INTRODUCTION

- JAVA (Part 1)

(Module 1)-

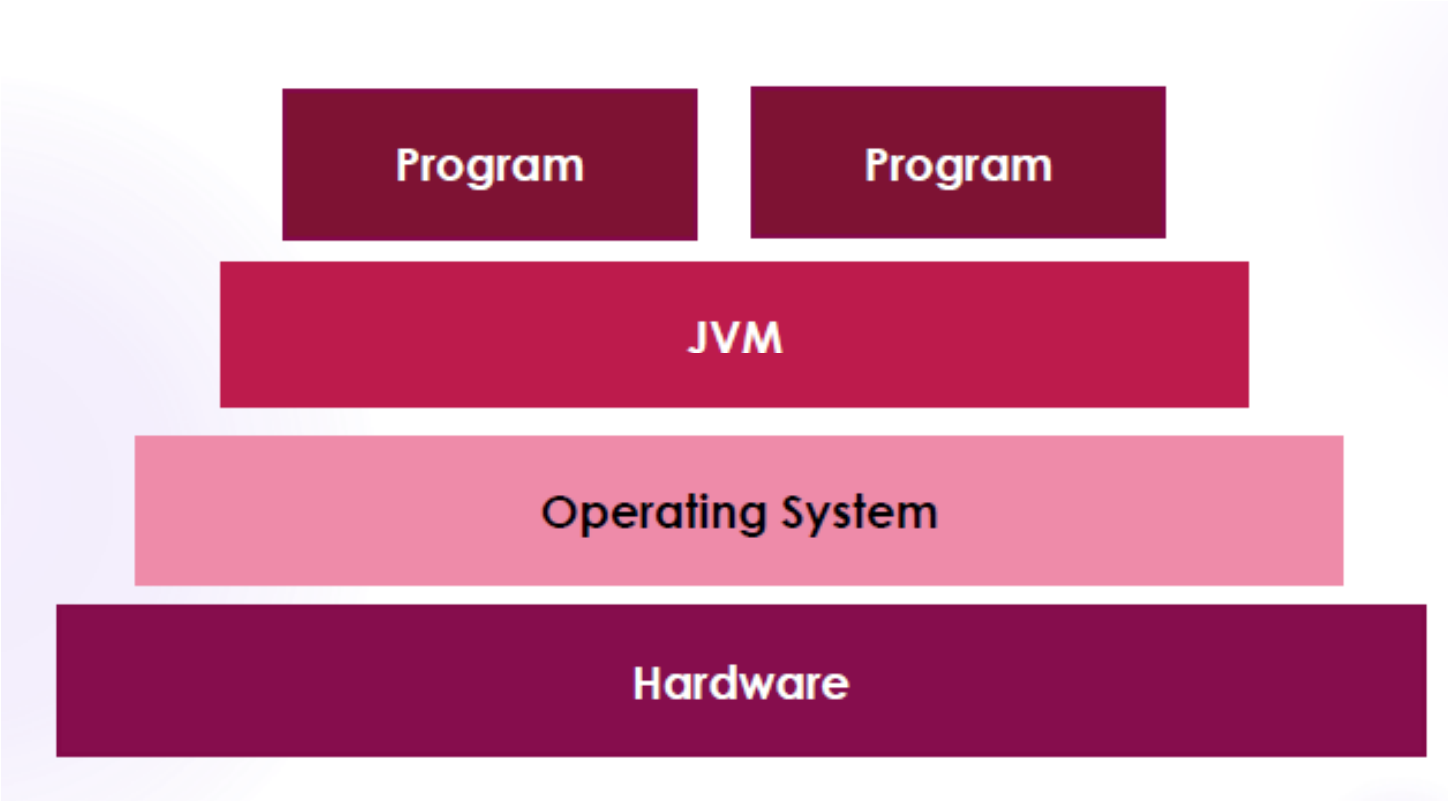
Prepared by Renetha J.B.(LMCST)

Topics

- ✓ Java programming Environment and Runtime Environment,
- ✓ Development Platforms
  - ➢ Standard, Enterprise.
- ✓ JVM
- ✓ Java compiler,
- ✓ Bytecode

Java

- Developed by **James Gosling** from **Sun Microsystems** in 1991.
  - This language was initially called "**Oak**," but was renamed "Java" in 1995.
- The target of Java is to **write a program once and then run this program on multiple operating systems**. (**WORA**)
- Java is a programming language
  - It has compiler, core libraries and a runtime (Java virtual machine(JVM)).

