Java SE & Spring

Module 1: Java SE

01.Introduction to Java

What is Java Programming Language?

The Java programming language is a high-level language that can be characterized by these keywords:

- Simple
- Object Oriented
- Distributed
- Multithreaded
- Dynamic
- Architecture neutral
- Portable
- High Performance
- Robust
- Secure

Further reading: https://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html

History of Java

- Developed by James Gosling in the early 1990s for project 7.
- First, Gosling and his team intended to use C++ in the project.
- But, they think, a new language is better for it
- So, they developed a new language with name OAK.
- Then renamed it to Java because of copyright reasons.
- Officially released at 23 May 1995 by Sun Microsystems

  that is acquired by Oracle in 2010.

Further reading: https://www.geeksforgeeks.org/the-complete-history-of-java-programming-language/

Why Java 11?

Question: Java has version 16 currently, so why we choose Java 11 for this course?

Answer: The latest Long Term Support(LTS) version is Java 11.

→ Next LTS version is Java 17 and will be released at 17 September 2021

Features of Java

→ <u>Simple & Object Oriented</u>

- Java is simpler than its ancestors like C and C++. It removes unnecessary complexities and low level features of C++ e.g. pointer arithmetic, operator overloading, memory management.
- Java inherits its syntax from C/C++, thus C & C++ developers familiar with it
- The Java programming language is designed to be object oriented from the ground up.
- Programmers using the Java programming language can access existing libraries of tested objects that provide functionality ranging from basic data types through I/O and network interfaces to graphical user interface toolkits. These libraries can be extended to provide new behavior.

Further reading: https://www.oracle.com/java/technologies/introduction-to-java.html

Features of Java

➔ <u>Robust & Secure</u>

- You can develop Java code with confidence that the system will find many errors quickly and that major problems won't lay dormant until after your production code has shipped.
- Java is designed to operate in distributed environments, which means that security is of paramount importance.

Further reading: https://www.oracle.com/java/technologies/introduction-to-java.html

Features of Java

➔ **Architecture Neutral & Portable**

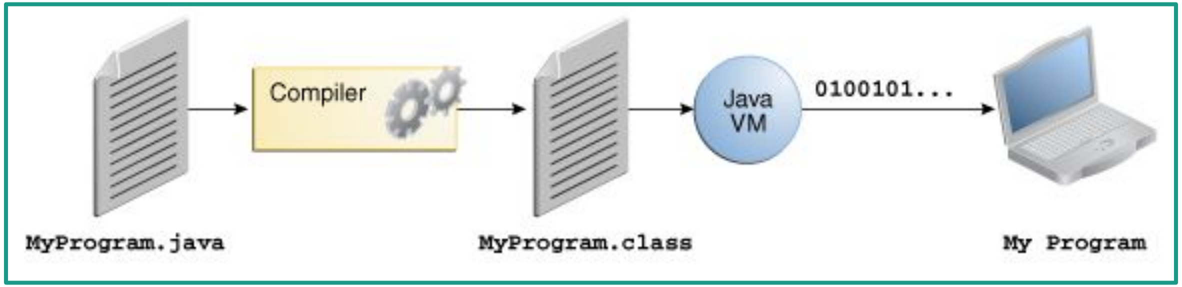
- Your programs are the same on every platform--there are no data type incompatibilities across hardware and software architectures.
- Rather than directly generating(compiling) binary code, Java compiler compiles java code to bytecode. So, same bytecode will run on any platform.
- We will talk about JVM and bytecode a few slides later.

Features of Java

- ➔ Interpreted & Threaded and Dynamic
  - Java is both compiled and interpreted language. At first Java code compiled to bytecode that is JVM’s machine code. Then, JVM interprets the bytecode and runs it on the real machine (hardware)
  - The Java platform supports multithreading at the language level.
  - the language and run-time system are dynamic in their linking stages. Classes are linked only as needed. New code modules can be linked in on demand from a variety of sources, even from sources across a network.

