What is Java ?

Java is a bit like C (syntax is very similar)

There are however some important differences:

- No explicit pointers (no * & ->)
- Automated memory management (no malloc/free)
- Support for various Object Oriented constructs
- Greater platform independence (WORA)...

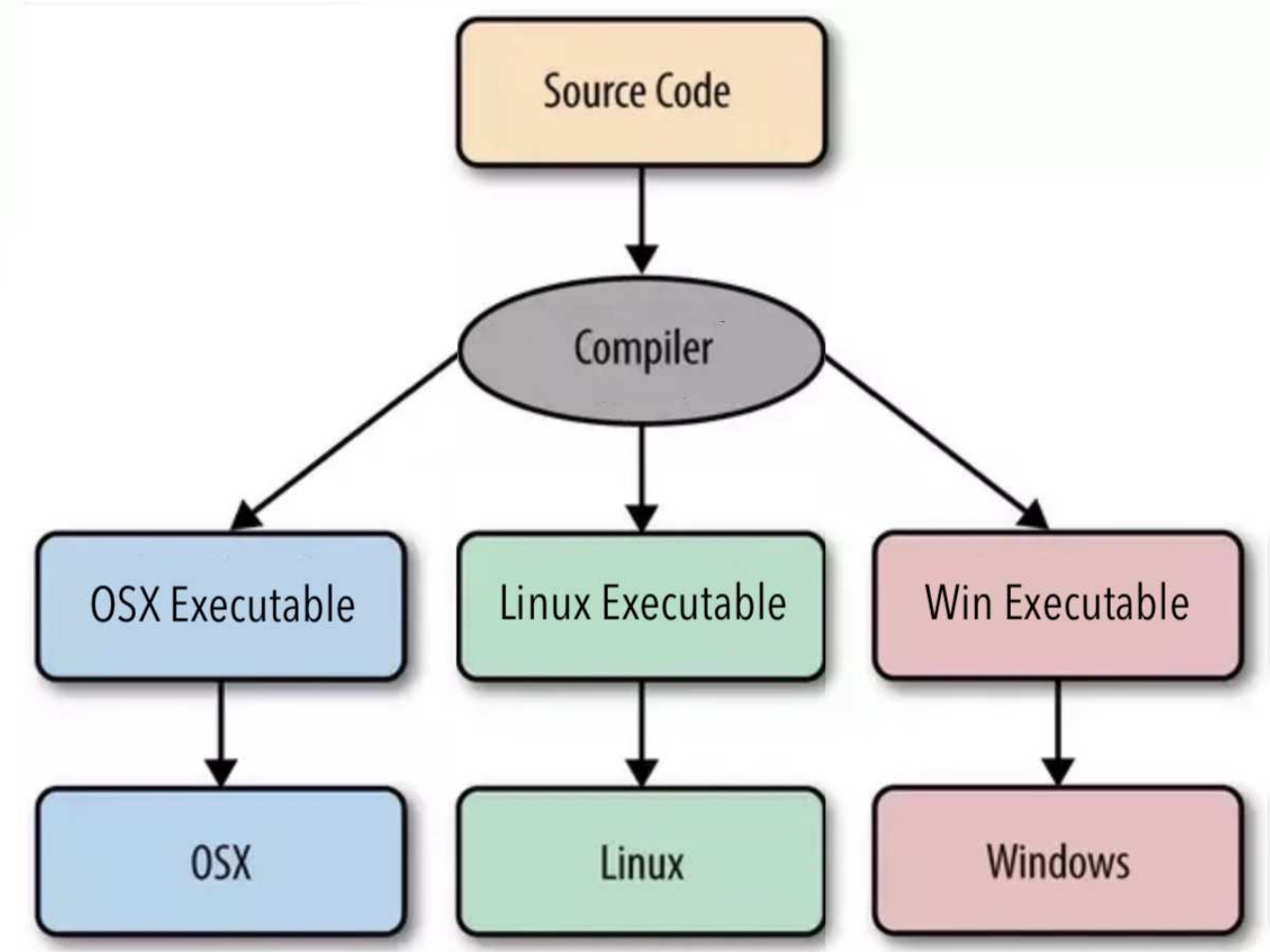
Write Once, Run Anywhere

Power feature: Java runs the same on all platforms
(Except Android Java which is different !)

