Java Overview

What is Java?

- Programming language developed in early 1990s by Sun Microsystems (now part of Oracle)
- Based on C/C++
  - Similar syntax, control, data structures
  - Imperative, object-oriented
- Originally designed for interactive television, found its initial success in Internet applications
  - Now viewed as a general-purpose programming language
- Currently enjoys widespread acceptance

Java: Compilation

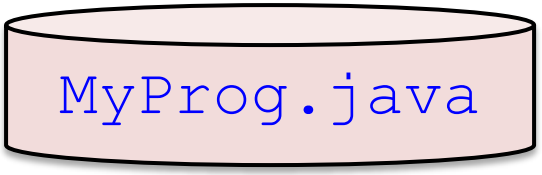
- The ***Java compiler*** checks the ***source code*** of a program in a `.java` file; if and only if there are no compile-time errors, it generates ***bytecode*** for that program and saves it in a `.class` file
  - Eclipse has its own Java compiler that continually and incrementally ***compiles*** source code even as it is being edited; a compile-time error is seen immediately and bytecode is available immediately if no errors
  - A batch-style compiler called ***javac*** is part of the Java Development Kit (JDK)

Java: Execution

- The ***Java Virtual Machine*** is a virtual computer
  - The JVM is just like any other program that runs on real physical hardware (e.g., an Intel chip) and operating system (e.g., Linux, Mac OS X, Windows)
  - The "launcher" of the JVM for your computer and OS ***loads*** the JVM and your program's `.class` file(s), and the JVM then ***executes*** your program by ***interpreting*** the bytecode that is loaded
  - A launcher called ***java*** and the JVM are part of the Java Runtime Environment (JRE) for your computer and OS

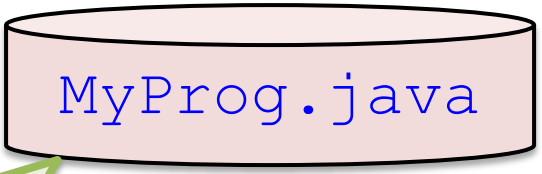
Java End-to-End

Java source code
(a text file)



Java End-to-End

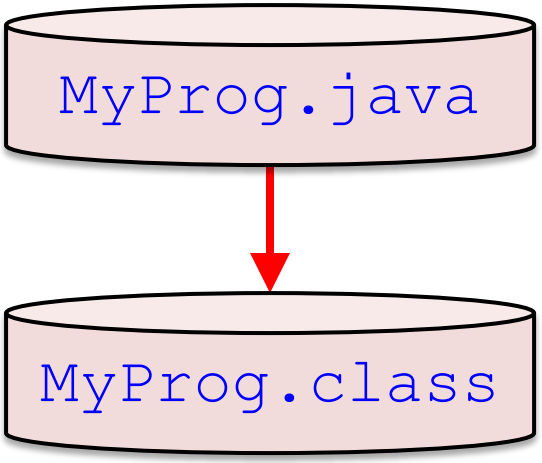
Java source code
(a text file)



Java compiler generates bytecode from (legal, though not necessarily “correct”) program source code.

Java End-to-End

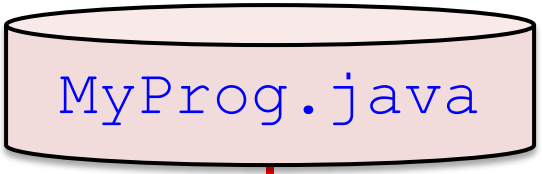
Java source code
(a text file)



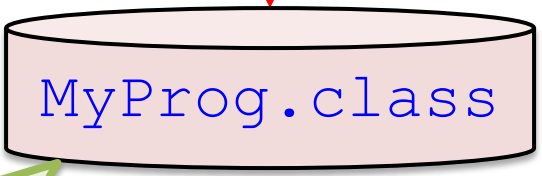
Java bytecode
(a "binary" file)

Java End-to-End

Java source code
(a text file)



Java bytecode
(a "binary" file)

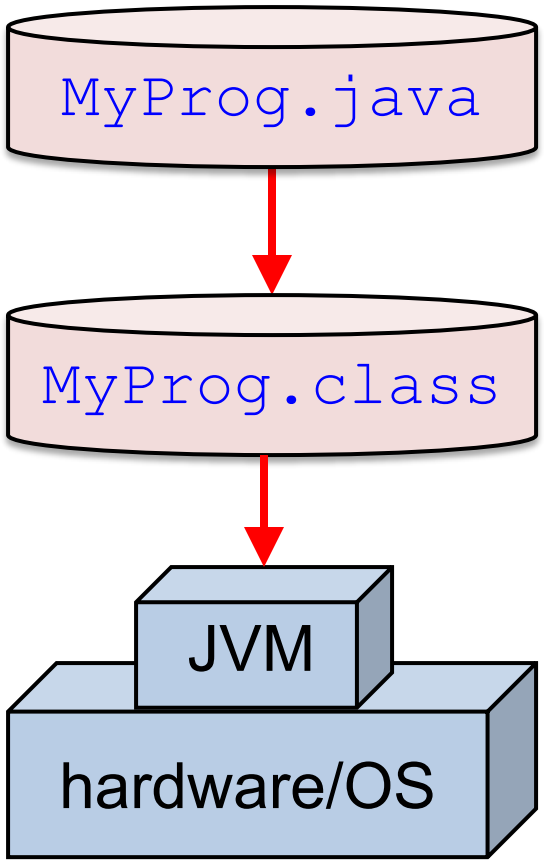


Launcher loads JVM and your bytecode, which JVM executes by interpreting it.