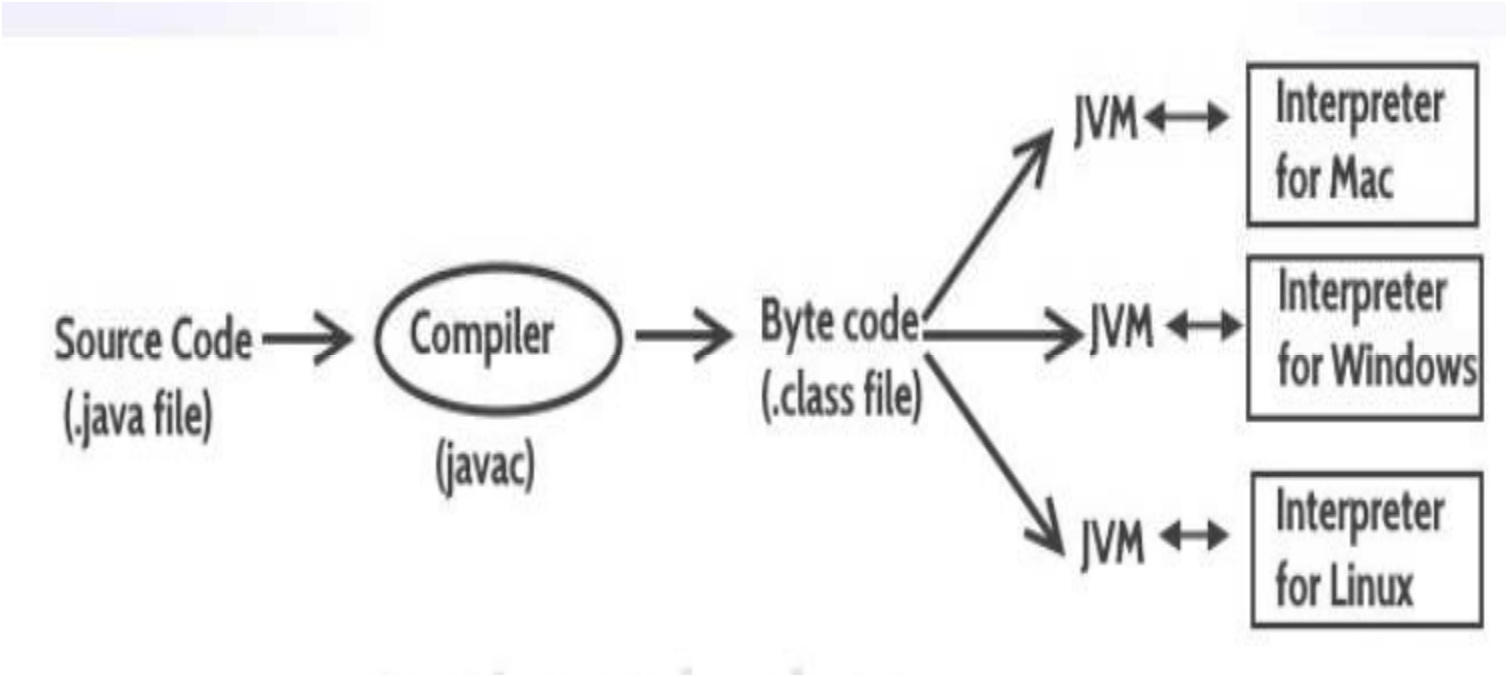

Program
Program
JVM
Operating System
Hardware

JVM

- JVM is the Java run-time system.
- **Java Virtual Machine** is called virtual because it **provides a machine interface** that *does not depend on* the operating system and machine hardware architecture.
- So Java programs are WORA (Write Once Run Anywhere)p rograms.

