Programming Languages Pragmatics

INTERPRETATION VERSUS COMPILATION

DR. ERIC CHOU   IEEE SENIOR MEMBER

Purpose of Compilation and Interpretation

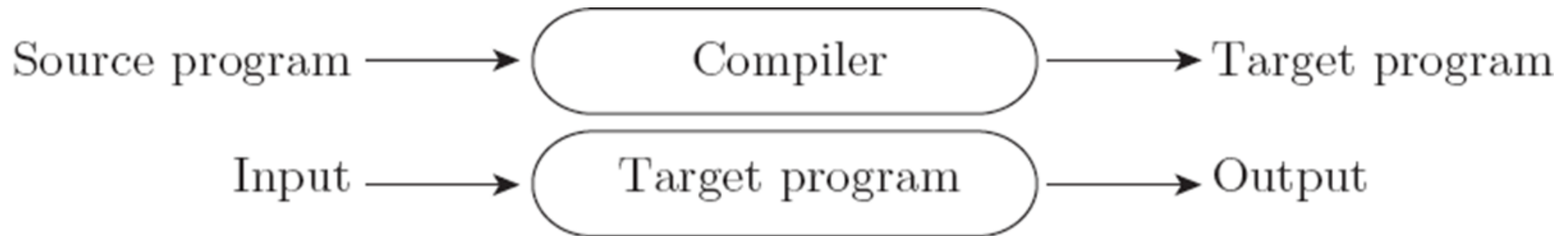Convert a language to another language or machine language for execution.

Pure Compilation

The compiler translates the high-level source program into an equivalent target program (typically in machine language), and then goes away:
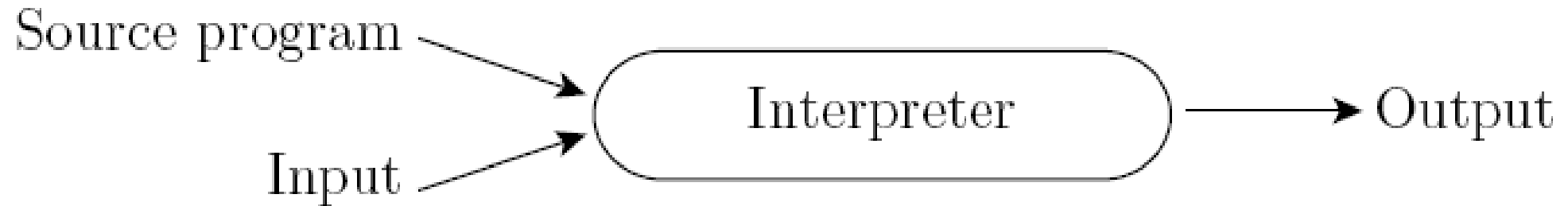

Is it OK to compile C into Java?

Pure Interpretation

1. Interpreter stays around for the execution of the program
2. Interpreter is the locus of control during execution

Source program ⟶ Interpreter ⟶ Output
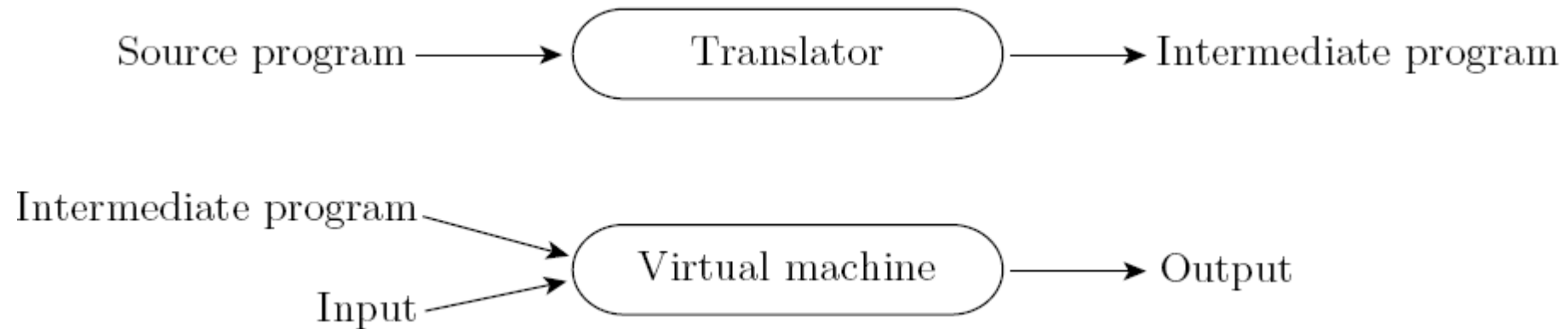Input ⟶

Comparison

- Interpretation:
  - Greater flexibility
  - Better diagnostics (error messages)

- Compilation
  - Better performance

Hybrids

- Common case is compilation or simple pre-processing, followed by interpretation