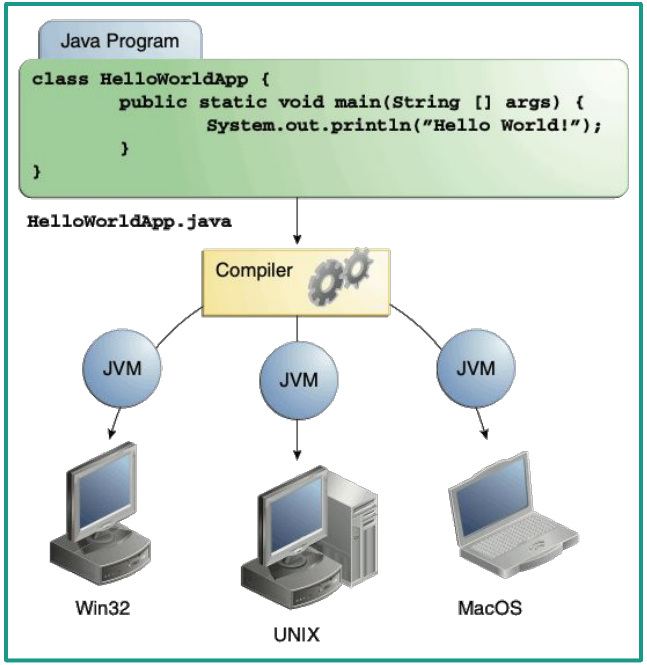
Execution of a Java Program

- 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(machine or binary code) 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.



Execution of a Java Program

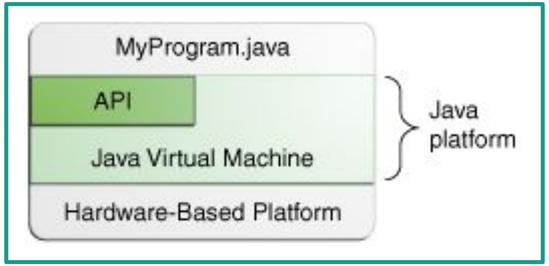
- Because the JVM is available on many different operating systems, the same `.class` files are capable of running on Windows, , Linux, Mac OS, Android etc.
- Through the JVM, the same application is capable of running on multiple platforms.



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

Execution of a Java Program

- <u>The Java Platform</u>
  - A platform is the hardware or software environment in which a program runs.
  - The Java platform has two components:
    - The Java Virtual Machine
    - The Java Application Programming Interface (API)



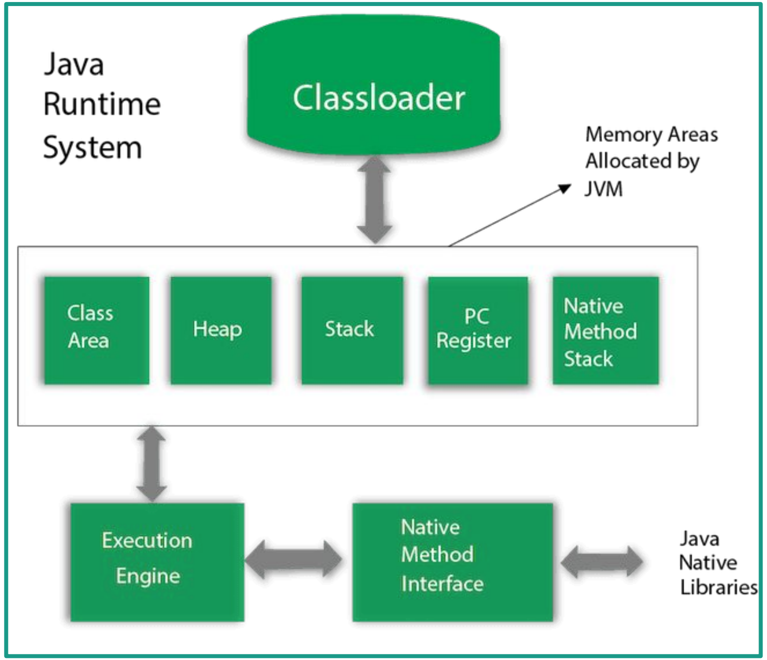
We already talked about JVM. The Java API is a large collection of ready-made software components that provide many useful capabilities. It is grouped into libraries of related classes and interfaces; these libraries are known as packages. As a platform-independent environment, the Java platform can 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.

