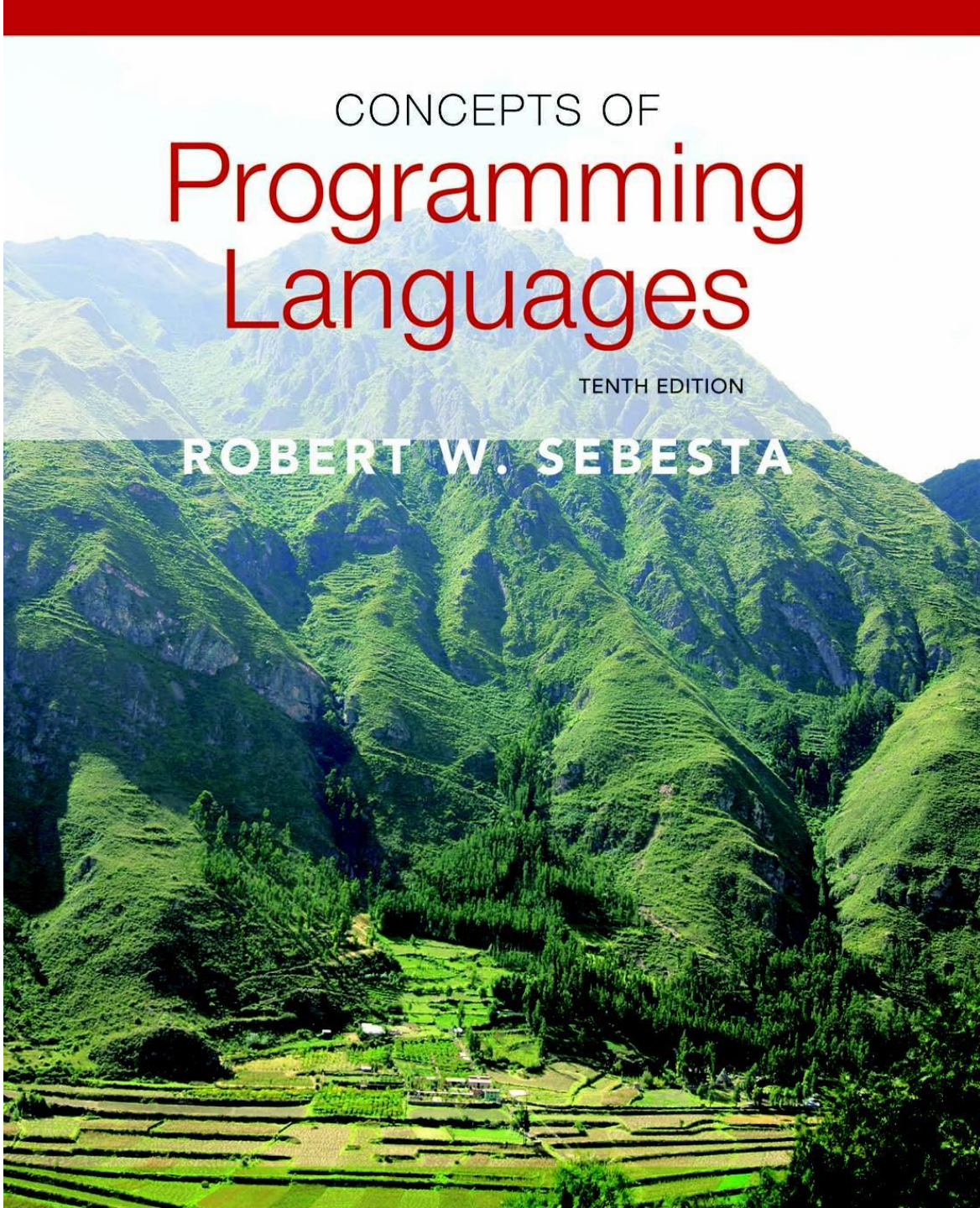
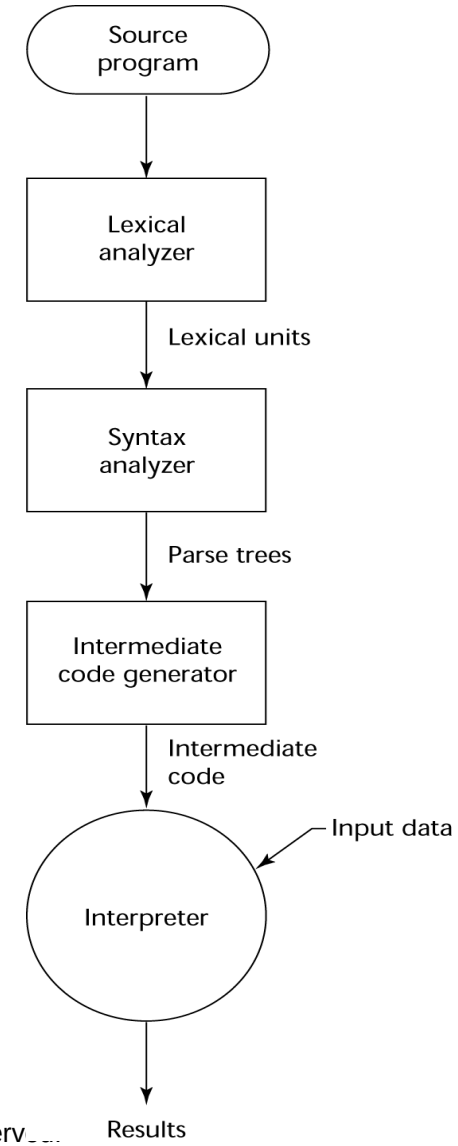
Chapter 1

