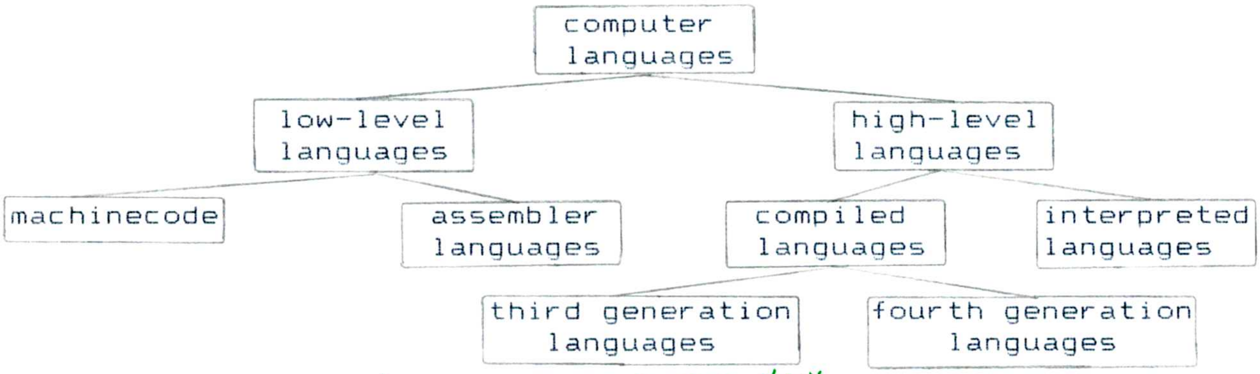
Unit 2: Class relationships (1) Classification
Consolidation: C
Title: A tree diagram classifying computer-languages.
Author: Erik van Eykelen, E11C
Date: nov-7-1988



The classification ~~of~~ in the above scheme diagram ~~is~~ shows the very clear difference between high-level and low-level languages. Everyone who has seen (or has been tortured with) a low-level language knows that the name 'low-level' is very aptly chosen. The program instructions have to be written in zeros and ones!

The low-level branch of computer-languages is again divided in two divisions. The machinecode division is the kind of language a computer (without translation) understands. Assembly language has been invented to soften the pain of programming in bits but still demands from the programmer to chop his program into very tiny bits (in this case not computer bits !) called mnemonics (English-like abbreviations of processor-instructions).

The high-level branch has a more diverse character. The first subdivision affects the compiled languages and the interpreted languages. Very briefly: the former is translated completely before being executed and the latter is translated during run-time.

The compiled languages are divided in two groups, the third and the fourth generation languages. The hardware is currently entering the fifth generation and a well-known fact is that the software always is a few paces behind.

OK .

branch
group
division
subdivision

a bit confusing!