

AQA Computer Science A-Level
**4.6.2 Classification of programming
languages**
Intermediate Notes



Specification:

4.6.2.1 Classification of programming languages:

Show awareness of the development of types of programming languages and their classification into low-and high-level languages.

Know that low-level languages are considered to be:

- machine-code
- assembly language

Know that high-level languages include imperative high-level language.

Describe machine-code language and assembly language.

Understand the advantages and disadvantages of machine-code and assembly language programming compared with high-level language programming.

Explain the term ‘imperative high-level language’ and its relationship to low-level languages.



The development of types of programming languages

Early computers were programmed using low-level languages. These languages **directly manipulate** the processor, require a **great deal of effort** on the part of the programmer and are **prone to errors**.

High-level languages were developed to allow for **instructions** to be communicated to a computer's processor, making the job of programming far easier.

Low-level languages

Programs written in low-level languages are **specific** to the type of processor they are written for and **directly affect** the computer's processor.

There are **two categories** of low-level language: **machine code** and **assembly language**.

Machine code

Machine code uses only the **digits 1 and 0** to represent instructions. This makes programs written in machine code **very long** and **extremely difficult** for humans to understand.

Because of this, machine code programs are **prone to errors** and **difficult to debug**.

```
01010101
11010110
01001011
10110110
```

Because machine code **directly manipulates** a computer's processor, it is a **very powerful** paradigm.



Assembly language

Assembly language was developed with the intention of **simplifying the process** of writing computer programs. **Mnemonics**, such as ADD and MOV, are used in place of the binary instructions that machine code uses. This makes assembly language **more compact** and **less error prone** than machine code.

```
STR R4, #45
ADD R1, R2, 3
MOV R2, R1
HALT
```

Note

This example uses AQA's own assembly language instruction set.

Each assembly language instruction has a **1-to-1 correlation** to a machine code instruction.

AQA have made **their own assembly language** for use in exams. You need to make sure you're familiar with using it to **understand** programs and even to **write your own**.

Synoptic Link

AQA's assembly language instruction set is covered in **structure and role of the processor and its components** under **computer organisation and architecture**.

High-level languages

High-level languages are the type of programming language that you're most likely used to using. Examples of high-level languages include C#, Java, Pascal, Python and VB.Net.

Unlike low-level languages, high-level languages are **not platform specific**. However, high-level languages must be **translated** into machine code before they can be executed.

Synoptic Link

Different types of **translators** are covered in the notes for **types of program translator**.

High-level languages don't use binary digits or mnemonics but use **English instructions** like "While" and **mathematical symbols** like "+". This makes high-level languages **much easier** for humans to learn and understand as well as making them **easier to debug**.

```
While x < y
    x = x + y
End While
```



High-level languages vs. low-level languages

	Low-level		High-level
	Machine code	Assembly language	
Portability	Not portable. Programs are processor specific .		Portable. Programs are not specific to certain processors.
Ease of use	Code is difficult for humans to understand.	Mnemonics help to make code slightly easier for humans to understand.	Code uses English, making it easy for humans to understand.
Ease of debugging	Errors are very difficult to spot and correct.	Debugging is easier than with machine code.	Debugging is fairly easy .
Ease of execution	Machine code is directly executed by processors.	An assembler must be used before a program is executed..	A compiler or interpreter must be used to translate code before it can be executed.

High-level languages include **imperative high-level languages**. In a similar way to low-level languages, imperative high-level languages are formed from instructions that specify **how** the computer should complete a task, in contrast to declarative programming which describes **what** a computer should do.

