Introduction to JAVA

Lecture - 1

Tokey Ahmmed

Lecturer, Dept. of CSE

Varendra University

tokey@vu.edu.bd

1st Textbook:

Java: The Complete Reference

7th Edition or higher

By Herbert Schildt

Content

About Java Technology

Java is both a

- **programming language** and
- **platform**.

Java as **Programming Language**

- The Java programming language is a high-level language that can be characterized by all of the following buzzwords:

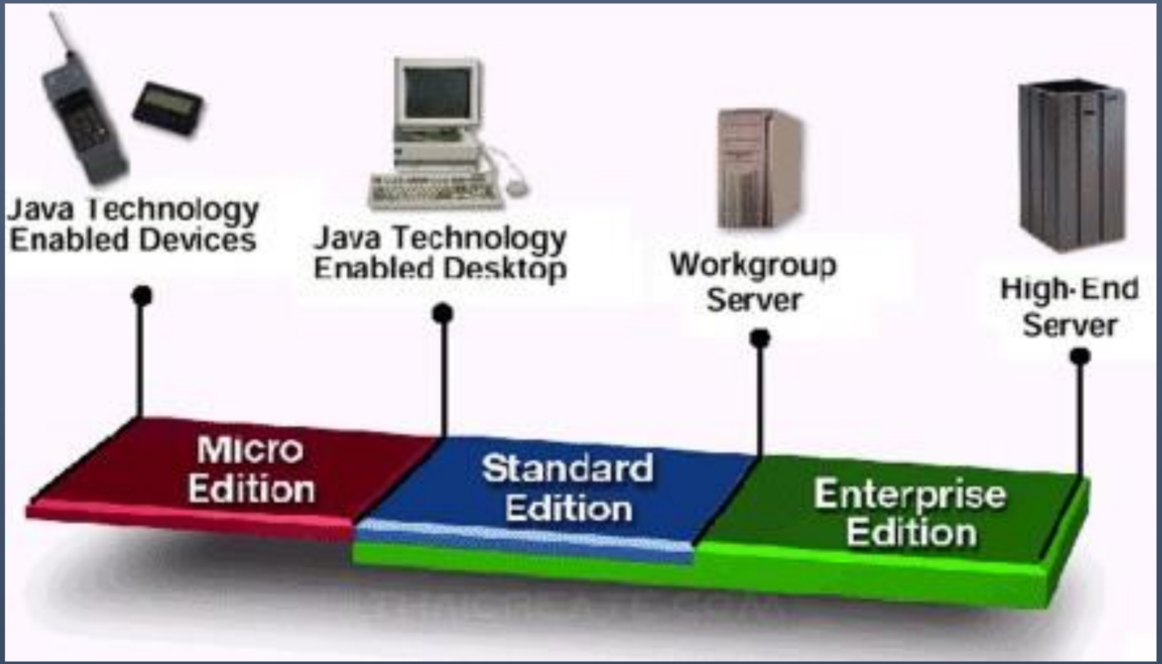
- Simple
- Object oriented
- Distributed
- Multithreaded
- Dynamic
- Architecture neutral
- Portable
- High performance
- Robust
- Secure

* Each of the preceding buzzwords is explained in *The Java Language Environment* , a white paper written by James Gosling and Henry McGilton.

Java as **Platform**

A Platform is the hardware or software environment in which a program runs.

- The Java platform is a software-only platform that runs on top of other hardware-based platforms.



*** As a platform-independent environment, the Java platform can be a **bit slower** than native code. However, advances in compiler and virtual machine technologies are bringing performance close to that of native code without threatening portability.

