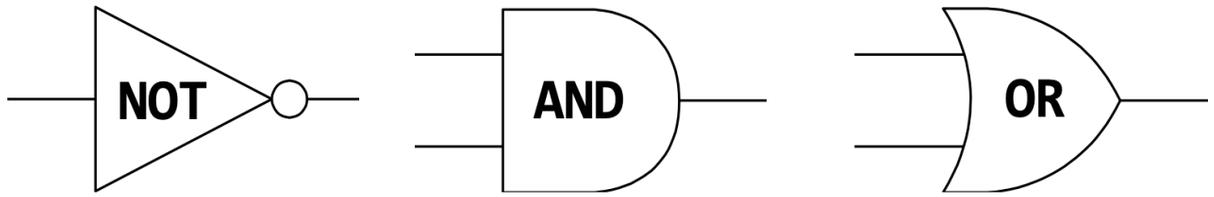
Knowledge Organiser 2.4 and 2.5 : Boolean logic, Programming Languages and IDEs

1. Logic Gate Symbols



~A | A ^ B A & B | A v B A | B

2. Truth Tables

A	NOT A
0	1
1	0

A	B	A AND B
0	0	0
0	1	0
1	0	0
1	1	1

A	B	A OR B
0	0	0
0	1	1
1	0	1
1	1	1

4. Translators

Assembler	Assembles' assembly language into machine code. Translates the whole code before execution
Compiler	Translates source code from high-level languages into object code and then into machine code ready to be processed by the CPU. The whole program is translated into machine code before it is run.
Compiler Advantages	• No need for translation software at run-time, and no need to share original source code • Speed of execution is faster because code is usually optimised.
Complier Disadvantages	• You cannot compile the program if there are syntax errors anywhere in it which can make it tricky to debug. • If you change anything you need to recompile the code
Interpreter	Translates source code from high level languages into machine code ready to be processed by the CPU. The program is translated line by line as the program

3. Levels of Programming Languages

Machine Code 1st Generation	• Binary representation of instructions in a format that the CPU can decode and execute. • Have an operation code (opcode) instruction and address or data to use (operand).
Low-Level Languages 2nd Generation	• Written in Assembly language. • Translated by an assembler into machine code. • Used for embedded systems and device drivers where instructing the hardware directly is necessary. • One instruction translated into one machine code instruction. • The code works on one type of processor only. • The programmer works with memory directly. • Code is harder to write and understand. • Memory efficient. • Code is fast to execute.

5. Integrated Development Environments

Debugging Tools	• Breakpoints – stopping at a line of code during execution. • Stepping through lines of code one at a time. • Tracing through a program to output the values of variables.
Run Time Environment	• Output window. • Simulating different devices the program can run on.
Usability Functions	• Navigation, showing/hiding sections of code. • Formatting source code often in different colours.