Computer Languages

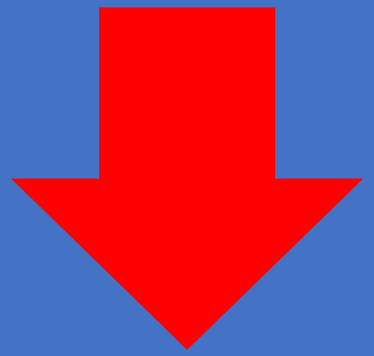
2.1

Two Categories of Computer Languages

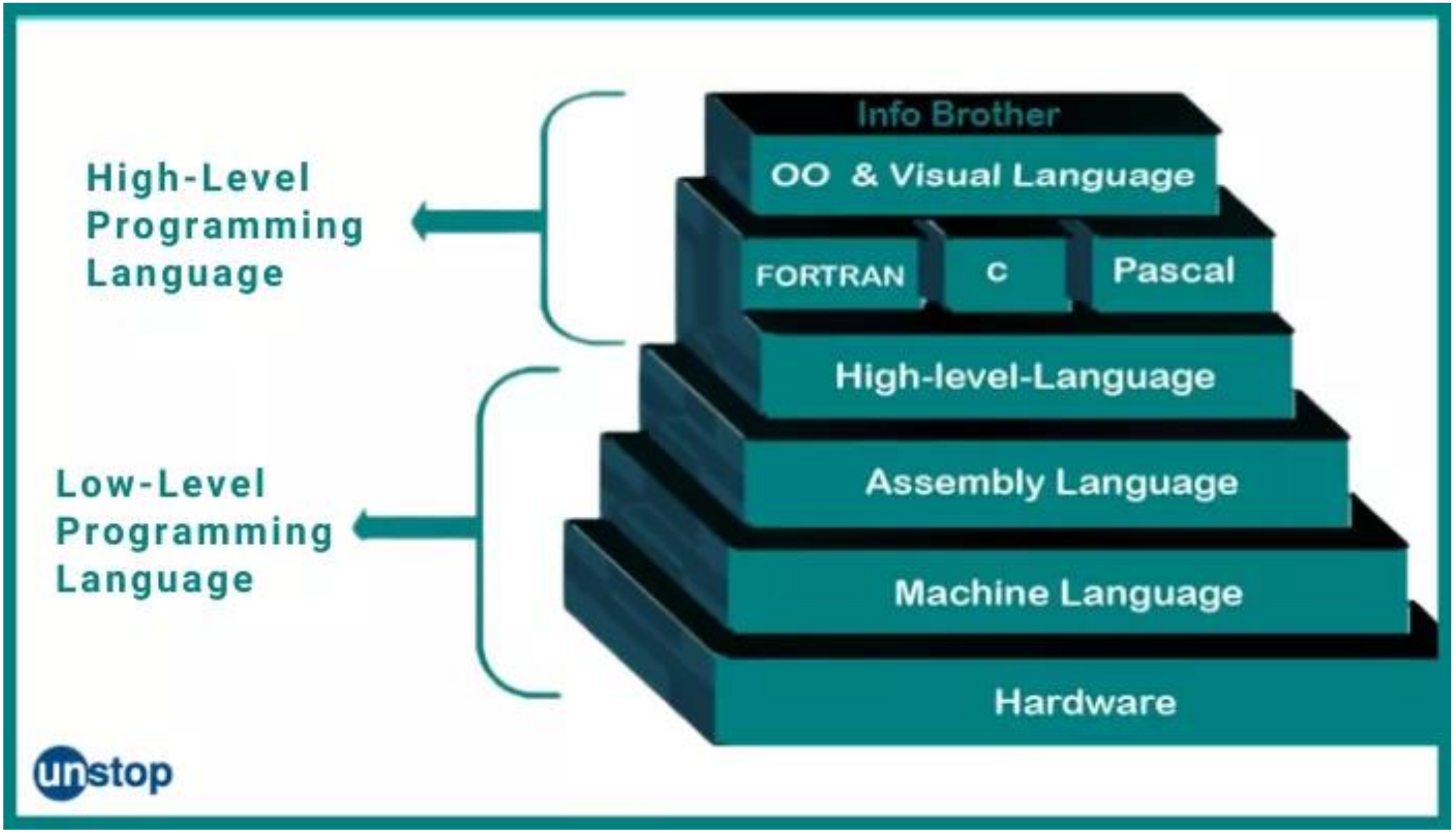
Guide Question

Between Low-level and high-level languages, what is the most essential and useful? Why?