- Most modern language implementations include a mixture of both compilation and interpretation
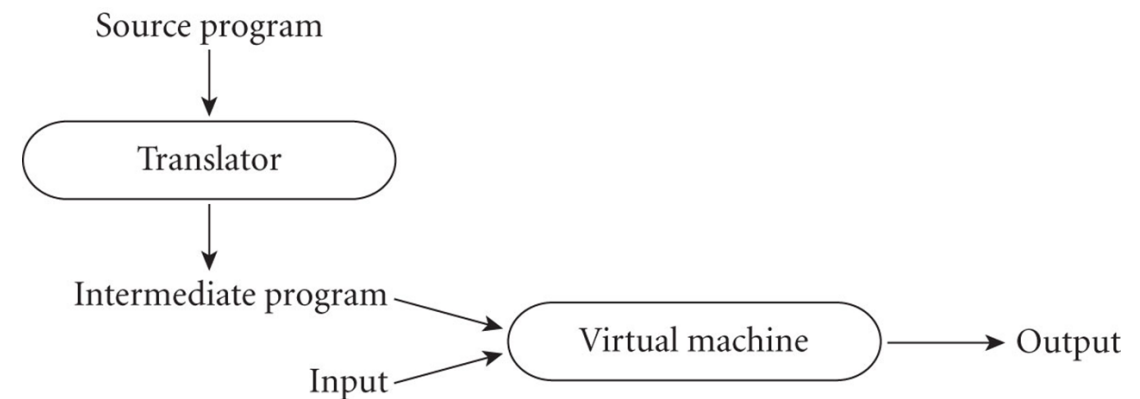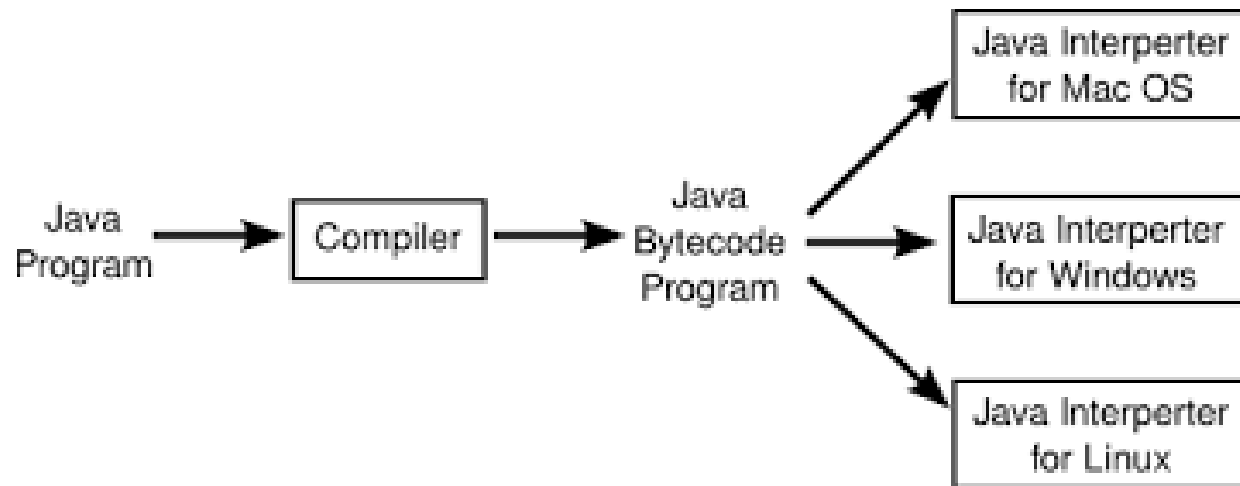
Source program ⟶ (Translator) ⟶ Intermediate program