What Can Java Technology Do (Just Skim)

The general-purpose, high-level Java programming language is a powerful software platform. Every full implementation of the Java platform gives you the following features:

- **Development Tools:** The development tools provide everything you'll need for compiling, running, monitoring, debugging, and documenting your applications. As a new developer, the main tools you'll be using are the javac compiler, the java launcher, and the javadoc documentation tool.
- **Application Programming Interface (API):** The API provides the core functionality of the Java programming language. It offers a wide array of useful classes ready for use in your own applications. It spans everything from basic objects, to networking and security, to XML generation and database access, and more. The core API is very large; to get an overview of what it contains, consult the Java Platform Standard Edition 8 Documentation.
- **Deployment Technologies:** The JDK software provides standard mechanisms such as the Java Web Start software and Java Plug-In software for deploying your applications to end users.
- **User Interface Toolkits:** The JavaFX, Swing, and Java 2D toolkits make it possible to create sophisticated Graphical User Interfaces (GUIs).
- **Integration Libraries:** Integration libraries such as the Java IDL API, JDBC API, Java Naming and Directory Interface (JNDI) API, Java RMI, and Java Remote Method Invocation over Internet Inter-ORB Protocol Technology (Java RMI-IIOP Technology) enable database access and manipulation of remote objects.

Java Editions

- Standard Edition (**Java SE**)
- Enterprise Edition (**Java EE**)
- Micro Edition (**Java ME**)

Java Standard Editions

Till now (18 Jan, 2021) Java has 15 versions of standard editions[1]

Standard Edition (**Java SE**)	Enterprise Edition (**Java EE**)	Micro Edition (**Java ME**)
J2SE 1.2 J2SE 1.3 J2SE 1.4 J2SE 5.0 Java SE 6 Java SE 7 Java SE 8 (LTS) Java SE 9 Java SE 10 Java SE 11 (LTS) Java SE 12 Java SE 13 Java SE 14 Java SE 15	Not Covering	Not Covering

[1] - https://en.wikipedia.org/wiki/Java_version_history

Java Environments

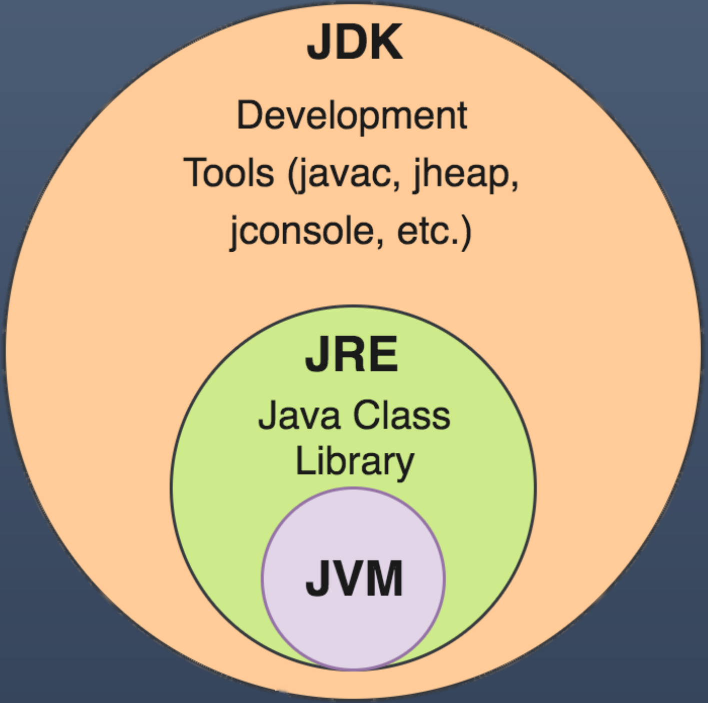
- **(JRE)** for Running Java Application
- **(JDK)** for Java Development

Java Environments

Oracle has 2 products that implement Java SE:

- Java Runtime Environment (**JRE**) and
  - An installation package
  - It physically exists
  - Provides environment to **only run (not develop)**
  - Part of the JDK
  - It contains a set of libraries + other files that JVM uses at runtime