High Level Languages



Low Level Languages

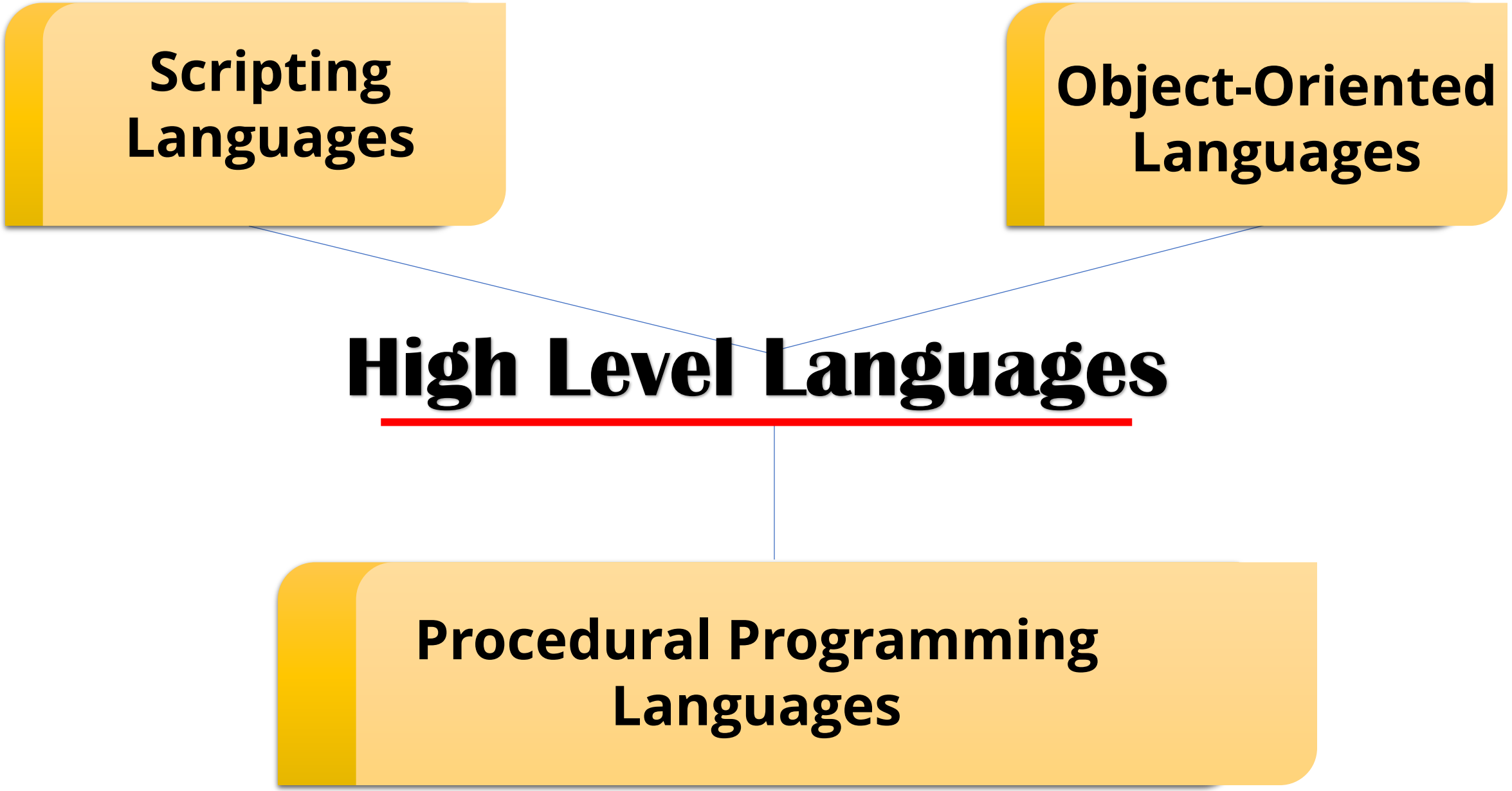

High-Level
Programming
Language
Low-Level
Programming
Language
Info Brother
OO & Visual Language
FORTRAN
C
Pascal
High-level-Language
Assembly Language
Machine Language
Hardware
unstop

High Level Languages

these are programming languages.

offers the programmer development tools such as libraries and built-in functions.

prominent Examples are PASCAL, FORTRAN, C++ etc.

