## 3.1.1 Computer Languages

- **Syntax** = grammar, each statement must be constructed correctly
- **Semantics** = meaning, each statement must implement what the programmer intends
- **Low level** languages e.g. **machine code** (0's and 1's) are easy for the computer to implement but hard for human programmers and so prone to errors.
- Assembly language is easier but requires some translation.
- **High level** languages e.g. BASIC, FORTRAN, Java portable(run on different machines), comprehensible (most are English based) and differ depending on their purpose.
- HL languages need a **translator** program to convert the programmer's **source code** into the computer's **object code** (usually machine code).
- The translator can be an **interpreter** (translates and executes line by line, does not save object code, stops at the first error, needs less memory, must be in RAM while running) or a **compiler** (produces a complete object code program which can be stored, gives an error listing after compilation, can be run on any computer as it no longer needs the compiler).
- Translators will indicate syntax errors but cannot detect semantics errors.

## 3.1.2 Software Development Tools

- Tools that aid the developer (not regarded as languages in their own right):visual IDE's, HTML, scripting languages, DBMS's, CASE, macros.
- **Visual Integrated Development Environments** are programs that allow software to be written, compiled, run and debugged in a user friendly way, usually have a text editor which highlights syntax errors, compiler, library of common routines, debugging help, GUI, e.g. BlueJ, NetBeans.
- **Hypertext Markup Language** uses tags e.g. <H4> to control the display behaviour of **browsers** e.g. *Firefox*, *IE*, not truly programming languages as authors do not have full control over the way their product is displayed, **HTML editors** e.g. *Dreamweaver* allow easier **WYSIWYG** (what you see is what you get) editing.
- **Scripting languages** allow the insertion of short routines into other programs e.g. *VBScript* for *Microsoft Office*, *JavaScript* (not the same as Java!) supports HTML and makes web pages more dynamic on the client side, *PHP* etc. make web pages more dynamic on the server side.
- **Database Management Systems** provide interfaces which make database access simpler, they hide the internal structure of the database, manage the data dictionary (database structure, relationships between data items, etc.), manage data integrity (schedule backups, etc.) and control data security (access rights and sharing violations).

- **Computer Aided Software Engineering** provides techniques and software that help in the design cycle: data flow diagrams, costing, scheduling tasks (e.g. Gantt charts), documentation preparation, version control.
- **Macros** allow common sets of tasks in general purpose applications e.g. *Office* to be recorded and re-used later (they automatically write the code required in an appropriate HL language e.g. *Visual Basic* for *Word* or *Excel*).