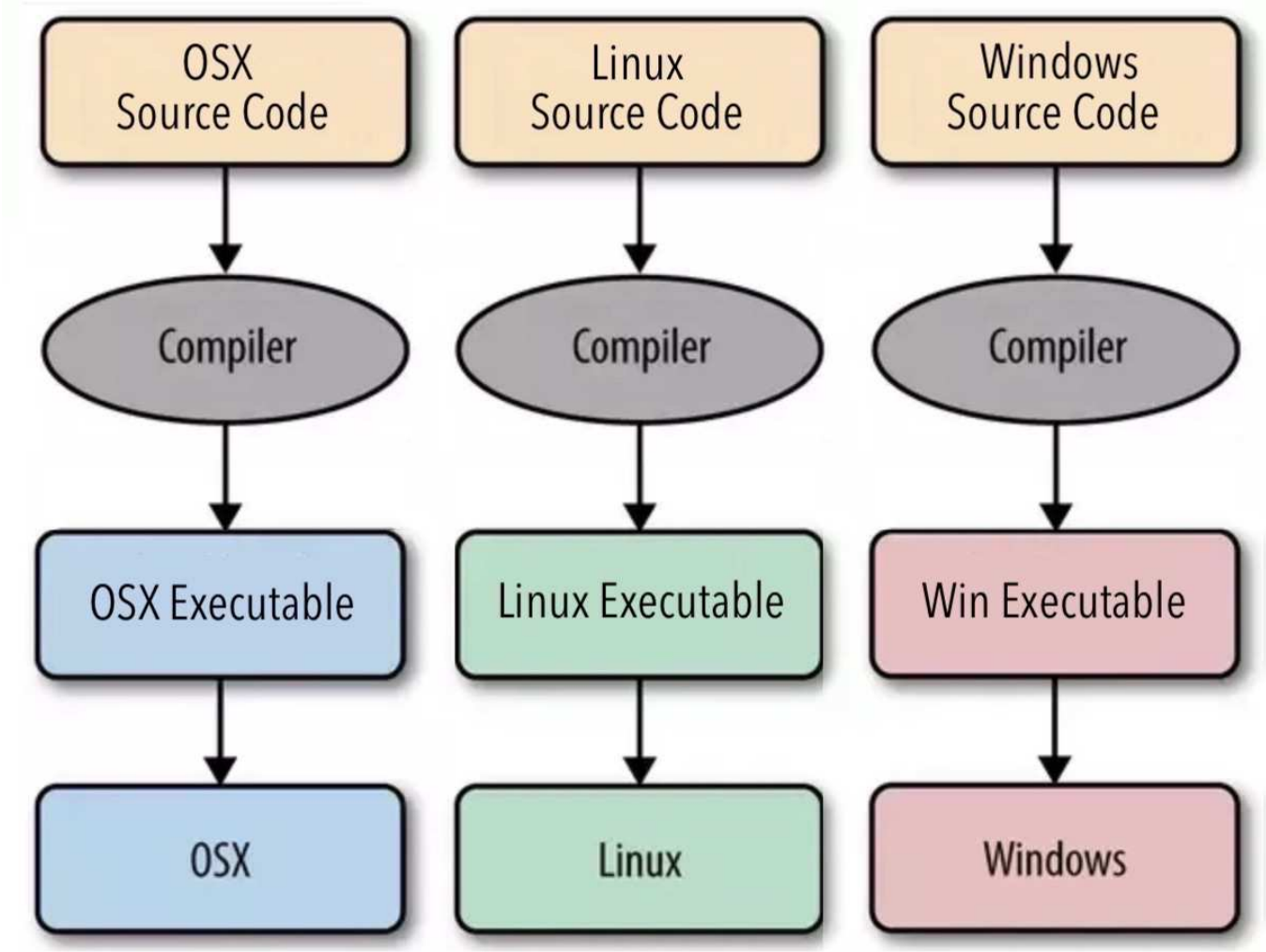
Source code compiled to cross-platform "bytecode"
(midway between source and binary executable)

Bytecode is then interpreted at runtime
By a standardised "Virtual Machine"
(That abstracts over the low-level detail of host OS)