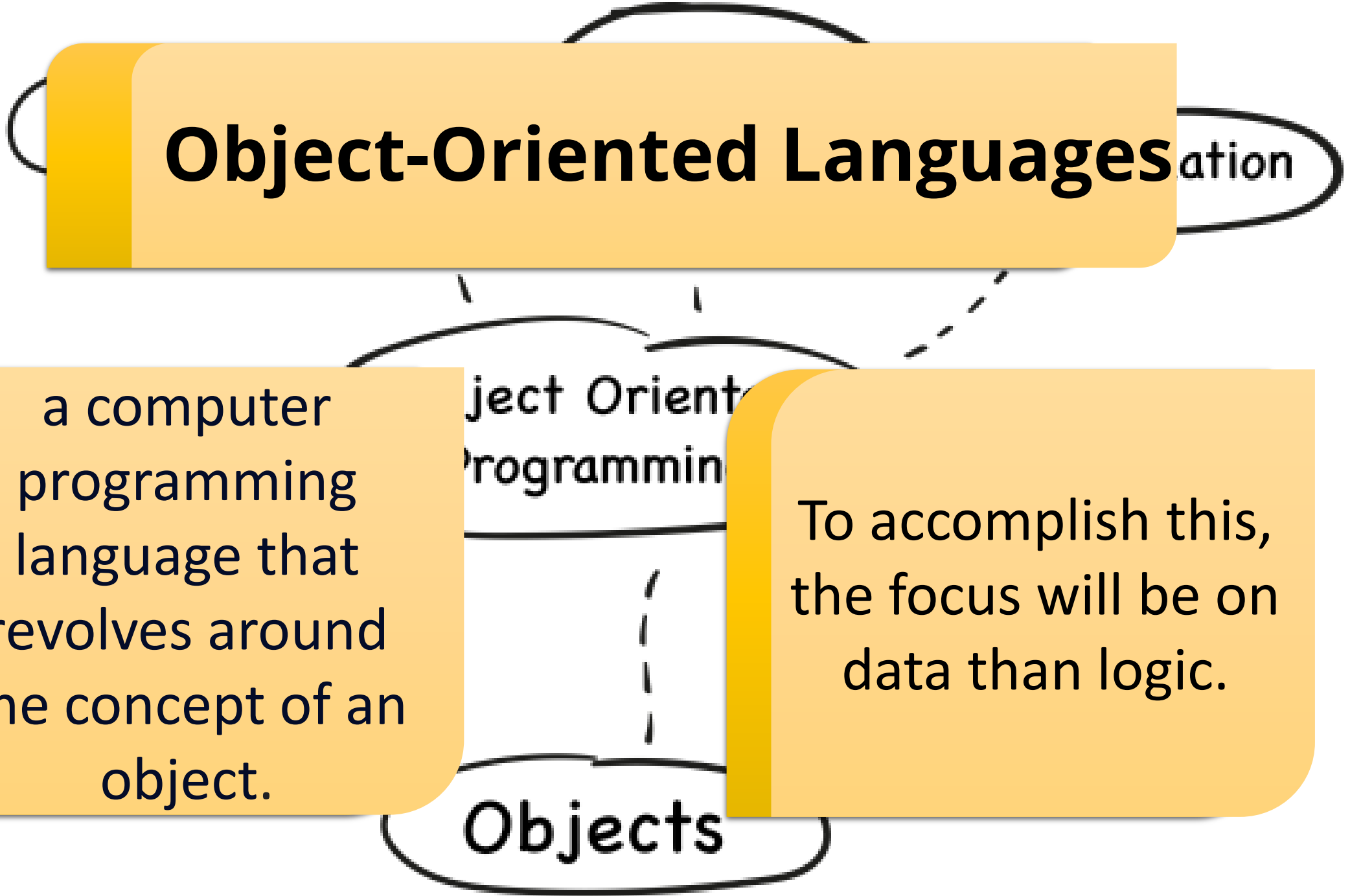

Scripting Languages
Object-Oriented Languages
High Level Languages
Procedural Programming Languages

Scripting Languages

scripts are essentially programming languages.

employ a high-level construct that allows it to interpret and execute one command at a time.

Object-Oriented Languages



a computer programming language that revolves around the concept of an object.