JVM(contd.)

- When we compile a Java program, we get **.class** file(bytecode) which is not executable.
- JVM **interprets the .class file into machine code** depending on the operating system and hardware.
- JVM **executes java programs like a machine.**
- JVM is also responsible for **garbage collection, array bond checking etc.**
- JVM is platform indepenedent.


Source Code
(.java file)
Compiler
(javac)
Byte code
(.class file)
JVM
Interpreter
for Mac
JVM
Interpreter
for Windows
JVM
Interpreter
for Linux

JRE

- It 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 END-USERS of the system who only wants to run the Java programs

- The JDK, along with the Java Virtual Machine (JVM) and the JRE, can be used by developers to program and run Java applications.

JDK

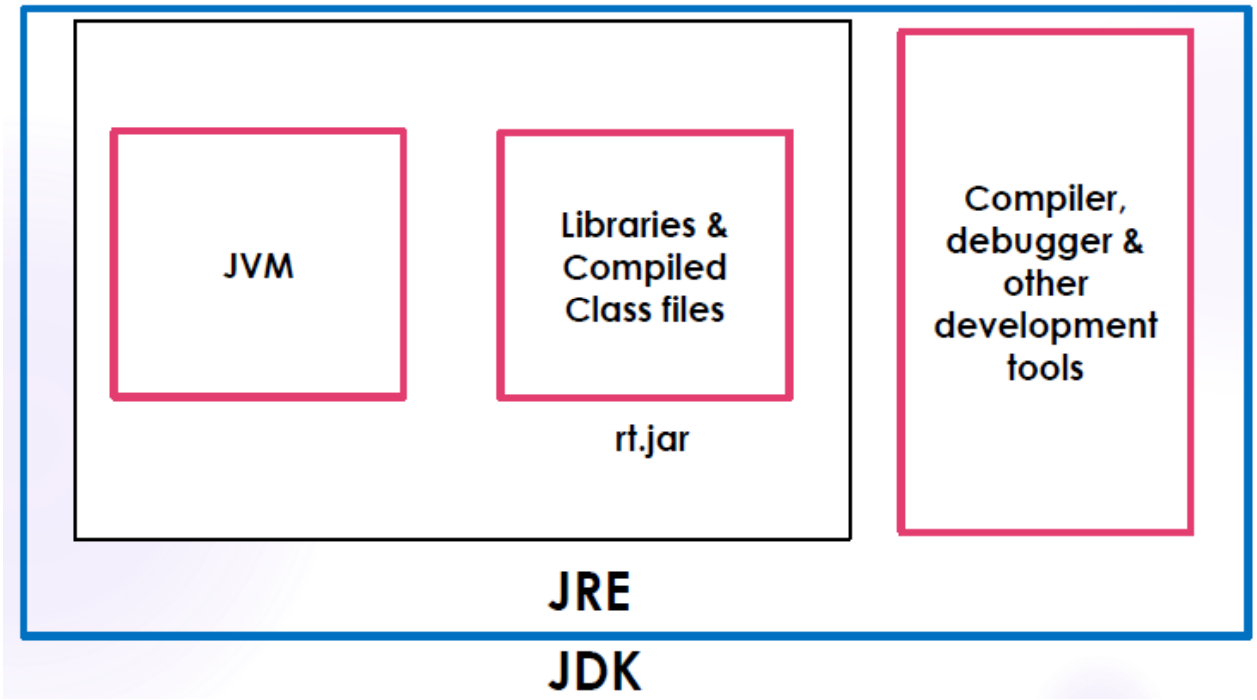
- The Java Development Kit (JDK) is a ***software development environment*** used for developing Java applications.
- **It includes**
  - The **Java Runtime Environment (JRE)**
  - An interpreter/loader (**Java**)
  - A compiler (**javac**)
  - An archive (**jar**)
  - A documentation generator (**Javadoc**)
  - Other tools needed in Java development.

