GENERATIONS OF PROGRAMMING LANGUAGE

ASSEMBLY LANGUAGE

An assembly language is a low-level programming language for a computer, or other programmable device.

Assembly language is converted into executable machine code by a utility program referred to as an assembler.

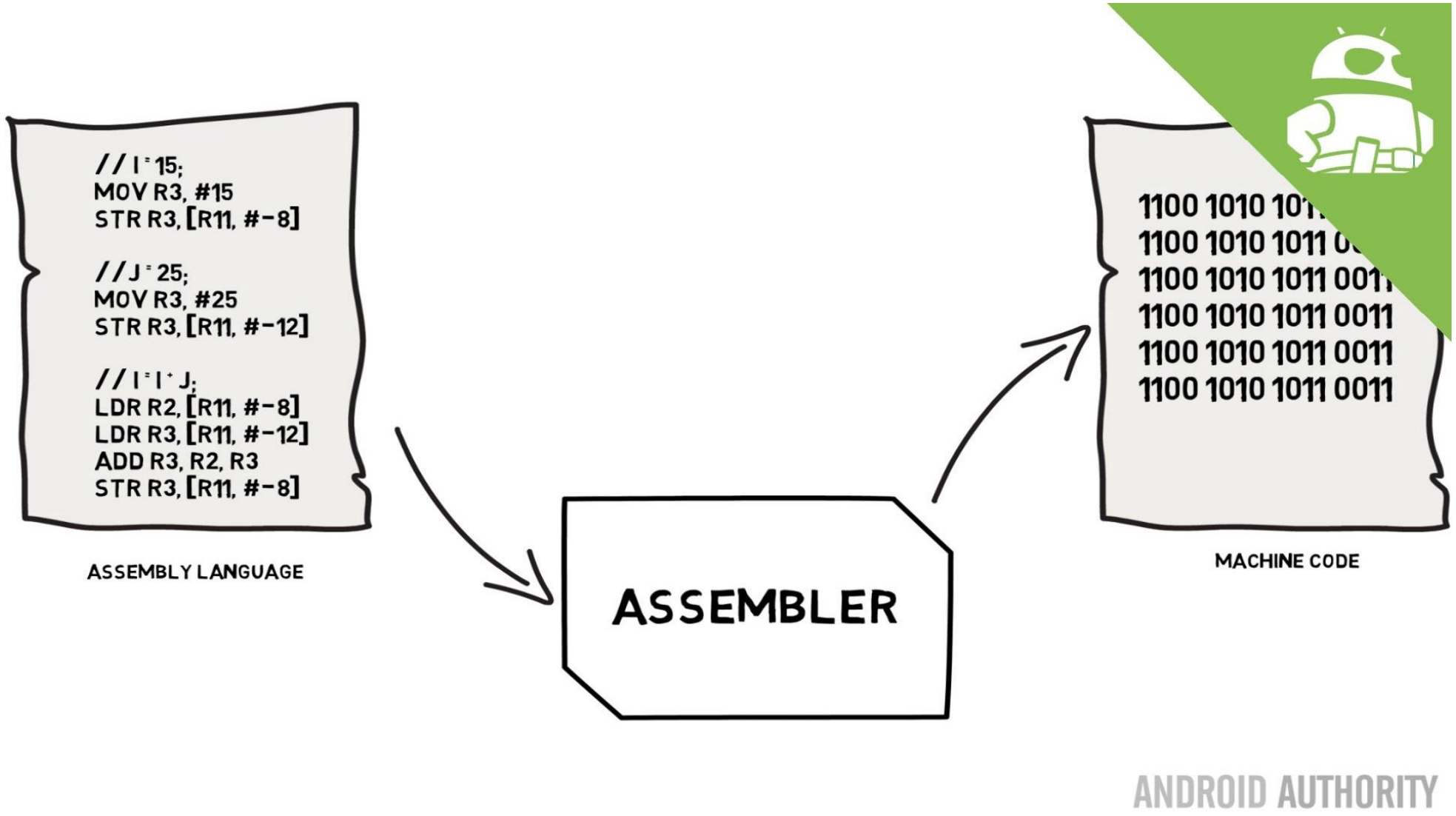
Advantages	Disadvantages
1)Very fast because of direct conversion from assembly language to binary 2) Direct access to hardware features e.g. embedded computers in home appliances	1) Low level languages are difficult to learn 2) Lack of portability because assembly language is dependent on the make of processor

MACHINE CODE

Machine code, also known as machine language, is the elemental language of computers. It is read by the computer's central processing unit (CPU), is composed of digital binary numbers and looks like a very long sequence of zeros and ones.

e.g. 10001000 01010111 11000101 11110001 10100001 00010110

Each instruction performs a very specific task, such as loading a value into a register, or adding two binary numbers together.



DECLARATIVE PROGRAMMING

declarative programming is a programming paradigm—a style of building the structure and elements of computer programs—that expresses the logic of a computation without describing its control flow.

IMPERATIVE LANGUAGE

In computer science, imperative programming is a programming paradigm that uses statements that change a program's state. In much the same way that the imperative mood in natural languages expresses commands, an imperative program consists of commands for the computer to perform. Imperative programming focuses on describing how a program operates.