

Programming

Compiler:

a computer program that translates a program written in a high-level language (the **SOURCE**) into another lower-level language, like machine code (the **OBJECT**).

- ☞ *a compiler translates the whole source at once*
- ☞ *a compiler checks also for mistakes in the source (made by the programmer)*

Interpreter:

translates high level language to machine code

- ☞ *an interpreter translates the source line-by-line*
- ☞ *the instructions will run immediately*

Application:

a task that can be done with the help of a computer. The computer runs a program which is designed to solve all problems of this particular task.

- ☞ *Business applications (payroll, reservations, stock control etc.)*
- ☞ *Scientific applications (statistics, car-design, weather forecast etc.)*
- ☞ *Control applications (traffic control, air conditioning, robots etc.)*

High Level Language

a computer language which is close to (mostly) English. They are PROBLEM-ORIENTATED, this means some high-level languages are dedicated to specific problems. By using a high-level language, the programmer can make very good solutions for specific kind of problems

<i>Cobol</i>	-	<i>Business orientated</i>
<i>Pascal</i>	-	<i>Education orientated</i>
<i>HyperTalk</i>	-	<i>Education orientated</i>
<i>C and C++</i>	-	<i>General (up-to-date)</i>
<i>Java and (-script)-</i>		<i>Internet orientated</i>

Low Level Language

a computer language which is close to the computers language. It is MACHINE-ORIENTATED, this means most low-level languages are dedicated to specific computers. By using low-level languages, the programmer can make optimal use of the computers hardware (= make a fast (!) program).

<i>Assembler</i>	-	<i>computer orientated</i>
------------------	---	----------------------------

Machine Code

a coding system specific to the hardware of a given computer model, into which any high-level or assembly program must be translated (compiled) before it can run on the computer.

<i>Machine Code</i>	-	<i>CPU orientated</i>
---------------------	---	-----------------------

How To Make a Program

- Choose the programming language which is the best suitable to your problem
- Think about the **Algorithm**, make a program flowchart / describe with Pseudo code
- Try your program using a **dry-run**
- Program all necessary statements, instructions, scripts, buttons etc.
- **Compile** your program (check the mistakes that the compiler will tell you)
- **Run and Test** your program (see if it works as you want . .)

Pascal / Cobol / C++ / Fortran / HyperTalk

**HIGH LEVEL
LANGUAGE**

Assembler for Intel / Motorola etc.

**LOW LEVEL
LANGUAGE**

COMPIRATION

The compiler is specific for every different computer (see below)

**MACHINE
CODE**

Can run (only) on the specific computer (MS-DOS / Windows / Macintosh / Sun / Unix / Linux / Mainframe etc.)