JDK

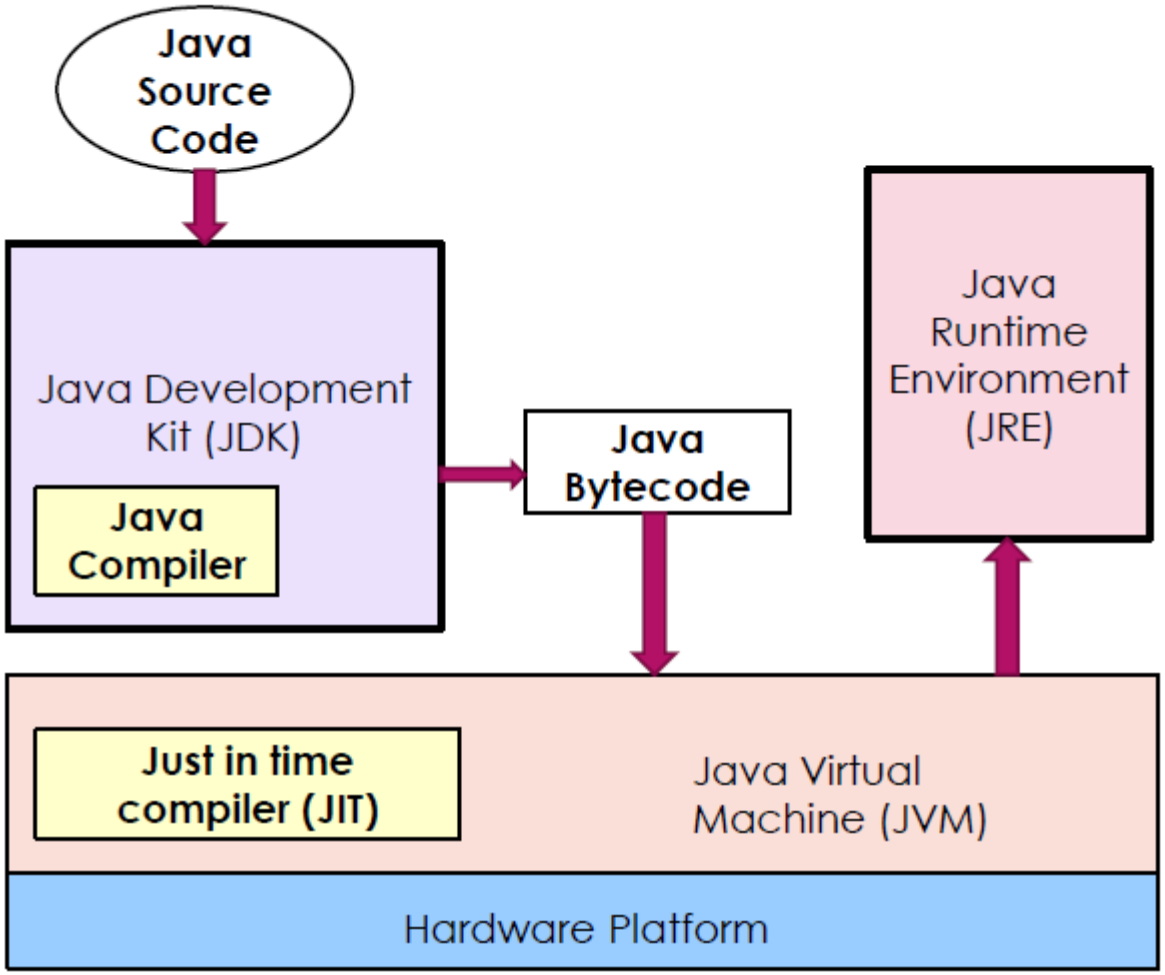
- JDK provides the **environment to develop and execute(run)**the Java program.
- JDK is a kit(or package) which includes two things
  - Development Tools(to provide an environment to develop your java programs)
  - JRE (to execute your java program).