Just-in-Time Compiler



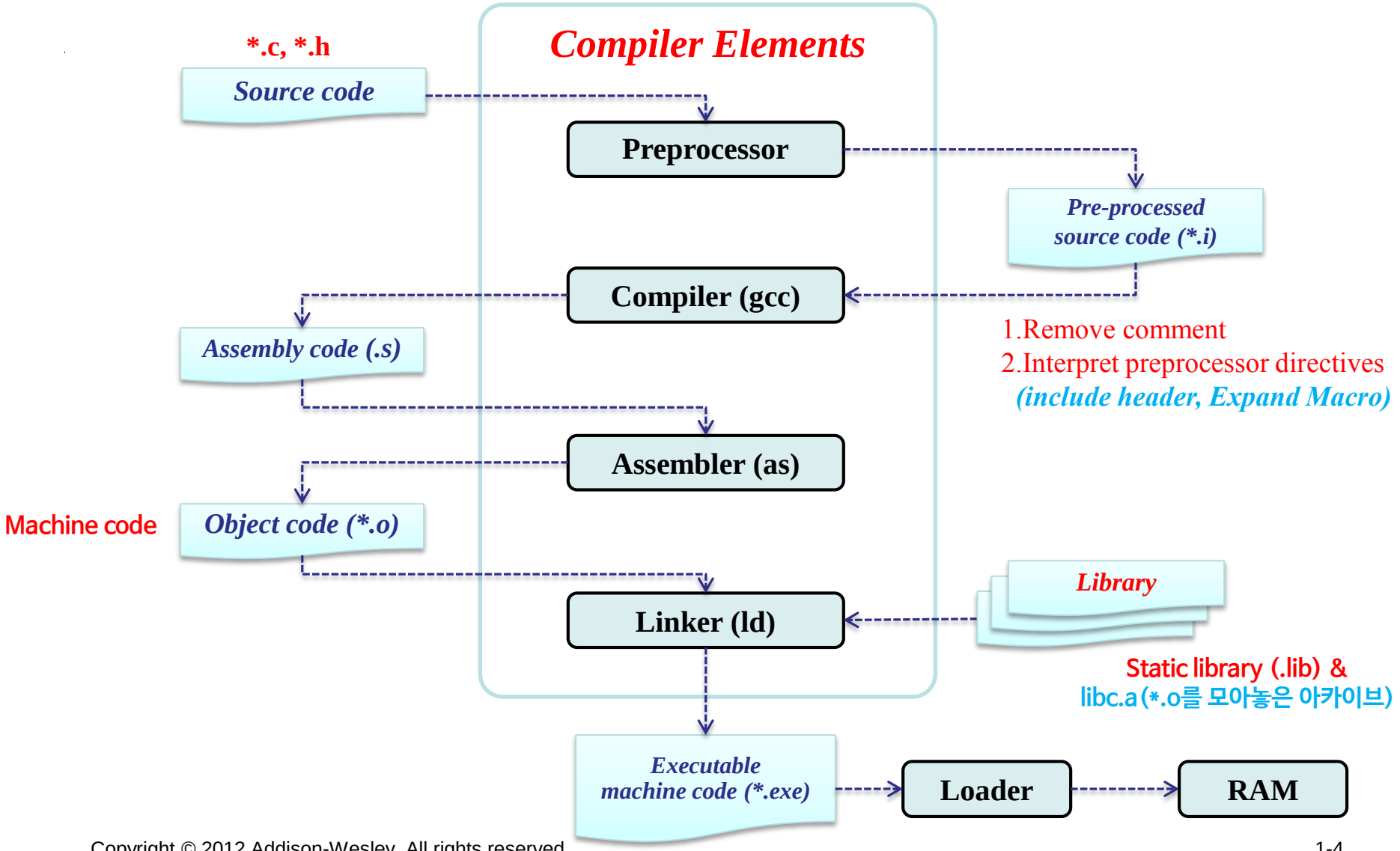
Hybrid Implementation Process



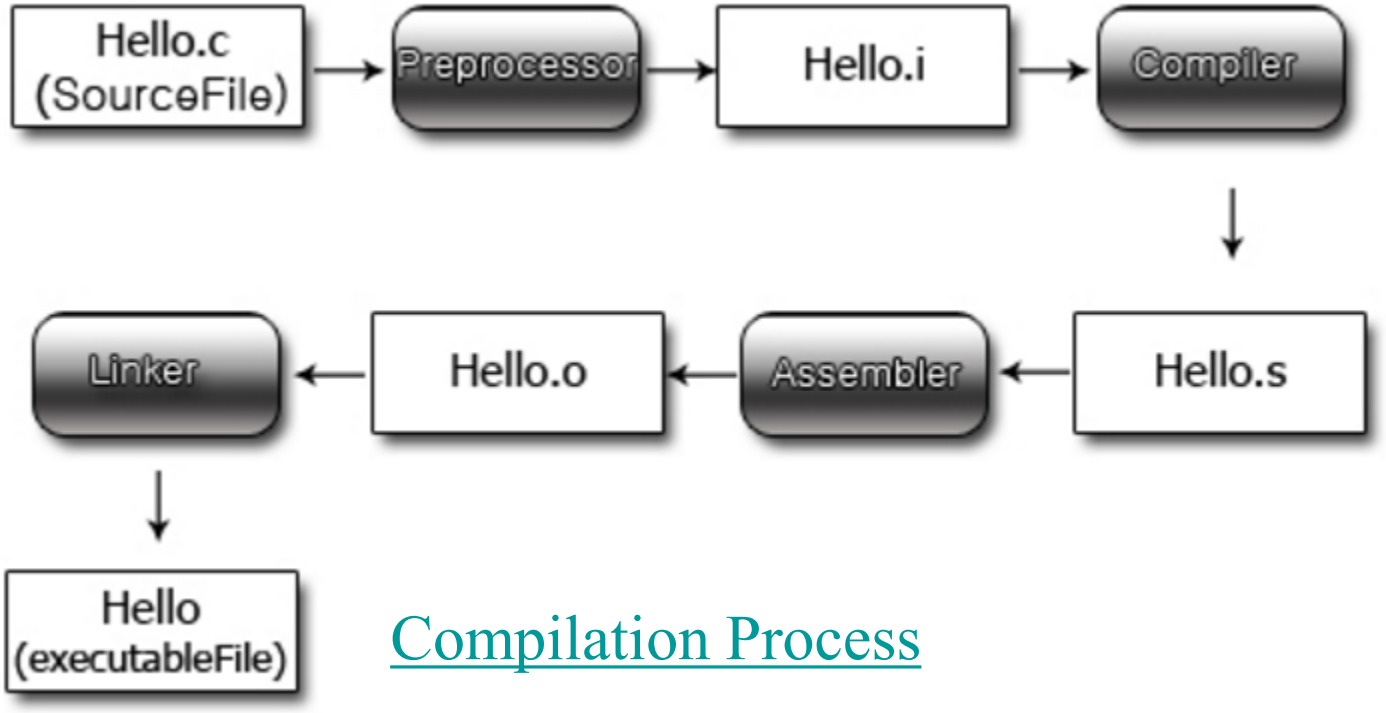
Traditional approaches to translation to machine code (Interpretation vs comilation)

- Interpretation
  - 작성된 원시코드(source code) 명령어 들을 한 줄씩 읽어 들여, 해당 기능에 대응하는 기계어코드(machine code)를 실행하는 방식
  - **매번 실행할 때마다 translation 필요 (overhead 발생)**
  - Interpreter
    - an interpreter is a computer program that directly executes, i.e. performs, instructions written in a programming or scripting language, without previously compiling them into a machine language program.
- Ahead-of-time compilation(AOT)
  - 정적 컴파일(Static compilation), 실행하기 전에 원시코드들을 기계어로 번역
  - **번역된 어셈블리어 또는 기계어 코드를 재사용**
  - Compiler
    - a computer program that translate source code from a high-level programming language to a lower level language (e.g., **assembly language** or **machine code**).

