

# Computer Language



Introduction to Java



# Computer: Hardware



# Computer: Software



# Computer: Programming Language

---

## ■ Machine language

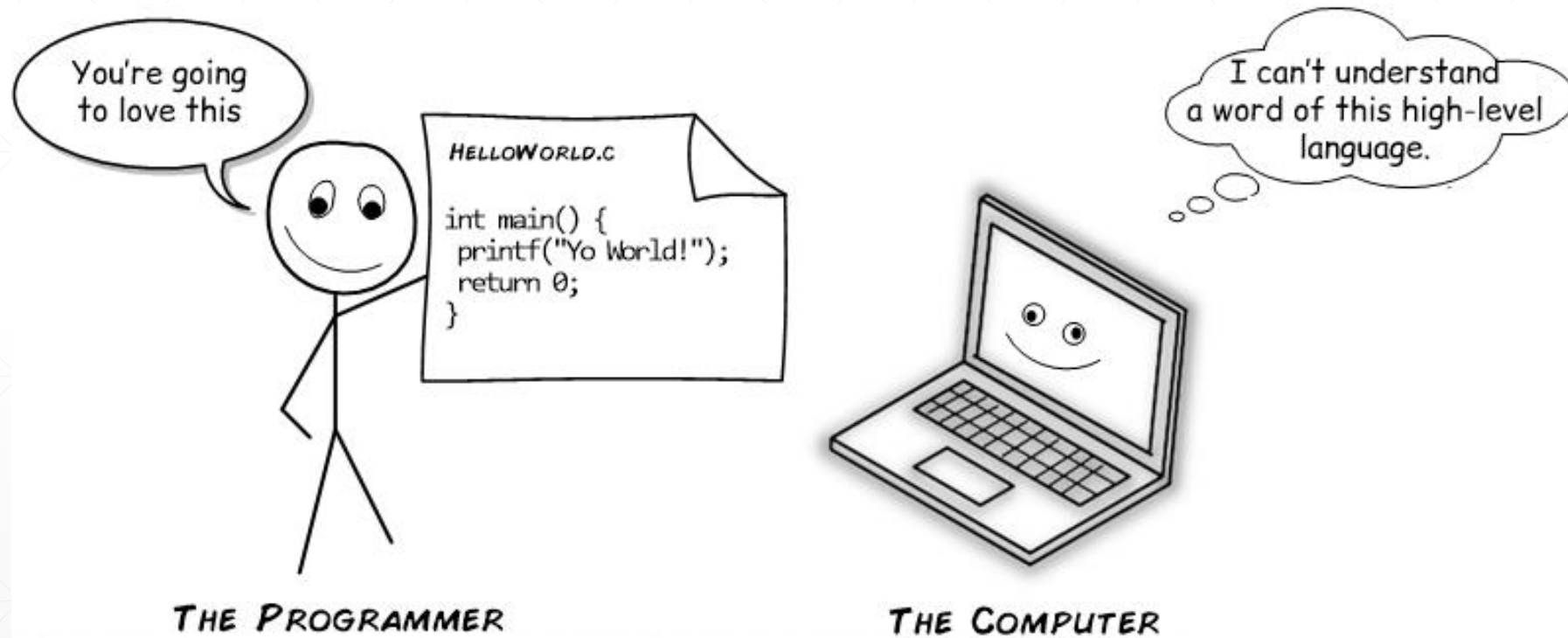
- Machine-readable code
- Binary representation (0 & 1)
- CPU only understands this language

