# Language Processor

A language processor is a hardware device designed or used to perform tasks, such as processing program code to machine code. Language processors are found in languages such as FORTRAN and COBOL.

**Types of language processing:**

**Assemblers:** Language processors that map low-level language instructions into machine code, e.g. the ARM assembler.

**Compilers:** Language processors that map high-level language instructions into machine code, e.g. Delphi, GCC, Visual C++ etc.

**Pre-processors:** Language processors that map a superset of a high-level language into the original high-level language, or perform simple text substitutions before translation takes place.

**Interpreters:** Language processors that include an execution component, i.e. they perform the operations specified in the source text, rather than re-expressing them in another language; e.g. Matlab.

**Disassemblers:** Language processors that attempt to take object code at a low level and regenerate source code at a higher level.

**Language Processor Examples:**

- GCC, Delphi, Visual Studio
- TeX/LaTeX
- Postscript
- HTML & Web browsers
- XML

How do we judge the quality of a language processor?

- (Most important property): Correctness of the generated code
- Conformity to the language specification; avoids temptations to implement a subset/superset of the language, which might result in a reduction of portability.

- Quality of the generated code with respect to size and speed.
- Speed of the language processor itself.
- User-friendliness, as evident in its quality of error reporting.