

Topic 2.6 | GCSE Computer Science | Language's and IDE's

| | Language | Syntax | Translation | Hardware dependent? | Example code |
|------------|---------------------------------|--|--|---|---|
| Low level | Machine Code | Data and instructions made up of 1's and 0's | Does need to be translated | YES (unique to each processor type) | 11000101 11100101 11001101 11010101 01010111 11001000 |
| | Assembly Language | Mnemonics/ Symbols | One statement translates to one machine code instruction | YES (unique to each processor type) | MDV1 #58 #6A LDA #6A |
| High Level | Python, JAVA, C++, Visual Basic | Resembles human language | One statement translates into many machine code language | NO - transferrable and usable on any computer | print ("Hello World") |

Python IDLE contains a variety of features that support the development of code including

- **Syntax Highlighting** – coloured illustration of coded elements
- **Auto indentation** – keeping subroutines in proper locations
- **Bracket Matching** – Indicating matching sets of delimiters
- **Auto complete** – finding key words from dictionaries to aid with code entry
- **Syntax error checking** – illustrating the lines within the code that contain errors

Common IDE Tools

Editor to enable program code to be entered/edited

Error diagnostics / debugging to display information about errors (syntax / run time) / location of errors and suggest solutions

Run-time environment to enable to the program to be run and check for run time errors / test the program

Translator / compiler / interpreter to convert the high level code into machine code / low level code / binary AND to enable to code to be executed / run

Breakpoint to stop/pause program execution at a specific point

Watch window to check contents of variables

Syntax completion suggests/corrects code

Keyword highlighting / colour coding keywords / pretty printing colours
command words / variables

| | |
|---|---|
| High Level programming | Also known as high level language. This is a computer programming language used to write programs. High-level languages need to be translated into machine code (binary) through a compiler, interpreter or assembler |
| Low Level programming | Also known as low level language. This is a computer programming language which closely represents machine language (binary). Low level languages are more difficult to understand than high-level languages, but they execute quicker. |
| Translator | Translators are needed to translate programs written in high level languages into machine code that a computer understands. Tools exist to help programmers develop error-free code. |
| Assembler | A program that translates assembly languages into machine code. |
| Compiler | A program that translates high-level programming language into machine code. It does the whole code in one go before running it. |
| Interpreter | A program that translates high-level programming languages into machine code. It does this one line at a time. |
| IDE - Integrated Development Environment | A piece of software with tools that helps a programmer to write error-free, maintainable code. |
| Editor | The part of an IDE that allows the user to write and amend code. |
| Error Diagnostics/ Debug | The process of finding and correcting programming errors |
| Run Time Environment | The part of an IDE where a program runs. |