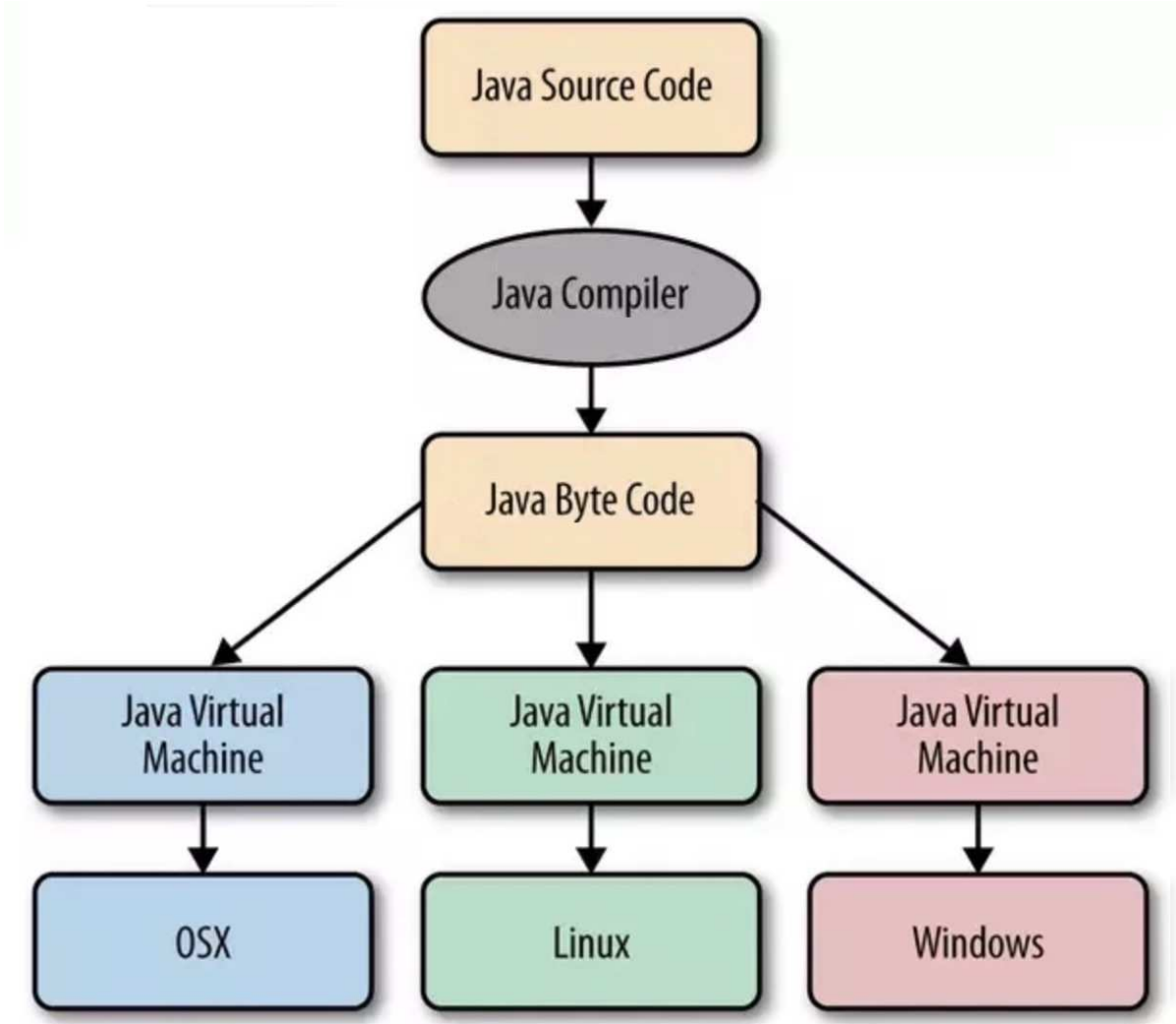
C Programming (in Theory)



Actual C Programming !



Java Programming



Performance

C has a reputation for being fast !

Java for being a little more "leisurely"

(Due to the overhead of Bytecode interpretation)

HOWEVER

Almost all Java Virtual Machines use "JIT" compilers

That convert the bytecode into native executables

"Just-In-Time" to be run

So the performance difference is not that big

Use of Java

Java is a very popular programming language
https://www.youtube.com/watch?v=Og847HVwRSI

Used for large servers, desktop applications,
mobile devices, embedded processors etc.

It has very little in common with JavaScript !
Other than a partially similar name
(and some common syntax)

What type of language ?

C was a procedural/imperative language

("do this and then do that")

With Functional Decomposition as main paradigm:

Functions call sub-funcs that call sub-sub-funcs

Java is also a procedural/imperative language

But its (intended) paradigm is Object Orientation:

Classes and Objects organise "functions" and data

Functional Decomposition (what C is)

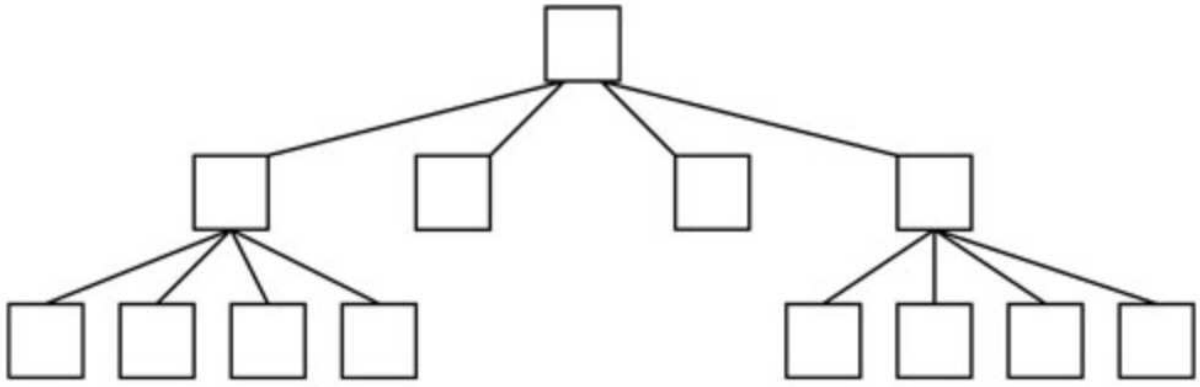
Main function is broken down into smaller functions

Forms a tree, with data flowing around everywhere

Although effective programs can be written this way

They tend to be monolithic (big and frightening)

Also "brittle" (resistant to long-term evolution)



Object Orientation

Program written as decentralised cooperating pieces

Each piece (Object) looks after its own internal data

Collaborate with each other to get the job done

Scales well to larger projects (if done properly !)

Works effectively for big teams (if done properly !)