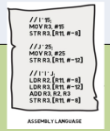
GCSE Computer Science
Topic 2.4 Translators and facilities of languages

Low Level		High Level
Machine Language	Assembly Language	Python, C, C++, Java, SQL, HTML etc.
Binary Programs are written as millions of 1s and 0s.	Each command word represents one binary instruction in machine code. ADD e.g. is used to replace the binary command 1011 0000	Resemble human language. Keywords used e.g. print, if, input. Deal with logic not how the CPU / Memory works.



ASSEMBLY LANGUAGE	HIGH Level
Used by embedded systems as it is used to control system hardware. Used in real-time systems where speed is essential. Specific code per CPU – Programs written in for one type and cannot be used on others.	Most software is developed. Programs are portable from one machine to another. Can be used on different models of CPU.

LOW level	HIGH Level
1 instruction in assembly = 1 in machine code.	1 instruction in high level = many instructions of machine code.
Written for 1 type of machine.	Same code will work on many different machines/processors.
The programmer needs to know about the specifics of the CPU.	The programmer doesn't need to know about the CPU.
Code is difficult to read, understand and modify.	Code is easy to read, understand and modify.
Commands in machine code can be executed directly without the need for a translator.	Must be translated into machine code before a computer can understand it.
Machine code controls exactly what the CPU does/ how it uses memory so programs will be memory efficient and faster.	You don't have control over the CPU does, so programs will be less memory efficient and slower

	Translators of High Level Code	
Assembler	Compiler	Interpreter
Assemblers are used to turn assembly language into machine code. They just have to assemble the mnemonics then turn them into machine code instructions. *Remember – 1 assembly instruction per machine language command.*	Compiler translates high level code in one go. It compiles the program first then executes it so it can be processed quicker. It creates an executable file of 'compiled code' which protects the source code from being viewed by others. Errors reported at the end.	An interpreter translates line by line and is required each time the program is run. When an error is encountered, the translation process is halted and the error is reported to the programmer. Easier to debug but slower as needs to be translated each time it is run. Easy to edit as source code is always available.

IDEs: A piece of software that provides a **combination** of tools to help the programmer develop their program.

The 5 main features of IDEs

Editors: This is where the code is written.
Line numbering, colour coding, auto-indentation.
Some IDEs have auto-correct and auto-complete

Run-time environment
Allows the code to be RUN within the IDE.

Error diagnostics / debugging tools help the programmer identify syntax errors.

They provide the location and type of error encountered.

Translators: **Compiler** / **interpreter** or both.

Which translates the program code into machine code within the IDE.

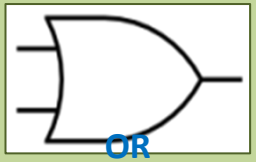
GCSE Computer Science
Topic 2.5 Computational Logic

Computers are DIGITAL .
Digital signals can only be ON or OFF.
Computers use binary to represent this:
1 = ON, 0 = OFF

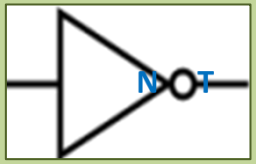
Logic gates are special switches built into computer chips that use transistors

They receive binary data 1s and 0s.
Apply a Boolean operation: AND, OR, NOT.
Then output a binary result: either 1 or 0

1 AND 1 = 1
Any other inputs = 0
∧



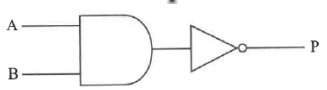
1 OR more 1s
INPUTTED = 1 OUTPUT.
∨



If 1 is INPUT,
it is **NOT** 1 on OUTPUT.
¬

Multiple logic gates can be added to the same circuit to carry out different operations.
You can work out the truth tables by working through each gate, in order.

State the input values when output P is 0.



A = 1 B = 1 [1]

GCSE Computer Science
Topic 2.4 Computational Logic & 2.5 Translators and facilities of languages

What I need to know:

State the two levels of programming language.
Describe the key features between machine, assembly and high level language.
Describe the uses of assembly language and high level language.
Describe the differences between high level and low level languages.
Outline the function of an assembler.
Outline the function of a compiler.
Outline the function of an interpreter.
State what IDE stands for.
Explain what an IDE is used for.
Describe the 5 main features of an IDE.
Explain why computers use binary.
Describe what a logic gate is.
Draw and label the 3 main logic gate symbols.
Draw a truth table to show the inputs and outputs for each logic gate.
Draw a logic diagram with multiple gates and explain how to work out the input/output combinations.

Natasha needs to translate her program into machine code. Outline **two** differences in the way a compiler and interpreter would translate her program.

A logic gate can be written as P = A AND B.

a) State the value of input B when input A is 1 and output P is 0.

B = ...

Two truth tables are given below. A and B are inputs. P and Q are outputs.

Draw the correct logic gates for each of these truth tables.

a)

A	P
0	1
1	0

[1]

b)

A	B	Q
0	0	0
0	1	1
1	0	1
1	1	1

[1]

A series of transistors make the two-level logic circuit (NOT A) AND (B AND C).

a) Complete the truth table below.

A	B	C	NOT A	B AND C	(NOT A) AND (B AND C)
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

[3]