To accomplish this, the focus will be on data than logic.

well structured steps and complex procedures within its programming to compose a complete program.

- ✓ derived from structural programming.
- ✓ consist several computational steps that are carried out during the execution of a program.

Procedural Programming Languages

Basic Characteristics of High-Level Computer Language

Characteristic 1

- Syntax that is easy for humans to understand

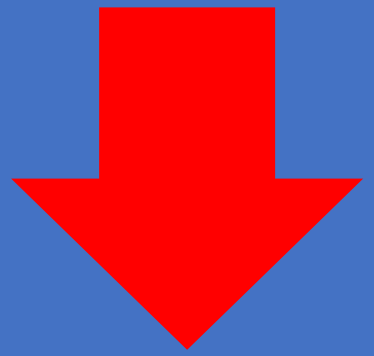
Characteristic 2

- Syntax that uses command words similar to natural human language

Characteristic 3

- A single line of code can accomplish multiple task

High Level Languages



Low Level Languages

Low Level Languages

are very easily understandable by the machine.

help in operating, syncing and managing all the hardware and system components of the computer.

Main function of low-level languages is to interact with the hardware of the computer.

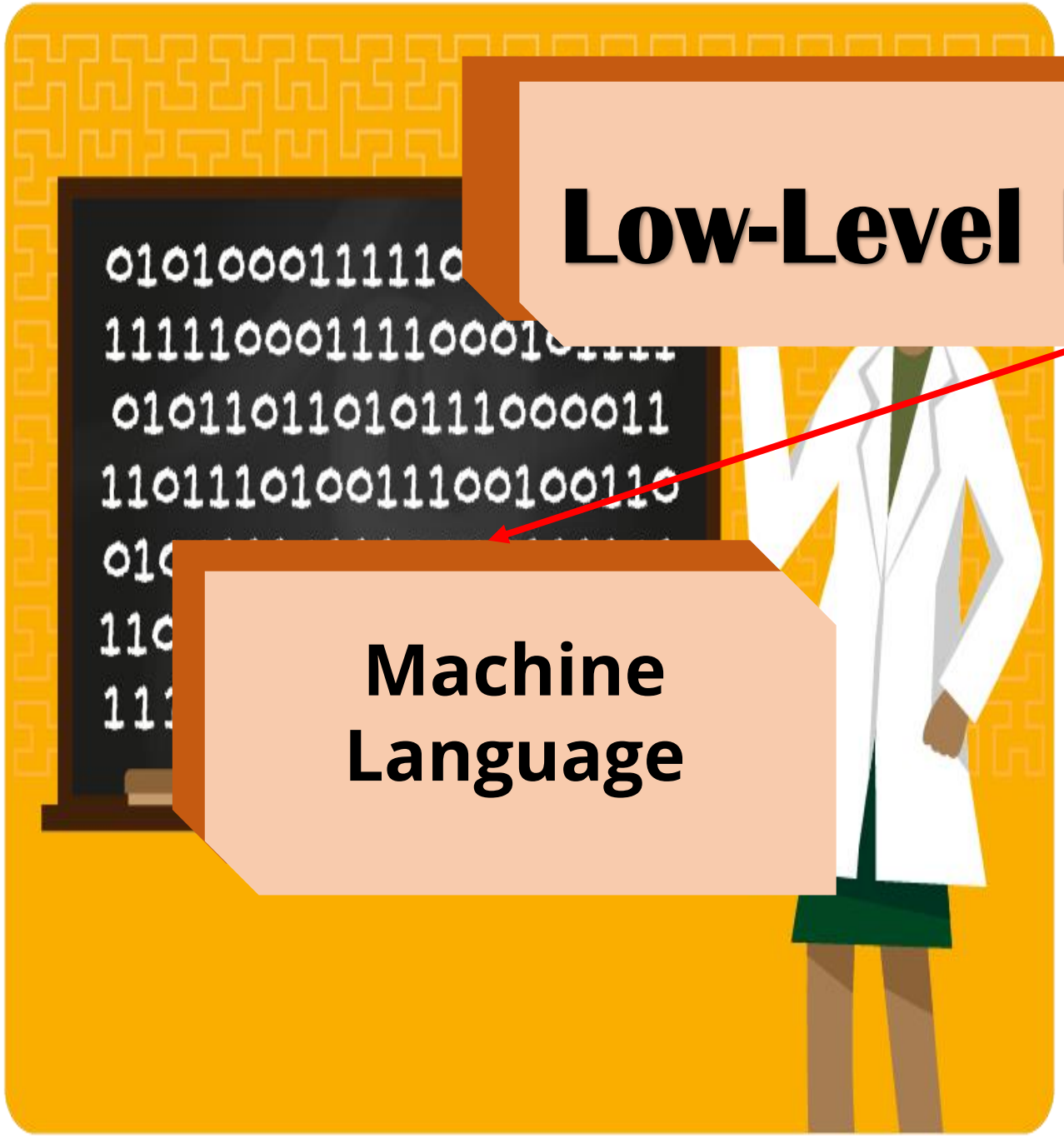
Low Level Languages

are very easily understandable by the machine.

help in operating, syncing and managing all the hardware and system components of the computer.