JVM, JRE and JDK

➔ JVM

- JVM is an engine that provides a runtime environment to drive the Java Code or applications. It converts Java bytecode into machine language. JVM is a part of Java Run Environment (JRE). It cannot be separately downloaded and installed. To install JVM, you need to install JRE.

- In many other programming languages, the compiler produces machine code for a specific system. However, Java compiler produces code for a virtual machine which is called as JVM.



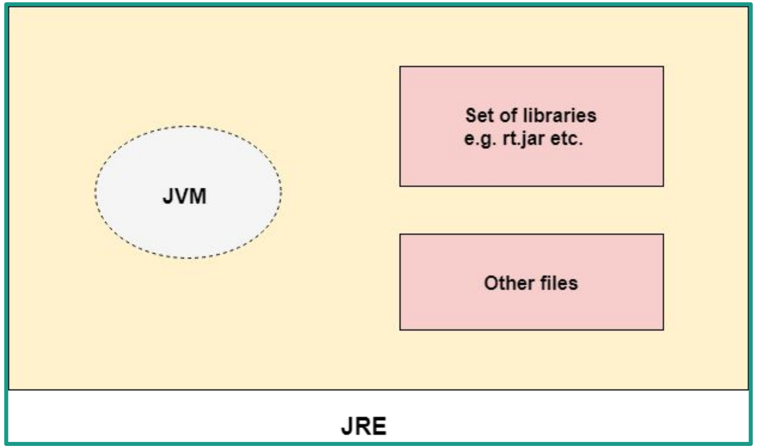
Further reading: https://www.geeksforgeeks.org/differences-jdk-jre-jvm/

JVM, JRE and JDK

➔ JRE

- JRE stands for “Java Runtime Environment” The JRE provides the minimum requirements for executing a Java application; it consists of the Java Virtual Machine (JVM), core classes, and supporting files.

- JRE is an installation package which provides environment to only run(not develop) the java program(or application)onto your machine. JRE is only used by them who only wants to run the Java Programs i.e. end users of your system.



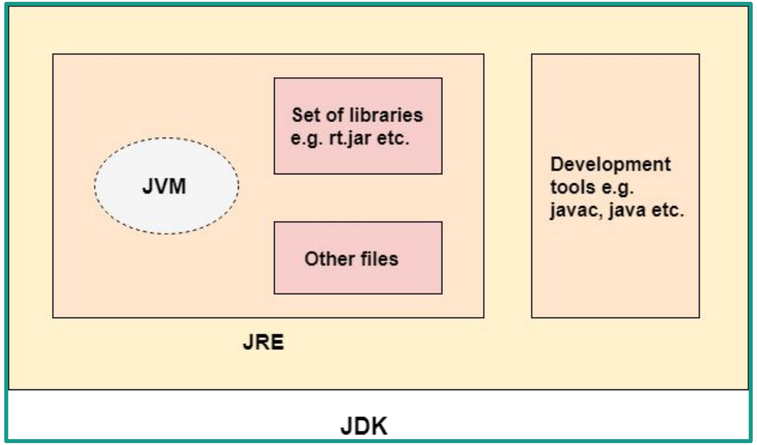
Further reading: https://www.geeksforgeeks.org/differences-jdk-jre-jvm/

JVM, JRE and JDK

➔ JDK

- The Java Development Kit (JDK) is a software development environment used for developing Java applications and applets. It includes the JRE, an interpreter/loader (Java), a compiler (javac), an archiver (jar), a documentation generator (Javadoc) and other tools needed in Java development.

- JDK provides the environment to develop and execute(run) the Java program. Includes two things: Development Tools (to provide an environment to development) and JRE. Note : JDK is only used by Java Developers.


Set of libraries e.g. rt.jar etc.
JVM
Development tools e.g. javac, java etc.
Other files
JRE
JDK

Further reading: https://www.javatpoint.com/jvm-java-virtual-machine

Questions ?

Next: Setting Up Development Environment