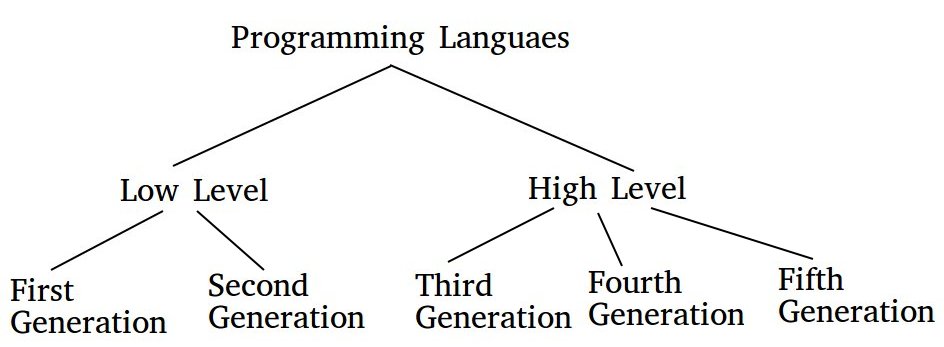
**Generations of Programming Languages**

There are five generations of Programming languages which are categorized as following:



**First-Generation Languages:**   
These are low-level languages like machine language.

**Second-Generation Languages:**   
These are low-level assembly languages used in kernels and hardware drives.   
**Third-Generation Languages:**   
These are high-level languages like C, C++, Java, Visual Basic, and JavaScript.   
**Fourth Generation Languages:**   
These are languages that consist of statements that are similar to statements in the human language. These are used mainly in database programming and scripting. Examples of these languages include Perl, Python, Ruby, SQL, and MatLab (MatrixLaboratory).   
**Fifth Generation Languages:**   
These are the programming languages that have visual tools to develop a program. Examples of fifth-generation languages include Mercury, OPS5, and Prolog.

The first two generations are called low-level languages. The next three generations are called high-level languages.

**1. First-Generation Language:**

The first-generation languages are also called machine languages/ 1G language. This language is machine-dependent. The machine language statements are written in binary code (0/1 form) because the computer can understand only binary language.

**Advantages:**

1. Fast & efficient as statements are directly written in binary language.

2. No translator is required.

**Disadvantages:**

1.  Difficult to learn binary codes.

2. Difficult to understand – both programs & where the error occurred.

**2. Second Generation Language:**

The second-generation languages are also called assembler languages/ 2G languages. Assembly language contains human-readable notations that can be further converted to machine language using an assembler.

**Assembler –** converts assembly level instructions to machine-level instructions.

Programmers can write the code using symbolic instruction codes that are meaningful abbreviations of mnemonics. It is also known as low-level language.

**Advantages:**

1. It is easier to understand if compared to machine language.

2. Modifications are easy.

3. Correction & location of errors are easy.

**Disadvantages:**

1. Assembler is required.

2. This language is architecture /machine-dependent, with a different instruction set for different machines.

**3. Third-Generation Language:**

The third generation is also called procedural language /3 GL. It consists of the use of a series of English-like words that humans can understand easily, to write instructions. It’s also called High-Level Programming Language. For execution, a program in this language needs to be translated into machine language using a Compiler/ Interpreter. Examples of this type of language are C, PASCAL, FORTRAN, COBOL, etc.

**Advantages:**

1. Use of English-like words makes it a human-understandable language.

2. Lesser number of lines of code as compared to the above 2 languages.

3. Same code can be copied to another machine & executed on that machine by using compiler-specific to that machine.

**Disadvantages:**

1. Compiler/ interpreter is needed.

2. Different compilers are needed for different machines.

**4. Fourth Generation Language:**

The fourth-generation language is also called a non – procedural language/ 4GL. It enables users to access the database. Examples: SQL, Foxpro, Focus, etc.

These languages are also human-friendly to understand.

**Advantages:**

1. Easy to understand & learn.

2. Less time is required for application creation.

3. It is less prone to errors.

**Disadvantages:**

1. Memory consumption is high.

2. Has poor control over Hardware.

3. Less flexible.

**5. Fifth Generation Language:**

The fifth-generation languages are also called 5GL. It is based on the concept of artificial intelligence. It uses the concept that rather than solving a problem algorithmically, an application can be built to solve it based on some constraints, i.e., we make computers learn to solve any problem. Parallel Processing & superconductors are used for this type of language to make real artificial intelligence.

Examples: PROLOG, LISP, etc.

**Advantages:**

1. Machines can make decisions.

2. Programmer effort reduces to solve a problem.

3. Easier than 3GL or 4GL to learn and use.

**Disadvantages:**

1. Complex and long code.

2. More resources are required & they are expensive too.