Main function of low-level languages is to interact with the hardware of the computer.

Low-Level Languages

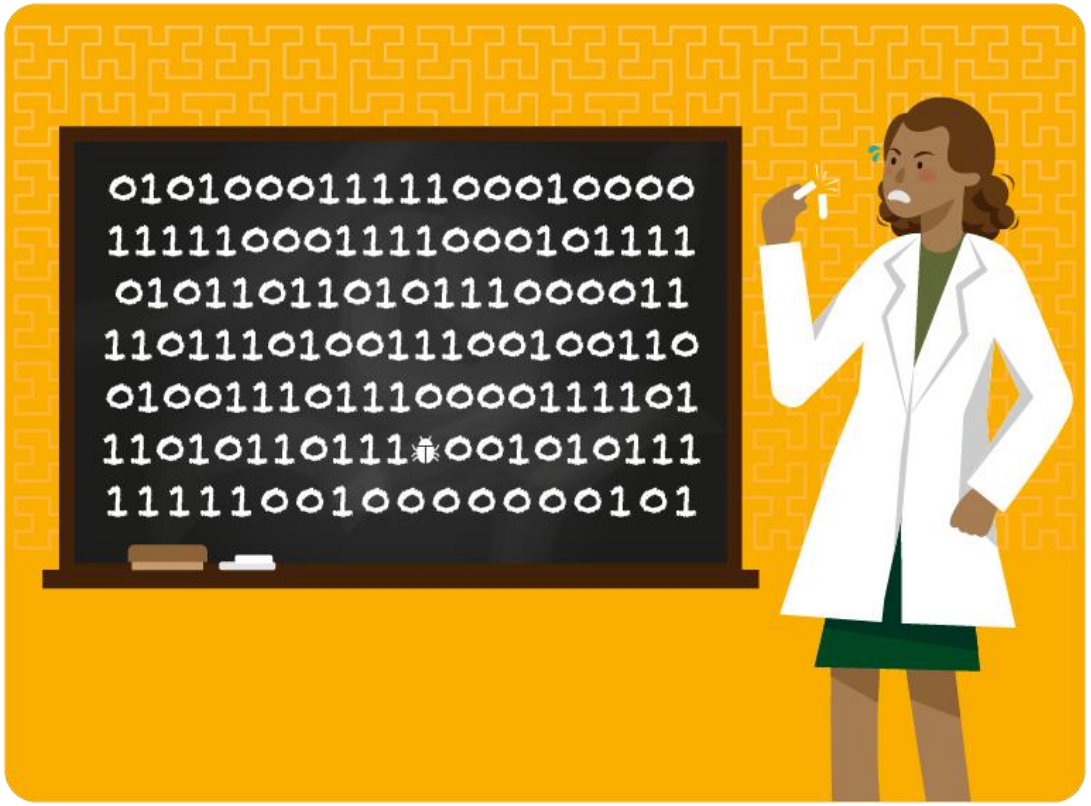


Machine Language

Assembly Language

```
C003 86 13     INITA   LDA A  #RESETA   RESET ACIA
C005 B7 80 04          STA A  ACIA
                       #CTLREG   SET 8 BITS AND 2 STOP

                       ON    GO TO START OF MONITOR

                       ********************
                       * FUNCTION: INCH - Input character
                       * INPUT: none
                       * OUTPUT: char in acc A
                       * DESTROYS: acc A
                       * CALLS: none
                                                terminal

C010 B6
C013 47                                         G INTO CARRY
C014 24                                         ADY
C016 B6
C019 84 7F
C01B 7E C0 79          JMP     OUTCH     ECHO & RTS

                       ****************************************
                       * FUNCTION: INHEX - INPUT HEX DIGIT
                       * INPUT: none
                       * OUTPUT: Digit in acc A
```