C Compilation Process (1/2)



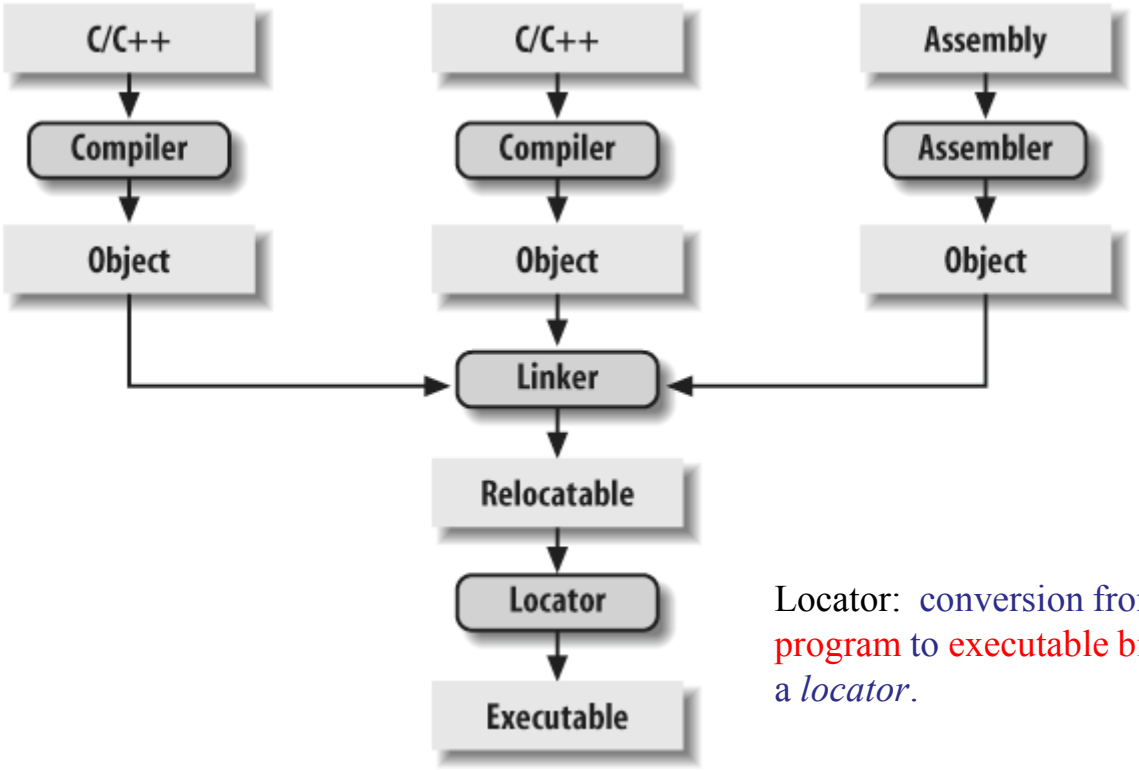
C Compilation Process (2/2)



Compilation Process

The embedded software development process

- *The embedded software development process*



Locator: conversion from relocatable program to executable binary image is called a *locator*.

Just-in-Time Compilation (1/2)

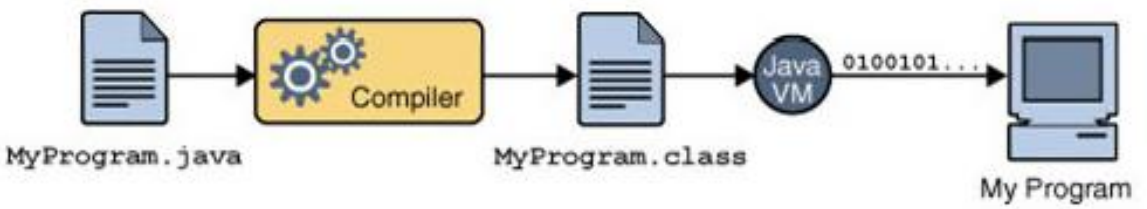
- Also known as **Dynamic translation**
- **프로그램을 실제 실행하는 시점에 기계어로 번역하는 컴파일 기법**
  - Then, during execution, JIT compiles the intermediate language methods into machine code when they are called
  - **실행 시점에서 기계어 코드를 생성하면서 그 코드를 캐싱하여, 같은 함수가 여러 번 불릴 때 매번 기계어 코드를 생성하는 것을 방지함**