## ■ (High-level) Programming language

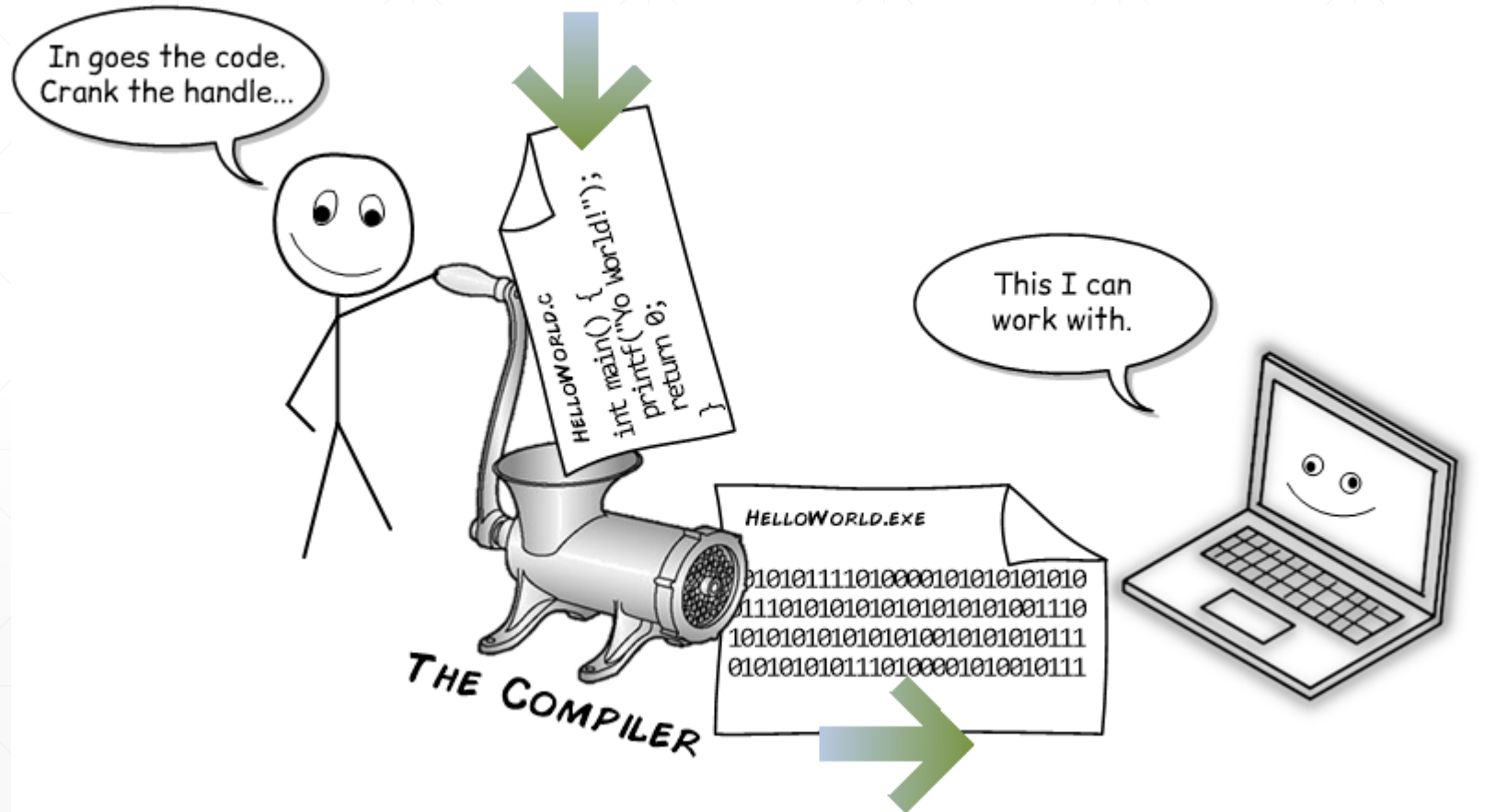
- Human-readable code
- Bridge between a machine and a human
  - But, CPU still cannot understand human-readable representation