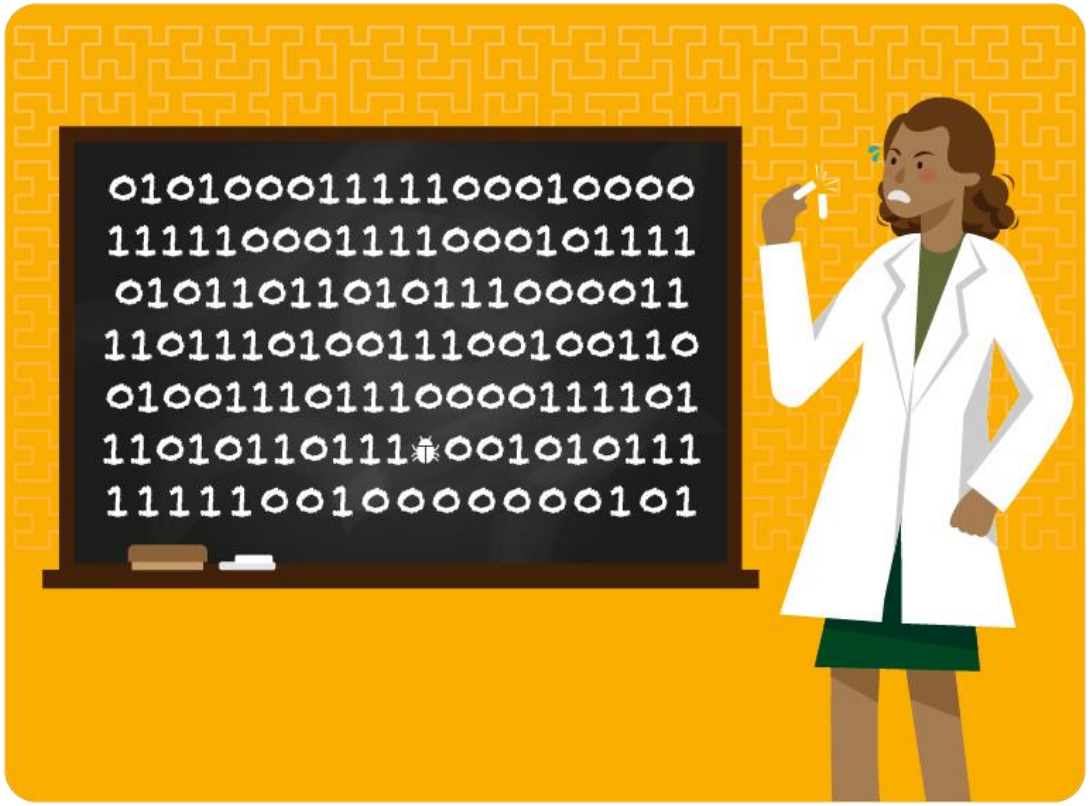
Machine Language



the only type of programming language that does not need to be translated before being run, because it is written in binary.

it is hard to read and write a program that consists of just 0s and 1s.

Machine Language



the only type of programming language that does not need to be translated before being run, because it is written in binary.

it is hard to read and write a program that consists of just 0s and 1s.

Assembly Language

allow programmers to write machine code using a set of **‘mnemonics’** that represent the binary equivalent in machine code.

the mnemonics are an English language representation of the operation the machine will perform when executed.

