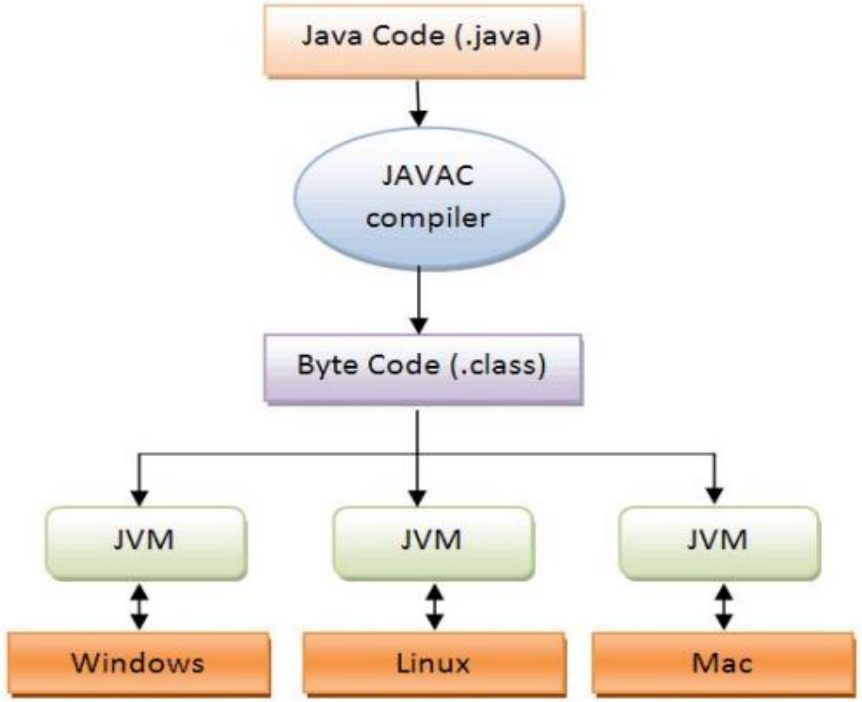
INTRODUCTION TO PROGRAMMING LANGUAGE II (JAVA)

Fariha Zahin
Lecturer
CSE, Southeast University

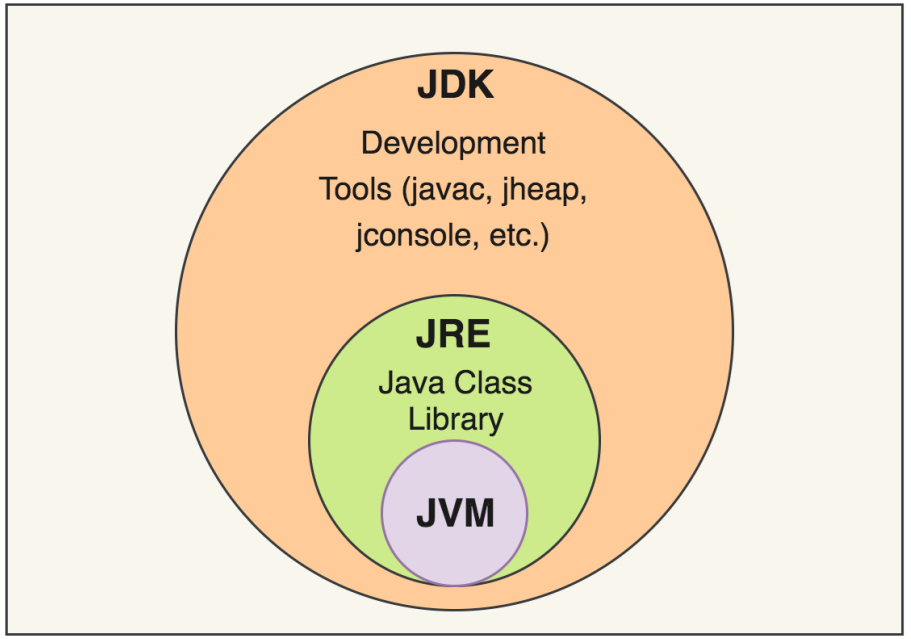
JAVA is a platform independent language:
It uses a virtual machine to execute.

Java Code: A developer writes Java code, which is compiled into bytecode (represented as .class files)
Bytecode: The bytecode is platform-neutral, and it is shown moving to the next stage.
JVM: The bytecode is processed by the Java Virtual Machine (JVM). The JVM is the layer that ensures platform independence. It translates the bytecode into native machine code for each specific platform.
Platforms: the bytecode is translated into platform-specific machine code for different operating systems, like Windows, macOS, and Linux. Each platform would have the appropriate JVM that runs the bytecode correctly.



JVM (Java Virtual Machine)

- Runs Java programs by converting bytecode into machine code.
- It's platform-dependent (Windows, Mac, Linux have different JVMs).
- It ensures Java's **"Write Once, Run Anywhere"** feature.



JRE (Java Runtime Environment)

- Provides everything needed to run Java programs.
- Includes **JVM** + **libraries** but **not** development tools.
- Used for running Java applications only.

JDK (Java Development Kit)

- Used for developing Java applications.
- Includes **JRE** + **JVM** + **development tools (compiler, debugger, etc.)**.
- Required for writing and compiling Java code.

🎬 Scenario: Watching a Movie

Java Term	Movie Analogy	Role
JVM	Actor on stage	Runs the script (bytecode)
JRE	Movie theater	Environment to watch the movie (run Java apps)
JDK	Director's studio	Tools to create and edit the movie (write/compile Java apps)

Class and Object in Java

Class:

- A **class** is a template or blueprint for creating objects.
- It defines **attributes (variables)** and **behaviors (methods)** but does not store actual data.

Object:

- An **object** is an instance of a class.
- It holds real values and performs actions as defined in the class.