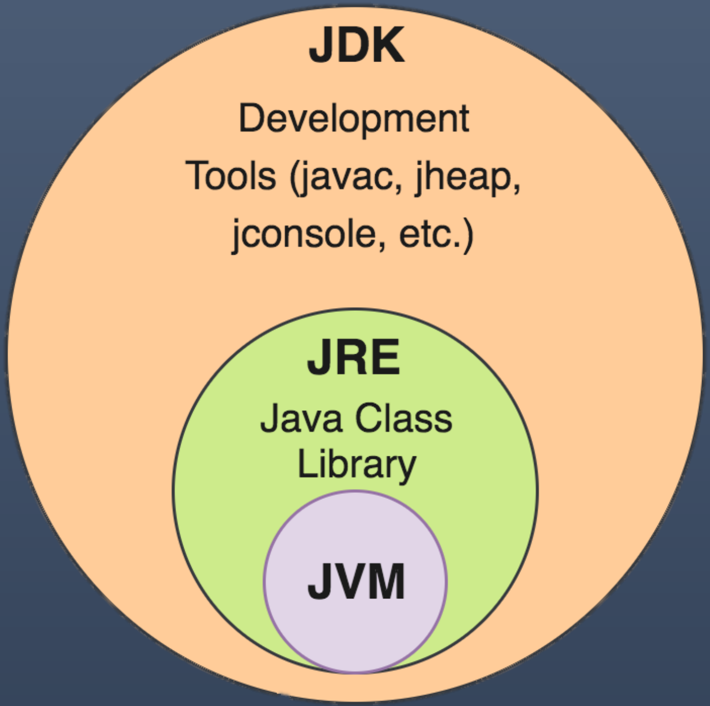
- Java Development Kit (**JDK**)
  - A software package
  - To create Java-based applications
  - Provides the environment to develop and execute(run) the Java program
  - Includes two things
    - Development Tools (such as Java compiler)
    - JRE (Executes java program)



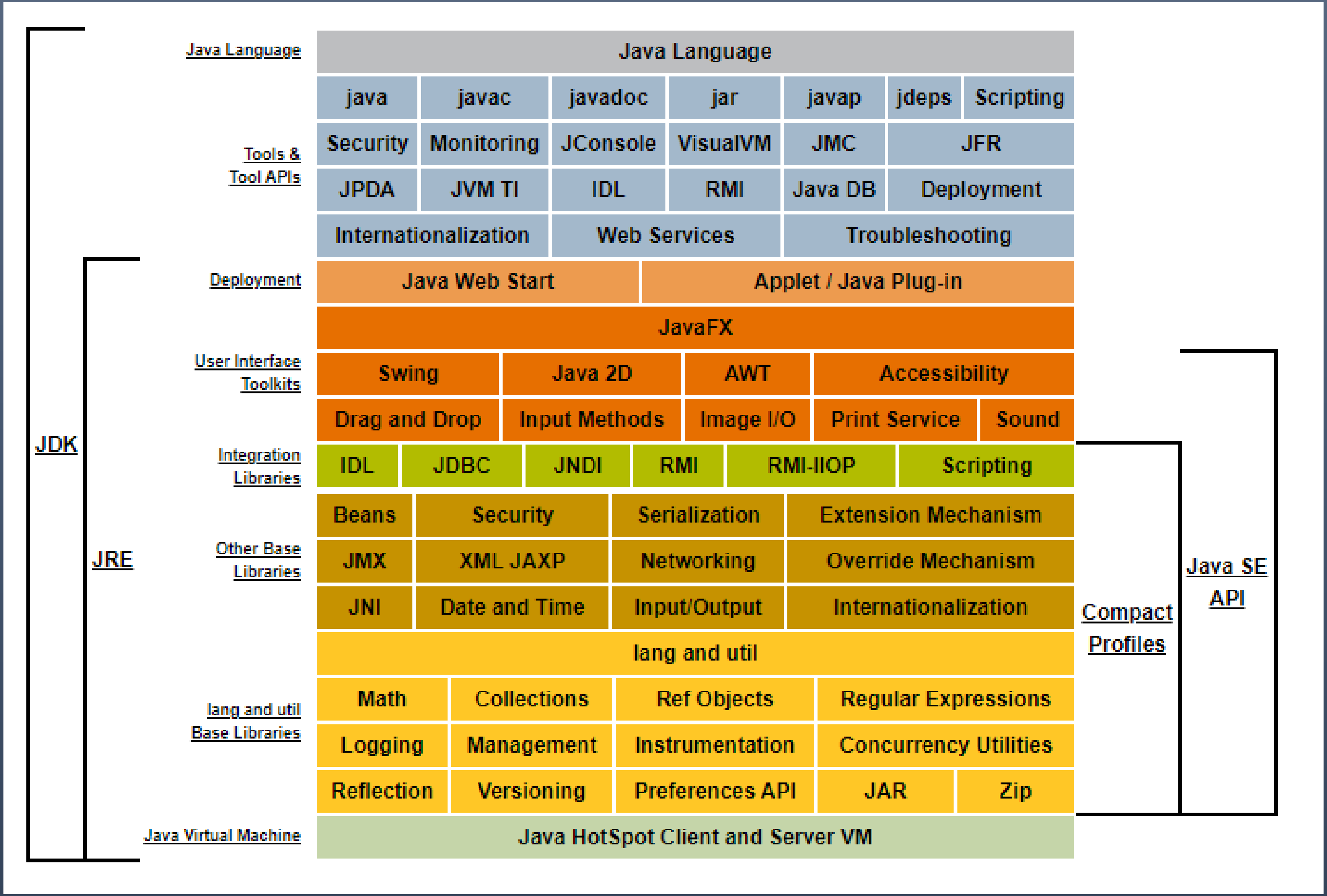
Java Virtual Machine (JVM)

