popular for requirements analysis is Use Case analysis. Normally the first step in debugging is to attempt to reproduce the problem. 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. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. 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. Implementation techniques include imperative languages (object-oriented or procedural), functional languages, and logic languages. Computer programmers are those who write computer software. 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. Whatever the approach to development may be, the final program must satisfy some fundamental properties. It involves designing and implementing algorithms, step-by-step specifications of procedures, by writing code in one or more programming languages. Some text editors such as Emacs allow GDB to be invoked through them, to provide a visual environment.