Assembly Language

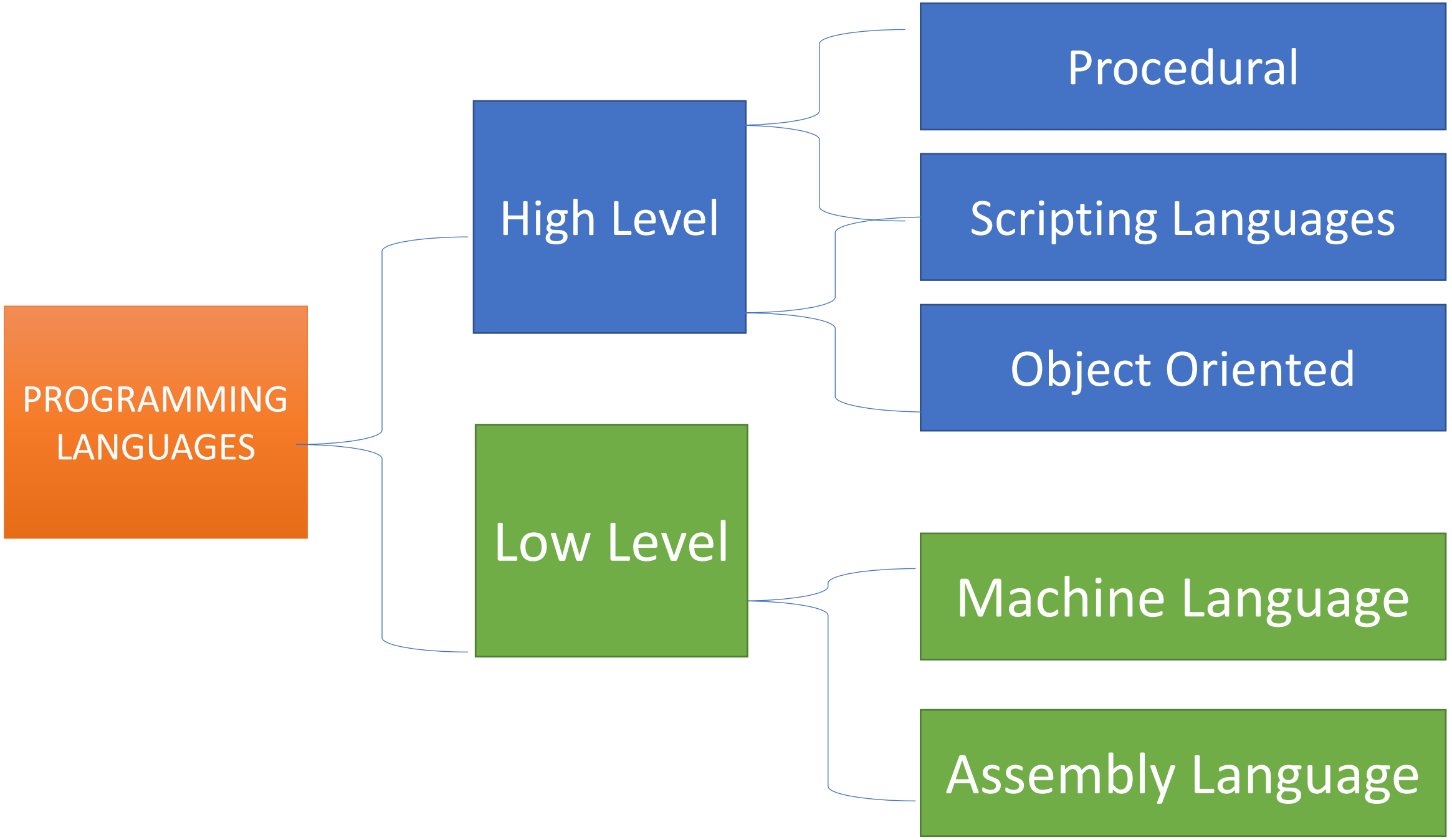
allow programmers to write machine code using a set of **‘mnemonics’** that represent the binary equivalent in machine code.

the mnemonics are an English language representation of the operation the machine will perform when executed.

OPERATION IN MACHINE CODE	ASSEMBLY LANGUAGE EQUIVALENT	MEANING
001	ADD	Adds a value into the accumulator
0101	LDA	Fetchs a value from a main memory location and loads it into the accumulator
0110	BRA	Branch (jump) to a specific instruction

