Third Generation Languages 3GLs (High Level Languages)

A third-generation programming language (3GL) is a refinement of a second-generation programming language. The second generation of programming languages brought logical structure to software. The third generation brought refinements to make the languages more programmer-friendly. This includes features like improved support for aggregate data types, and expressing concepts in a way that favors the programmer, not the computer (e.g. no longer needing to state the length of multi-character (string) literals in FORTRAN). A third generation language improves over a second generation language by having the computer take care of non-essential details, not the programmer. Third Generation Languages feature more abstraction that previous implementations of those languages, and thus can be considered higher level languages then their first and second generation counterparts.

First introduced in the late 1950s, FORTRAN, ALGOL, and COBOL are early examples of this sort of language.

Most popular general-purpose languages today, such as C, C++, C#, Java, BASIC and Pascal, are also third-generation languages, although C++, Java and C# follow a completely different path as they are object-oriented in nature. Third generation focused on structured and had no meaning for the object encapsulation concepts.

Most 3GLs support structured programming.

A programming language such as C, FORTRAN, or Pascal enables a programmer to write programs that are more or less independent from a particular type of computer. Such languages are considered high-level because they are closer to human languages and further from machine languages. In contrast, assembly languages are considered low-level because they are very close to machine languages.

The main advantage of high-level languages over low-level languages is that they are easier to read, write, and maintain. Ultimately, programs written in a high-level language must be translated into machine language by a compiler or interpreter.

The first high-level programming languages were designed in the 1950s. Examples of early high level languages

are ALGOL, COBOL, FORTRAN, and Ada. These programs could run on different machines so they were machine-independent. As new, more abstract languages have been developed, however, the concept of high and low level languages has become rather relative. Many of the early "high level" languages are now considered relatively low level in comparison to languages such as Python, Ruby, and CommonLisp.