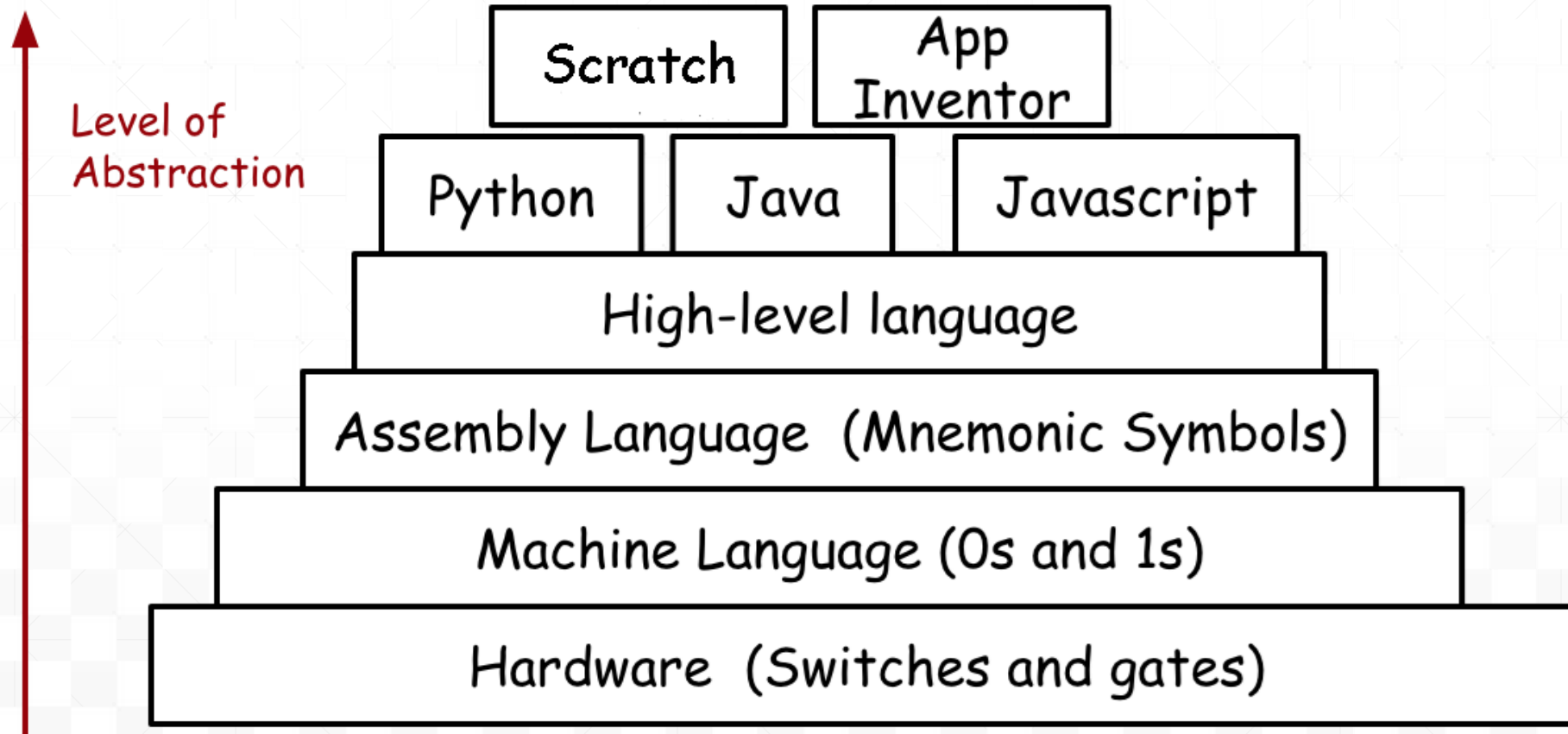
# Computer: Programming Language (cont'd)



# Computer: Programming Language (cont'd)



# Computer: Programming Language (cont'd)



# Computer: Programming Language (cont'd)

---

## ■ Source file

- Text file written in a (High-level) programming language
- Example) .c, .java file

## ■ Compile

- Translation from high-level to low-level
- Translates source codes into machine instructions



# Java

## ■ Class-based, object-oriented programming language

- Originally, “Oak” programming language
- Developed by James Gosling
- Released in 1995 by Sun Microsystems
- Still very popular language to develop various kinds of applications
- Maintained by Oracle since 2010