Java Virtual Machine (JVM) is a engine that provides runtime environment to drive the Java Code or applications.

- ❑ Java Virtual Machine (**JVM**)
  - ⤍ An abstract machine
  - ⤍ It is called a virtual machine because it doesn't physically exist
  - ⤍ Part of both JDK and JRE
  - ⤍ JVM is responsible for executing the java program line by line
  - ⤍ It is the heart of the Java technology.



Java Conceptual Diagram (of Java SE 8)

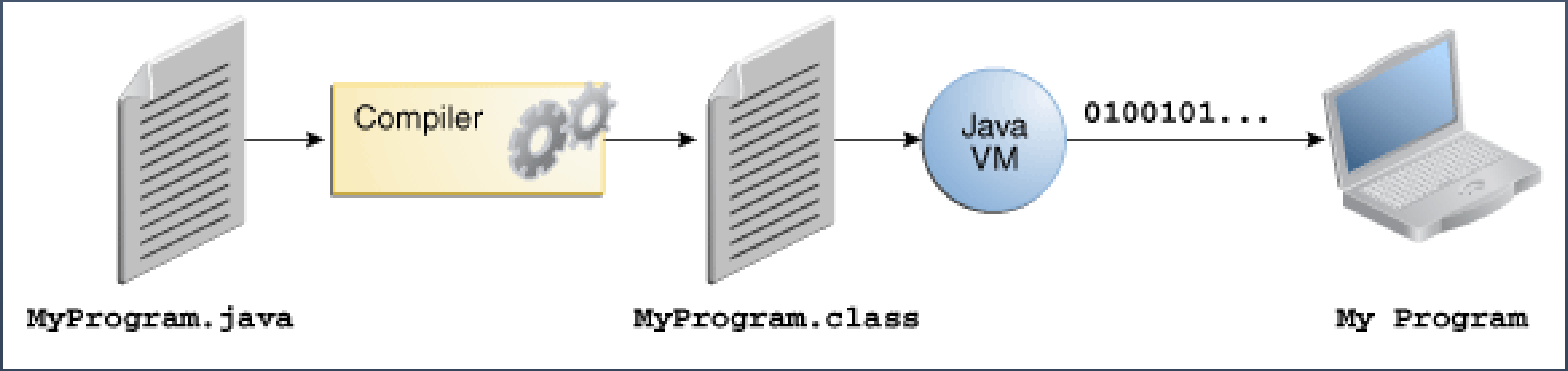


Source: https://docs.oracle.com/javase/8/docs/index.html

Java Compilation Process

Java code **Compilation Process**

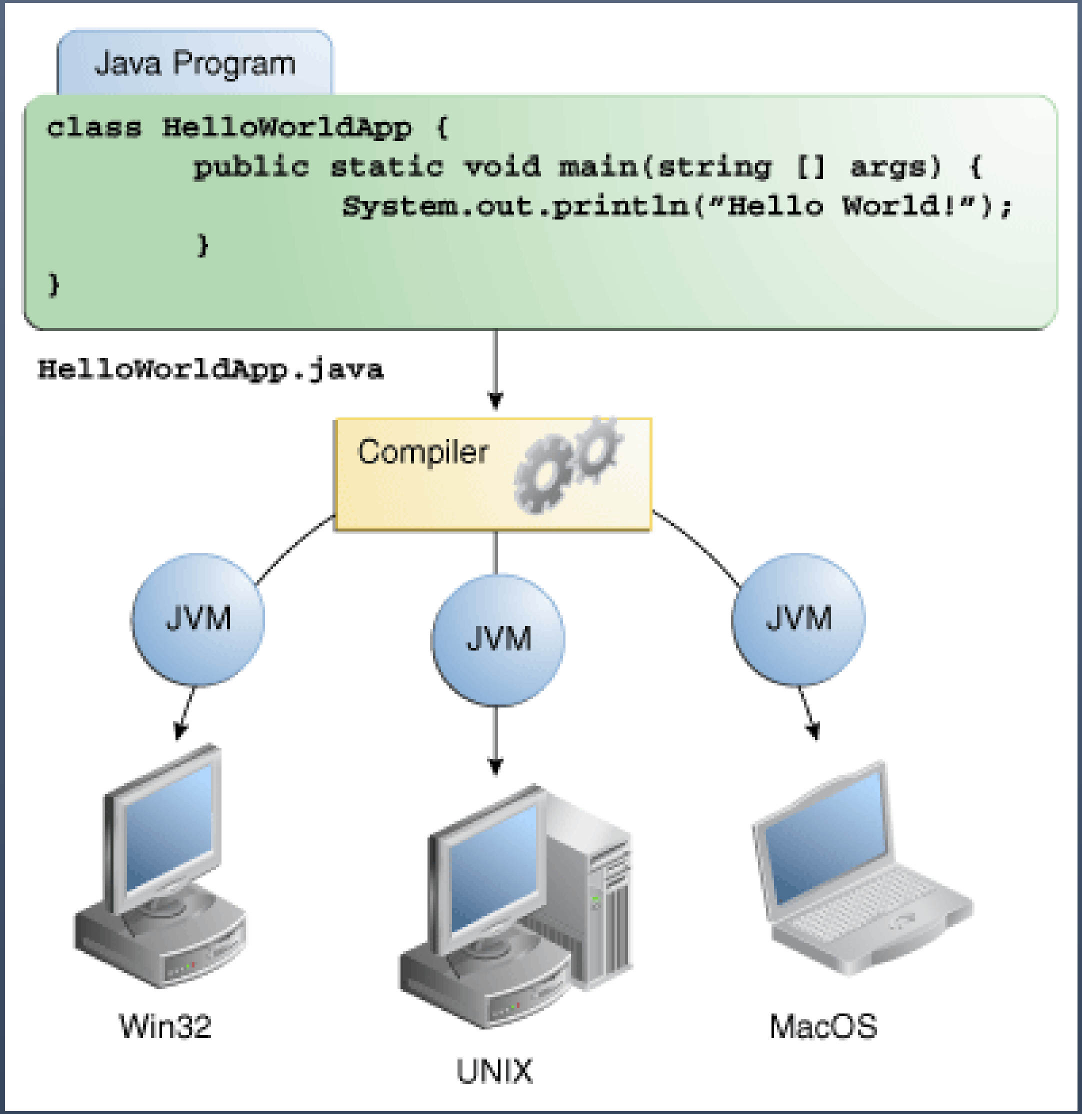
- In the Java programming language, all source code is first written in plain text files ending with the `.java` extension.
- Those source files are then compiled into `.class` files by the `javac` compiler.
- A `.class` file does not contain code that is native to your processor; it instead contains **`bytecodes`** - the machine language of the Java Virtual Machine (JVM).
- The java launcher tool then runs your application with an instance of the Java Virtual Machine.



Java's **Platform Independency**

- Because of **JVM**, Java is supported on many different operating systems.
- So the same `.class` files are capable of running on different operating systems.

  such as:
  - Microsoft Windows,
  - The Solaris™ Operating System (Solaris OS),
  - Linux or
  - Mac OS.



```
class HelloWorldApp {
        public static void main(string [] args) {
                System.out.println("Hello World!");
        }
}
```