

# **Fourth generation language, Assembler, Compiler, and Interpreter**

**BBA | BBA-TT | BHCM**

## **Fourth generation language (4GL – 1985's to till now)**

Fourth generation language (4GL) is more non-procedural, object-oriented and conversational than prior language. It is also non-structural language. It can be employed directly by the end user or less skilled programmer to develop computer application more rapidly than the conventional programming language. Some advantages and disadvantages of this language as follows:

### **Advantages:**

- Fourth generation language (4GL) is friendly for users.
- This language is easy to develop programs.

### **Disadvantages:**

- Programs run slower because it is heavy in size and design.
- Required long time to convert into machine code.

## **Difference between third-generation language (3GL) and fourth generation language (4GL)**



---

3GL

4GL

It stands for third generation language.  
 The stepwise instructions are written.  
 Difficult to learn.  
 Difficult to debugging.  
 Typically file oriented.  
 Requires specification of how to perform the task.  
 For e.g. programming in C.

It stands for fourth generation language.  
 Requirement oriented instruction is written.  
 Easy to learn.  
 Easy to be debugging.  
 Typically, database oriented.  
 Requires specification of what task is to be performed.  
 Structured query language (SQL).

## Difference between three translators

Translators are the system software that is used to translate the source code into object code (machine code). Generally, there are three types of translators compiler, interpreter and assembler. These are all translators are interdependence and not interdependence to each other. For it, it as given below:

<b>Compiler</b>	<b>Interpreter</b>	<b>Assembler</b>
It translates the entire program first and translates it into machine code.	Also, it translates the program line by line.	Similarly, it converts the entire program into machine code.
It converts the entire program to machine code when all the syntax errors are removed, and execution takes place.	Each time the program is executed when every line is checked for syntax errors and converted to equivalent machine code.	Same as the compiler i.e. (It converts the entire program to machine code when all the syntax errors are removed, and execution takes place.)
Slow for debugging.	Faster for debugging.	Faster for debugging.
Overall execution time is less.	It is overall execution time is more.	Overall execution time is more than the interpreter.
It creates only one object program after executing of the program.	It does not create an object program.	Same as compiler.
It translates high-level language into machine code.	Same as the compiler.	It translates assembly level language into machine language.

For more notes visit [tyonote](http://tyonote.com)