James Gosling

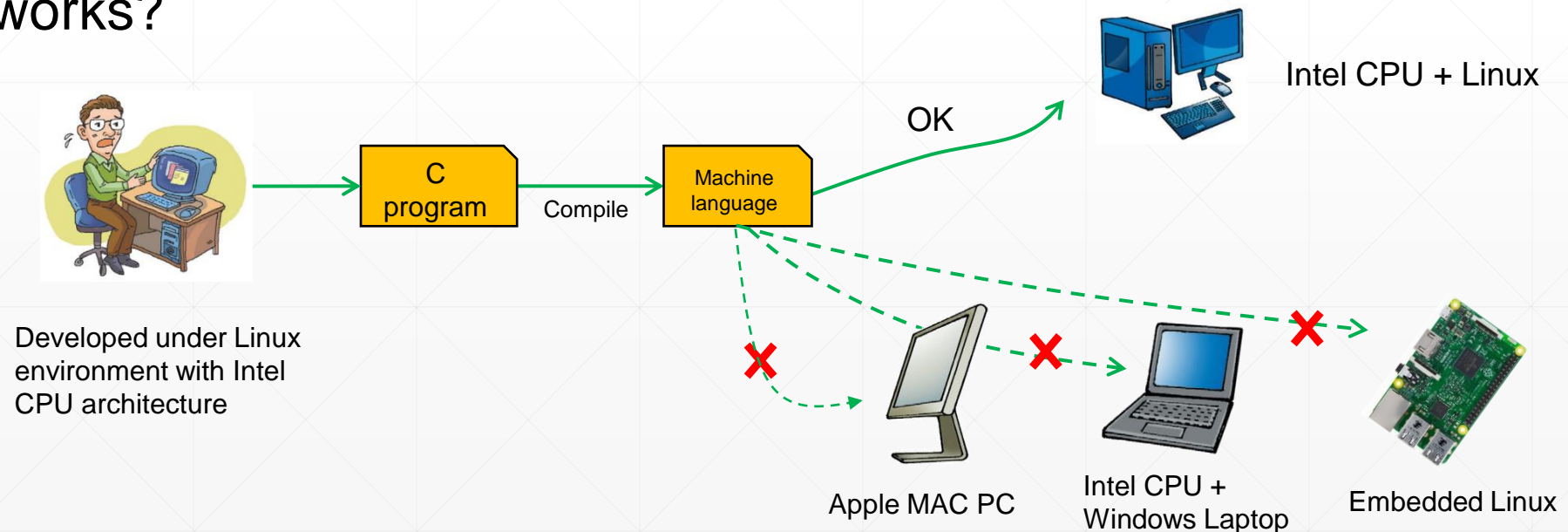
- Father of Java programming language

# Java (cont'd)

## ■ WORA (Write Once Run Anywhere)

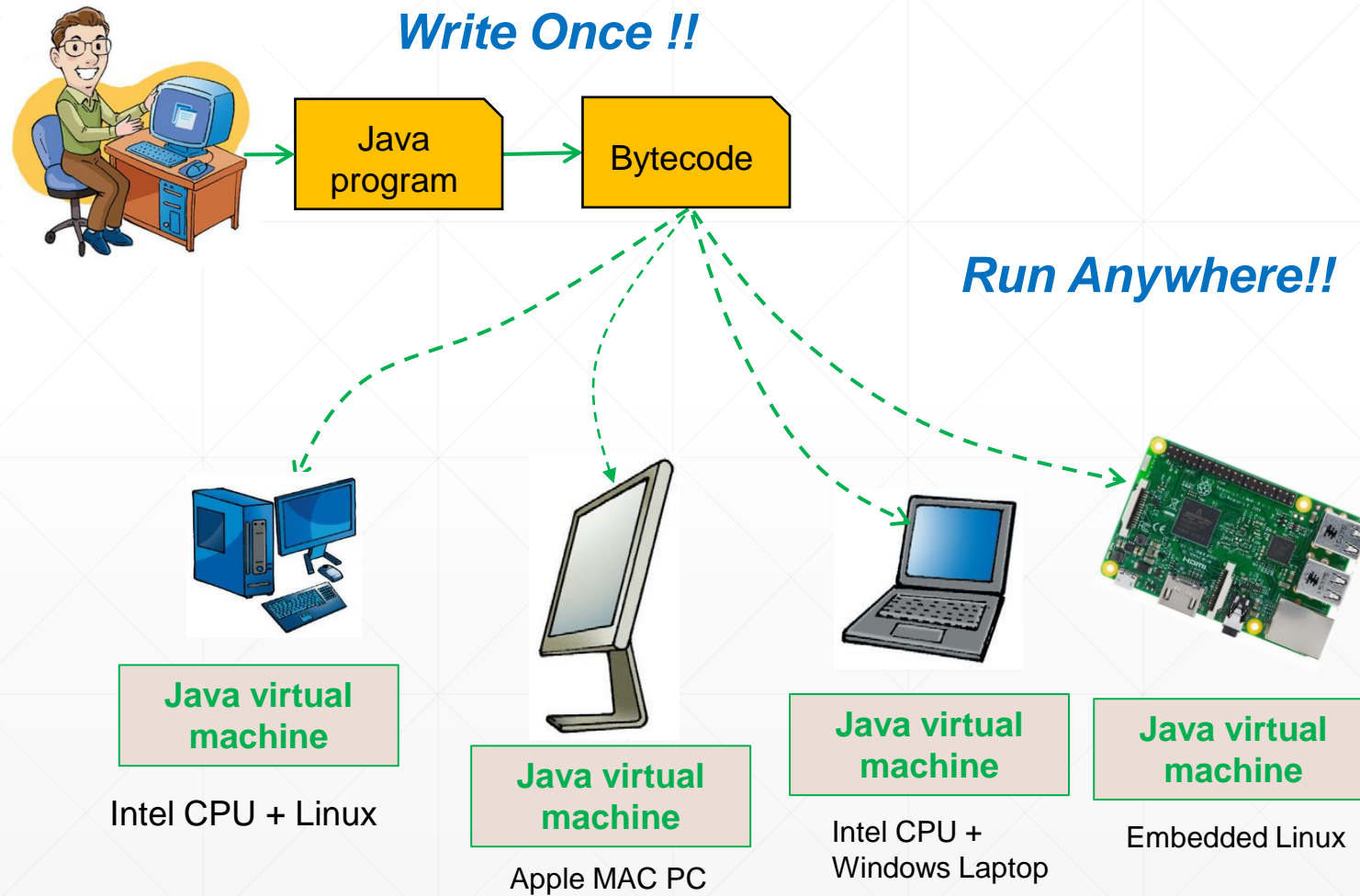
- Platform independent (cross-platform)
  - Weakness of traditional languages like C, C++, etc.
- A program written in Java can be executed on any architecture

