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. New languages are generally designed around the syntax of a prior language with new functionality added, (for example C++ adds object-orientation to C, and Java adds memory management and bytecode to C++, but as a result, loses efficiency and the ability for low-level manipulation). 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. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. 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. Many factors, having little or nothing to do with the ability of the computer to efficiently compile and execute the code, contribute to readability. Machine code was the language of early programs, written in the instruction set of the particular machine, often in binary notation. Ideally, the programming language best suited for the task at hand will be selected. Debugging is a very important task in the software development process since having defects in a program can have significant consequences for its users. There exist a lot of different approaches for each of those tasks. Sometimes software development is known as software engineering, especially when it employs formal methods or follows an engineering design process. 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. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. Computer programming or coding is the composition of sequences of instructions, called programs, that computers can follow to perform tasks. Compilers harnessed the power of computers to make programming easier by allowing programmers to specify calculations by entering a formula using infix notation. While these are sometimes considered programming, often the term software development is used for this larger overall process – with the terms programming, implementation, and coding reserved for the writing and editing of code per se. Techniques like Code refactoring can enhance readability. Following a consistent programming style often helps readability. Also, specific user environment and usage history can make it difficult to reproduce the problem. Ideally, the programming language best suited for the task at hand will be selected. Some languages are very popular for particular kinds of applications, while some languages are regularly used to write many different kinds of applications. Programming languages are essential for software development. 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. Also, specific user environment and usage history can make it difficult to reproduce the problem.