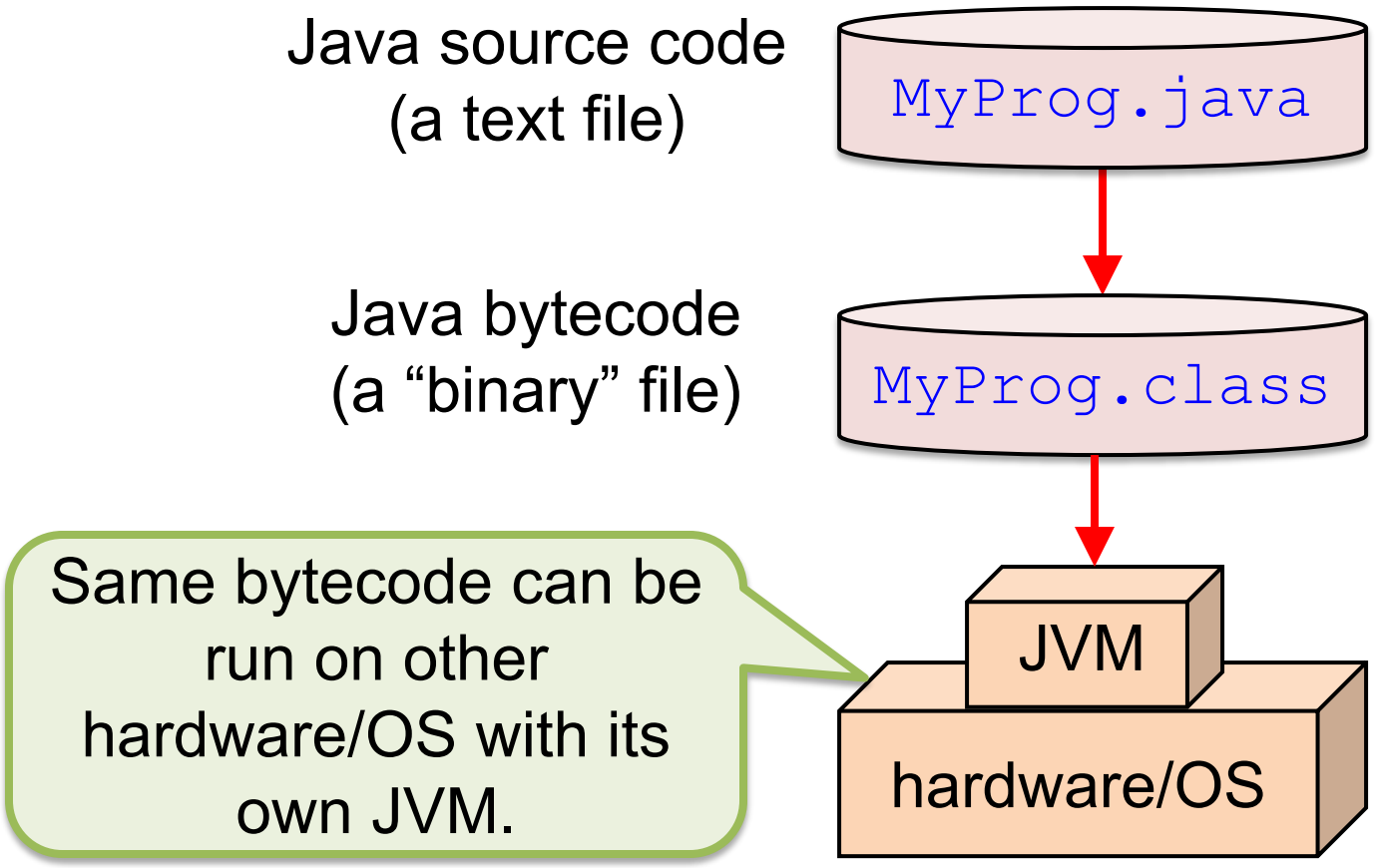
Java End-to-End

Java source code
(a text file)

Java bytecode
(a "binary" file)



Java End-to-End



Significance of JVM

- Portability
  - Java slogan: “Write once, run anywhere”
  - JVM is ubiquitous
- Universality
  - Program source code need not be in Java
- Performance
  - Extra layer comes at (surprisingly small) penalty in performance