## ■ How C works?



# Java (cont'd)

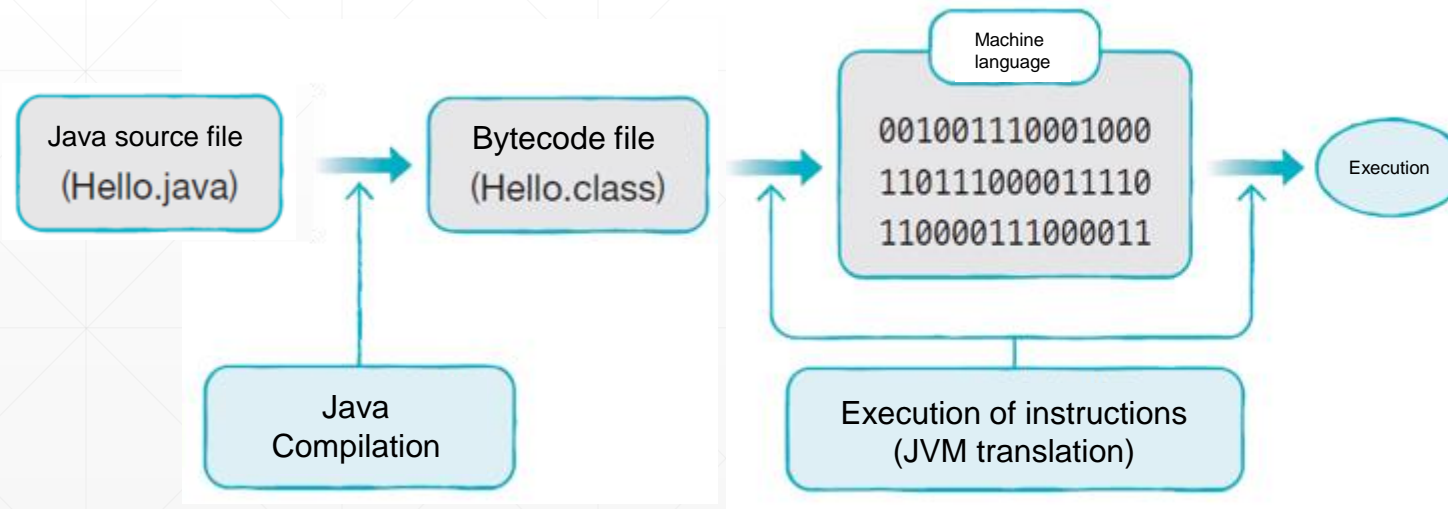
## ■ How Java works?



# Java (cont'd)

## ■ WORA (Write Once Run Anywhere)

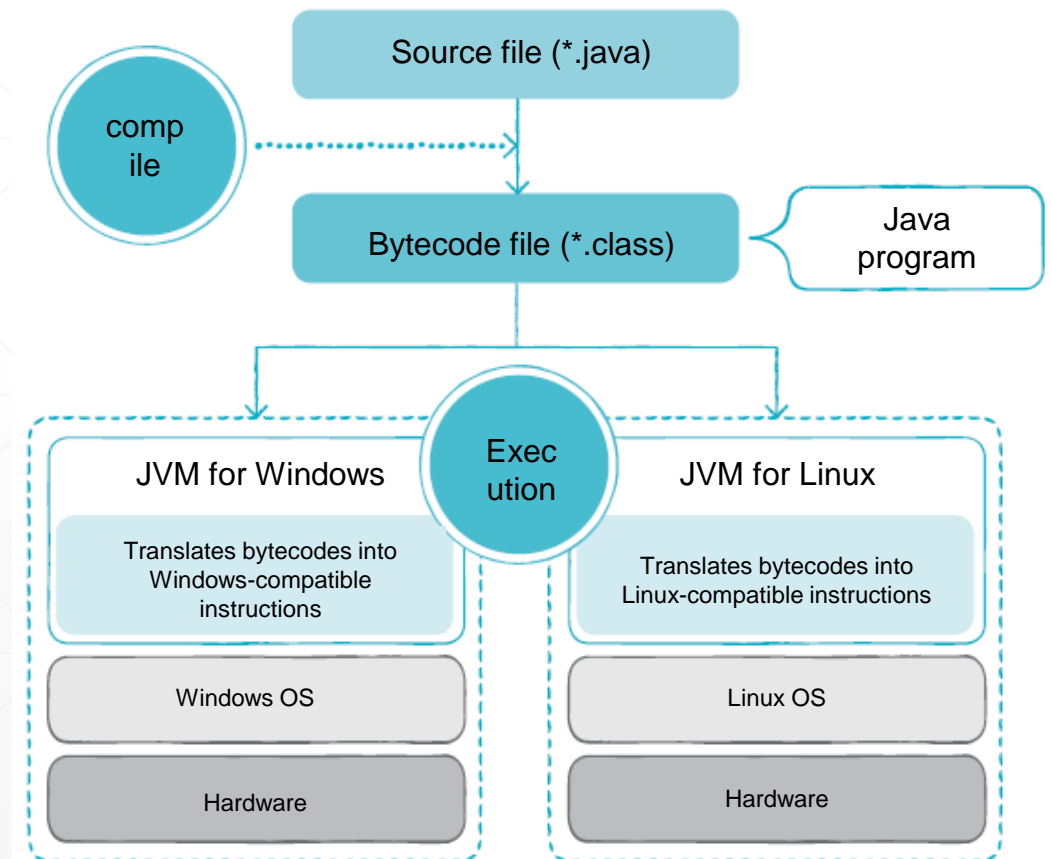
- Java programs consist of byte code files (.class), not complete machine language
  - Bytecode files cannot be executed directly on the OS
- Java Virtual Machine (JVM) translates and executes complete machine language