Uses of Low Level Languages

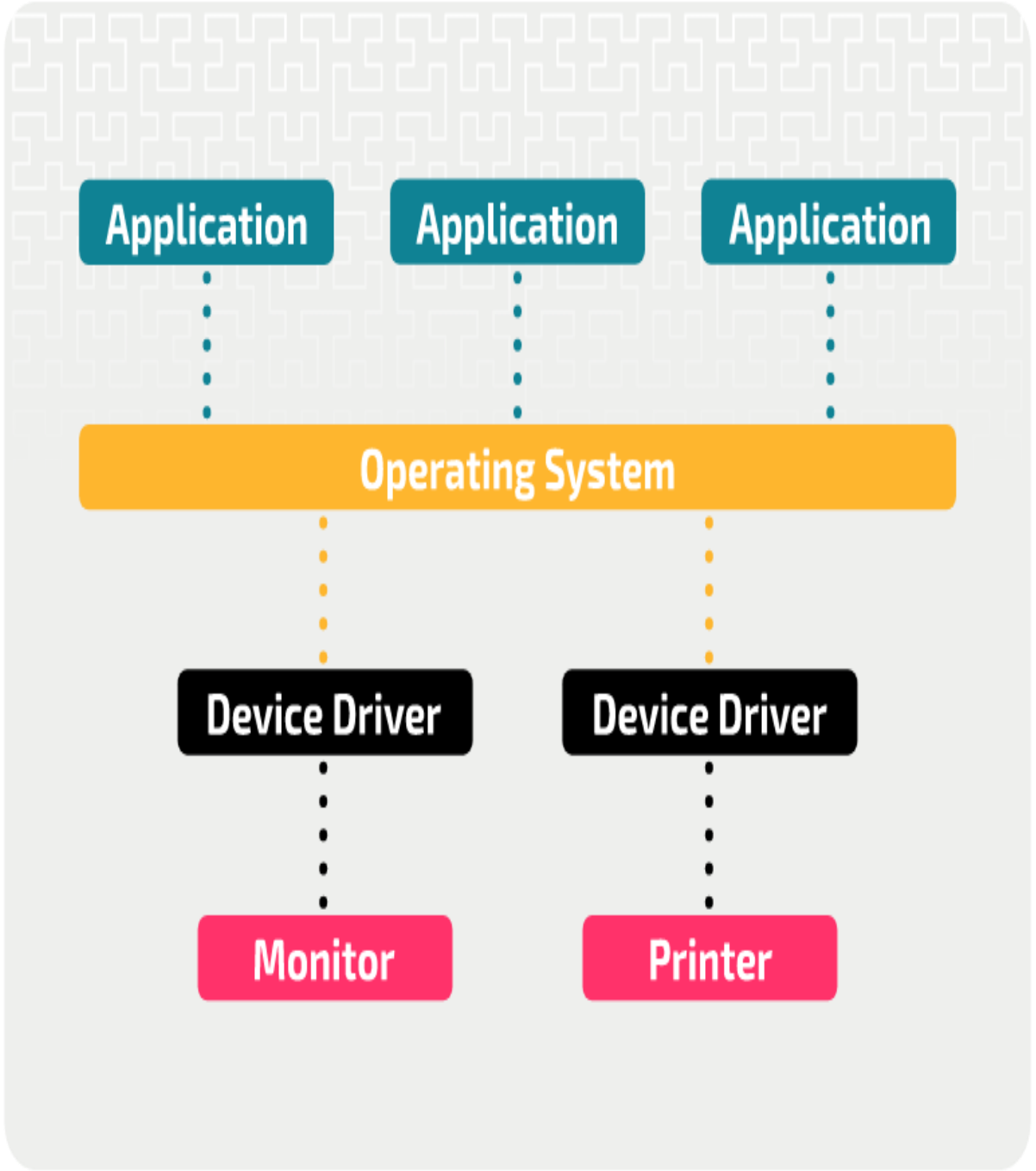
An **embedded system** is a complete computer system that has a dedicated function; examples include digital watches, domestic appliances, and central heating systems. Embedded systems are designed to complete a limited number of specific tasks very efficiently.


PROGRAMMING LANGUAGES
High Level
Low Level
Procedural
Scripting Languages
Object Oriented
Machine Language
Assembly Language

Device drivers

Device drivers are programs that allow a computer to interact with a peripheral, like a mouse or a printer.

Operating systems install and manage device drivers for every peripheral connected to a computer.



Uses of Low Level Languages

- A 1:1 relationship with processor instructions
- Non-portable (they only work on specific processors)
- Allow for individual control of a processor's components and registers

- Programs typically require less memory and execute faster than a high-level language program performing the same task

The advantages of low-level languages are:

•They allow a programmer to create optimised programs

•When a computer system has limited resources (processing power and memory) low-level languages allow a programmer to more directly control how the resources are used

disadvantages of low-level languages:

•It's more difficult to write programs in low-level languages than in high-level languages; programmers need to have a very good understanding of the hardware being used.

•Low-level languages **are not portable,** because In contrast, a high-level language program can be compiled or translated for different processors.

•Unlike high-level languages, low-level languages **do not have libraries of functions that can be imported and used by the programmer.**