1. Machine Language

Definition:
Machine language is the lowest-level programming language which is also known as binary. It uses only binary numbers (0s and 1s) and it directly understand by computer.

Advantages

1. High Speed:

Since no conversion is needed, the applications developed using machine language are extremely fast. It is usually used for complex applications such as space control system, Chemical processing, etc.

2.Translation is not needed

It is only language that the computer Indirectly execute without the need for conversion, and it is only language that the computer is able to understand. Every high-level language program must be converted into machine readable form so that the computer can understand the instructions.

3.Gives full control of hardware.

As machine language is written in binary, it gives programmers direct rights to manipulate memory, registers, and system resources without difficulty.

Disadvantages

1.Machine Dependent

Every computer differs from the other based on its architecture, processing power, memory, etc hence application developed for a particular type of computer may not run on the other.so it may be one problem after developing applications.

2.Complex Language

Machine Language is very difficult to read and write. All the data an instruction must be converted to binary code. So, it is almost impossible to remember all instructions and programmer must master in each operation and specific location for each piece of data, so to code on machine language programmer must also hardware knowledge.

3.Poor readability:

As all code is written in binary (0s and 1s), it is very difficult to understand or debug log strings of 0s and 1s, as it may take a day, week or even month to fix large code.

2. Assembly Language

Definition:
Assembly language is a symbolic version of machine code. It uses short codes (mnemonics) like MOV,ADD ,SUB instead of 0s and 1s.

Advantages

1.Easy to Use and Understand:

Assembly language use mnemonics codes instead of binary used in machine language. Hence, the program written in the assembly language are much easier to understand and is more user friendly compared to machine language.

2.More Control on Hardware:

Assembly language also gives direct access to key machine features essential for using certain kinds of low level routines such as operating system, kernel,. device driver and machine control.

3.Efficiency

Assembly programs can run much faster, use less memory, and consume fewer resources than high-level languages because they are closely tied to machine instructions and avoid the overhead of compilers or interpreters.

Disadvantages

1.Machine Dependent

Each computer has their won machine and assembly language which means that program return in this language are not portable to other. This makes it a low-level language.

2.Harder to learn

Assembly requires working very close to the hardware. Even simple tasks take many steps and can be confusing.

3. No Standardization

Assembly languages cannot be standardized because each type of computer has different instructions

3. High-Level Language (HLL)

Definition:
A high-level language is a programming language that is designed to understand easily by human. Examples of high-level programming language : Python, Java, and C#.

Advantages

1.Easy to learn and use

High-level languages is easy to learn because they use words and symbol that is regularly used by humans. You don't need to know how memory work to write programs like in low-level language. This makes them beginner-friendly and faster to pick up.

2.Portable across systems

Programs written in high-level languages can run on different computers without big changes. You don't have to rewrite code for each type of machine/device. This saves time and effort when deploying software.

3.Rich libraries and features

High-level languages come with built-in tools and libraries that handle complex tasks. Instead of making everything, you can use ready-made functions for things like networking or graphics. This speeds up development and reduces errors.

Disadvantages

1.Less control over hardware

High-level languages hide the details of how the computer works. This means you can't fine-tune performance or directly manage memory as easily. For tasks needing maximum speed, this can be limiting.

2.Slower execution compared to low-level code

High level program must be converted to machine level program due to this it takes more time and execution is slow compare to low-level code.

3.More resource usage

Programs in high-level languages usually consume more memory and processing power. They rely on interpreters or compilers that add extra load. On small devices or embedded systems, this can be a drawback.

4. Assembler

Definition: Assembler is translator that converts the assembly language into machine language.

Advantages:

1. Makes assembly code understandable to the computer.

2. Very fast in converting instructions.

3.. Allows direct control over hardware.

Disadvantages:

1. It only work for assembly language.

2. Assembly programs are hard to maintain.

3. Not portable across different systems.

5. Compiler

Definition: A compiler is a translator the translates the entire high-level program into machine code at once.

Advantages:

1. Once compiled, programs run very fast.

2. Shows all errors together after compiling whole program.

2. Produces optimized machine code for better performance.

Disadvantages:

1. Program won't run until all errors are fixed.

2. Slower Execution as Compilation can take time.

3. Not ideal for quick testing of small code parts.

6. Interpreter

Definition: A compiler is a translator the translates the entire high-level program into machine code line by line.

Advantages:

1. Easy to debug since errors show up immediately.

2. Beginner-friendly and simple to use.

3. Runs the program instantly without waiting for compilation.

Disadvantages:

1. Slower than compiled programs.

2. Must interpret the program every time it runs.

3. Not suitable for large or complex programs.

🔗 **7. Linker**

Definition: Linker combines multiple object files and libraries into one executable.

Advantages:

1. It helps you merge code from different files/modules.

2. Enables reusability of libraries.

3. Supports modular programming for large projects.

Disadvantages:

1. Linking errors can be tricky to fix.

2. Adds extra time to the build process.

3. Requires careful management of dependencies.

8. Loader

Definition: Loads the executable program into main memory (RAM) for execution.

Advantages:

1. Prepares the program to run smoothly.

2. Efficiently manages memory loading.

3. Essential for program execution.

Disadvantages:

1. Errors during loading can crash the program.
2. Adds overhead before execution begins.
3. Limited flexibility if memory is constrained.