# Java (cont'd)

## ■ Java Virtual Machine (JVM)

- Provides an equivalent Java execution environment
- JVM itself is platform dependent



# Java (cont'd)

---

- Object-oriented programming (OOP)

- Encapsulation
- Inheritance
- Polymorphism
- ...

- Memory management

- Multi-threaded

- Opensource libraries

- ...



# Java (cont'd)

## ■ JDK/JRE

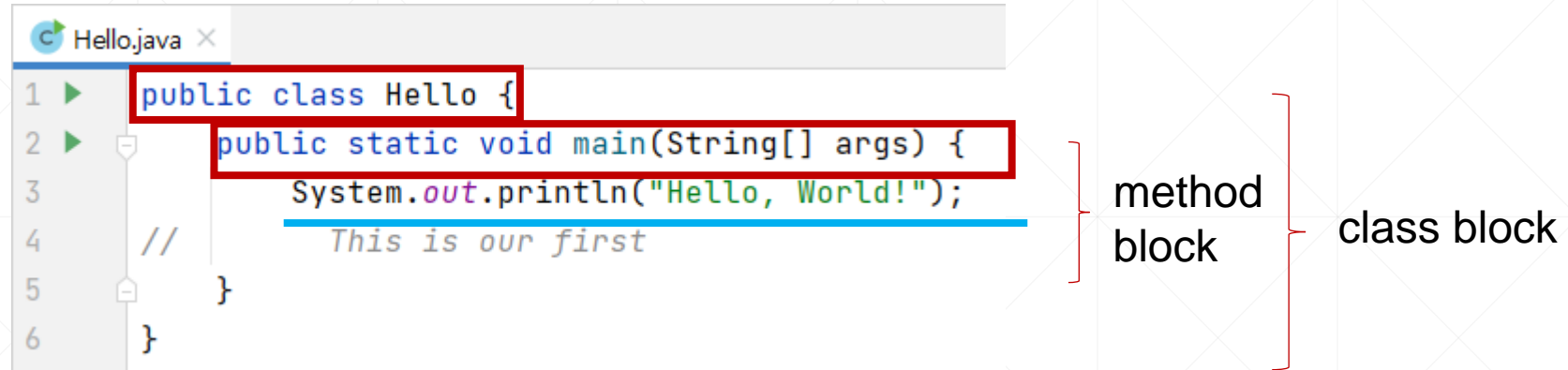
- JDK (Java Development Kit)
  - Environment for developing and executing Java programs
  - JRE + Development Kit
- JRE (Java Runtime Environment)
  - Environment for executing Java programs
  - JVM + Java standard class libraries

## ■ API/API reference

- Application Programming Interface (API)
  - Pre-built Java class libraries
- API Reference
  - Specification document for APIs
  - <https://docs.oracle.com/en/java/javase/11/docs/api/>

# Getting Started

## ■ Java project



- Defining class and methods
- Statement
  - Semicolon (;) must be appended to represent the end of statements
- Comment
  - Single line (`// ...`)
  - Multiple lines (`/* ... */`)