- **JVM in JAVA**
- **Jit compiler in *.Net* Framework**

Just-in-Time Compilation (2/2)

- Bytecode – opcode (1byte, 00~FF)

Java JVM vs JIT Compiler in .Net



자바 원시 프로그램 --------> /javac --------> 자바 바이트 코드

(Helloworld.java) 자바 컴파일러 (Helloworld.class)

C:\>javac Helloworld.java

자바 바이트 코드 ---------> /java ----------> 실행결과

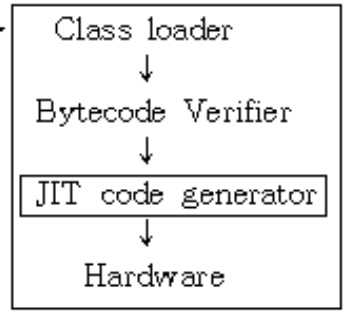
(Helloworld.class) 자바 인터프리터(JVM)

C:\>java Helloworld

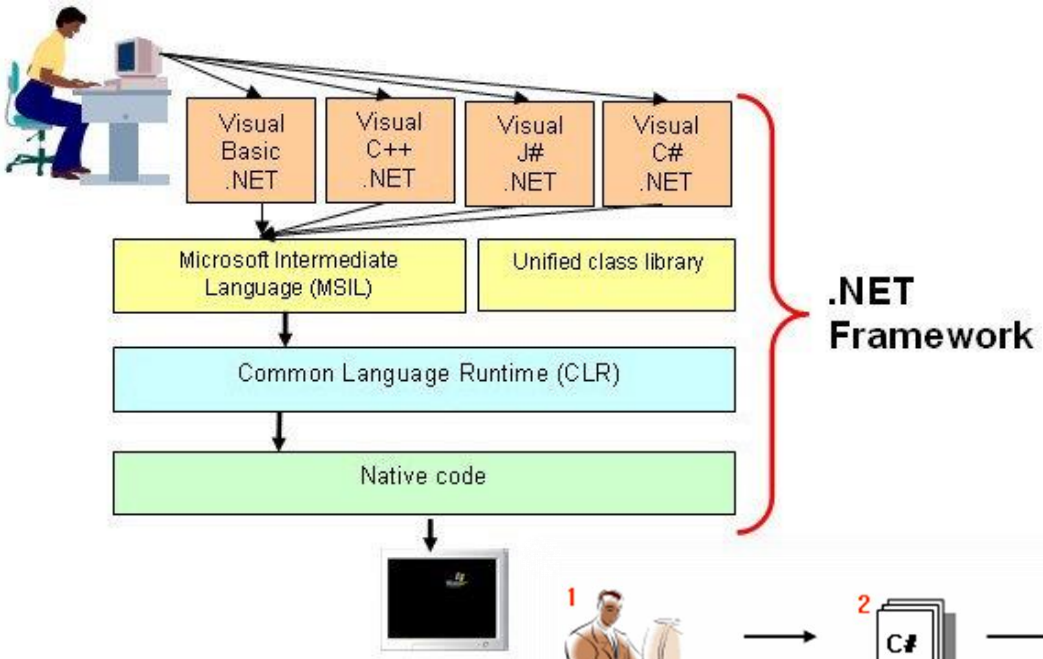
Loading byte code

*.class ---(network) -->

Verifying bytecode



JIT Compiler in .Net vs Java JVM



CIL (Common Intermediate Language)

- 공통 언어 기반과 닷넷 프레임워크에서 인간이 이해할 수 있는 가장 낮은 수준의 프로그래밍 언어(MSIL → CIL, 공식표기)
- 닷넷 프레임워크를 대상으로 하는 언어들은 **bytecode**로 변환되는 ***CIL***로 컴파일 됨
- **객체 지향 어셈블리어**이며 완전한 스택기반
- VM을 통해 실행