High-level: language that can be comprehended by humans

Low level: Language that can be read by a computer but is difficult to be comprehended by humans.

Translators: Translate high-level languages to low-level languages and vice versa.

Compilers: Translates high-level languages to low-level languages so that computers can proceed forward with the instructions.

Interpreter: Reads a high-level language instruction and carries out that function and continues doing so.

Errors:

1. **Syntax error:** When a high-level language does not follow the rules of the programming language
2. **Logic error:** When the programme does not do what the programmer wanted it to do.

Compilers

A compiler takes the source code as a whole and translates it into machine code all in one go. Once converted, the object code can be run unassisted at any time. This process is called compilation.

Advantages of compilers:

- Compiled programs run quickly, since they have already been translated.
- A compiled program can be supplied as an executable file, which is a file that is ready to run, Since an executable file cannot easily be modified, programmers prefer to supply executables rather than source code.
- Compilers optimise code, Optimised code can run quicker and take up less memory space.

Disadvantages

- The source must be recompiled every time the programmer changes the the program
- The source code compiled on one platform will not run on another - the machine code is specific to the processor's architecture.

Interpreters

An interpreter translates source code into machine code one instruction at a time. It is similar to a human translator translating what a person says into another language, sentence by sentence, as they speak. The resulting machine code is then executed immediately. The process is called interpretation.

Advantages

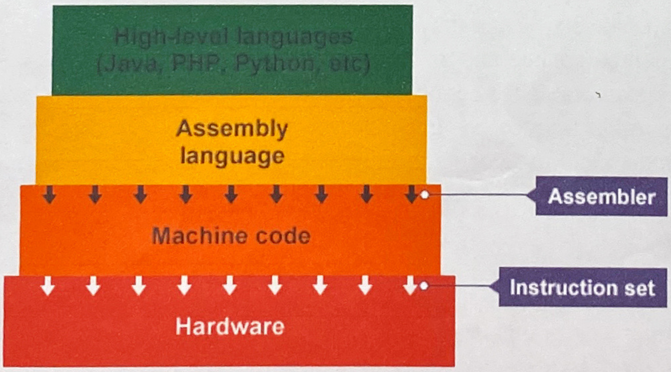
- Instructions are executed as soon as they are translated.
- Errors can be quickly spotted - once an error is found, the program stops running and the user is notified at which part of the program the interpretation has failed. This makes interpreters extremely useful when developing programs

Disadvantages

- Interpreted programs run slowly as the processor has to wait for each instruction to be translated before it can be executed.
- Additionally, the program has to be translated every time it is run.
- Interpreters do not produce an executable file that can be distributed. As a result, the source code program has to be supplied, and this could be modified without permission.
- Interpreters do not optimise code - the translated code is executed as it is.

Assemblers

Assemblers are a third party type of translator. The purpose of an assembler is to translate assembly language into machine code.
Whereas compilers and interpreters generate many machine code instructions for each high-level instruction, assemblers create one machine code instruction for each assembly instruction.


High-level languages
(Java, PHP, Python, etc)
Assembly language
Assembler
Machine code
Instruction set
Hardware