Intermediate program ⟶ (Virtual machine) ⟶ Output
Input ⟶

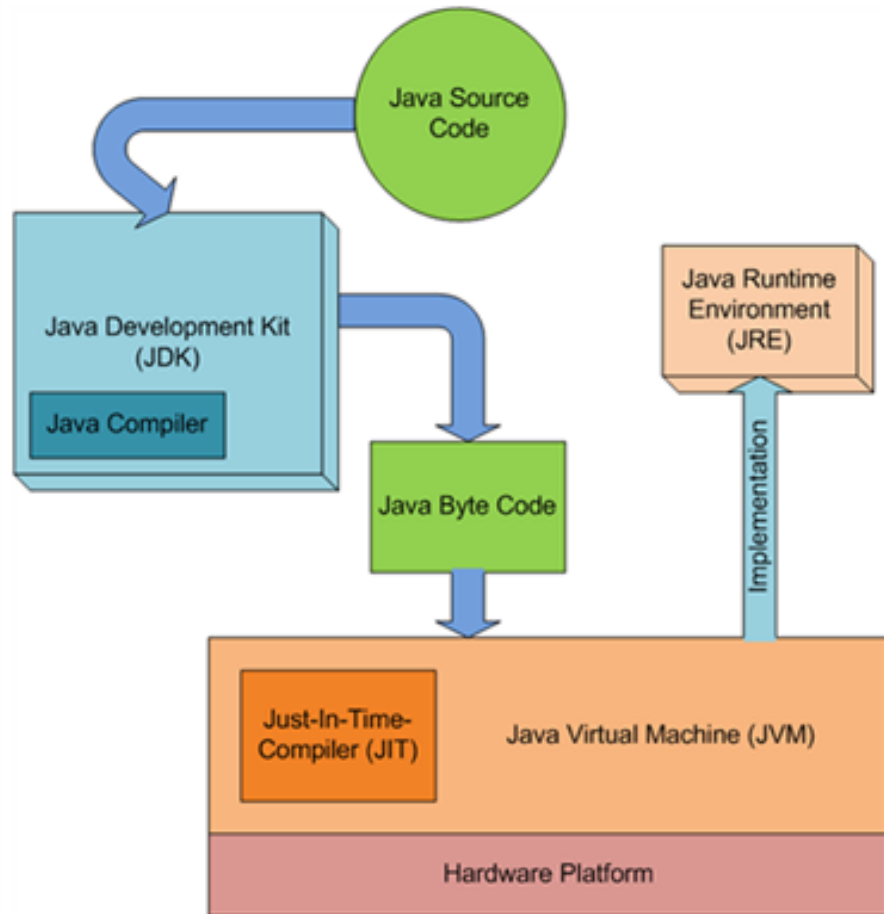# Virtual Machines

Step 1: Compilation from Java to Byte Code

Step 2: Executing the byte code on a machine with native codes

Just-In Time Compilation
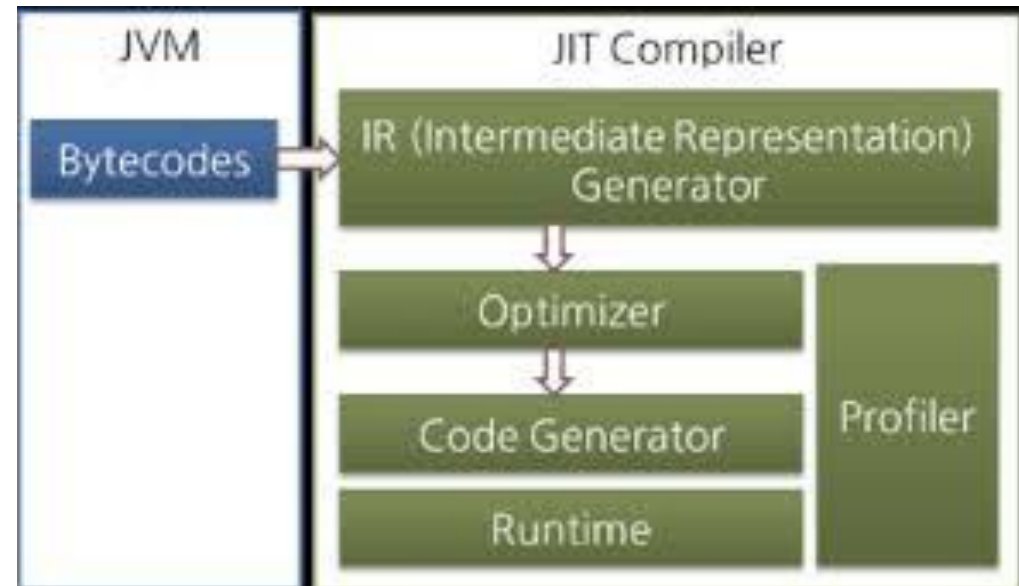Wait until you can no longer wait!

JDK
javac, jar, debugging tools, javap

JRE
java, javaw, libraries, rt.jar

JVM

Just In Time Compiler (JIT)

Java Source Code

Java Development Kit (JDK)

Java Compiler

Java Byte Code

Java Runtime Environment (JRE)

Implementation

Just-In-Time-Compiler (JIT)

Java Virtual Machine (JVM)

Hardware Platform

JVM

Bytecodes

JIT Compiler

IR (Intermediate Representation) Generator

Optimizer

Code Generator

Profiler

Runtime

Notes:

- Note that compilation does NOT have to produce machine language for some sort of hardware

- Compilation is translation from one language into another, with full analysis of the meaning of the input

- Compilation entails semantic understanding of what is being processed; pre-processing does not

- A pre-processor will often let errors through. A compiler hides further steps; a pre-processor does not

Notes:

- Many compiled languages have interpreted pieces, e.g., formats in Fortran or C

- Most use "virtual instructions"
  - set operations in Pascal
  - string manipulation in Basic

- Some compilers produce nothing but virtual instructions, e.g., Pascal P-code, Java byte code, Microsoft COM+