- **JDK is used by Java Developers.**

Jvm jre jdk


JVM
Libraries & Compiled Class files
rt.jar
Compiler, debugger & other development tools
JRE
JDK

Interaction between JDK and JRE



Java Programming Environment

- Java is a **concurrent, class-based, object-oriented programming and runtime environment**, consisting of
  - A programming language
  - An API specification
  - A virtual machine specification

Development Platforms

- All Java platforms consist of a **Java Virtual Machine (JVM)** and an **Application Programming interface (API).**
  - The Java Virtual Machine is a program, for a particular hardware and software platform, that *runs Java technology applications*.
  - An API is a collection of software components that you can *use to create other software components or applications*.

Development Platforms

- Java development platform is a particular ***environment in which Java programming language applications run.***
  - Java Platform, Standard Edition (Java SE)
  - Java Platform, Enterprise Edition (Java EE)
  - Java Platform, Micro Edition (Java ME)
  - Java FX

Development Platforms - Standard Edition

- When most people think of the Java programming language, they think of the Java SE (**Standard Edition**) API.
- Java **SE**'s API **provides the core functionality of the Java** programming language.
- It **defines** everything from the **basic types and objects** of the Java programming language to **high-level classes** for networking, security, database access, graphical user interface (GUI) development, and XML parsing.
- Java SE platform **consists of a virtual machine, development tools, deployment technologies, and other class libraries and toolkits** commonly used in Java technology applications.

Development Platforms –Enterprise Edition

- The Java EE (**Enterprise Edition**) platform is built on top of the Java SE platform.

- The Java EE platform provides an API and runtime environment for developing and running **large-scale, multi-tiered, scalable, reliable, and secure network applications.**

Development Platforms –Micro Edition

- The Java ME platform provides an API and a small-footprint virtual machine for **running Java programming language applications on small devices, like mobile phones.**
- This API is a subset of the Java SE API, along with special class libraries useful for **small device application development**.

Development Platforms –Java FX

- Java FX technology is a platform for **creating rich internet applications written in Java FX Script**.
- Java FX Script is a statically-typed declarative language that is compiled to Java technology bytecode, which can then be run on a Java VM.

Java Compiler

- A Java compiler is a compiler for the Java programming language.
- Java programs are compiled using **javac** command.
- Command for compilation

  javac Programname***.java***
- The output of compiling the java code is not executable code. It is called **bytecode** (Programname***.class***)

Program.java **Program.class**

Java's Magic: The Bytecode

- The output of compiling the java code is not executable code. It is called **bytecode.**
- *Bytecode is* a highly optimized set of instructions designed **to be executed by the Java run-time system**, which is called the *Java Virtual Machine (JVM).*
- JVM was designed as an ***interpreter*** *for bytecode.*
- Bytecode is a class file.

Bytecode(contd.)

- Translating a Java program into bytecode **makes it much easier to run a program in a wide variety of environments** because only the JVM needs to be implemented for each platform. - PORTABILITY
- Although the details of the JVM will differ from platform to platform, all JVM **understand the same Java bytecode.**
- Bytecode has been highly *optimized*, so the use of bytecode enables the JVM to **execute programs much faster**.(eventhough compilation and interpretation is needed)

Bytecode(contd.)

- When a **JIT(Just In Time) compiler** is part of the JVM, *selected portions of bytecode are compiled* into executable code in real time, on a piece-by-piece, demand basis.
- JIT compiler compiles code as it is needed, during execution.
  - Not all sequences of bytecode are compiled—only codes that will benefit from compilation.
  - The remaining code is simply interpreted.
- ***Java is a compiled interpreted language.***

Bytecode(contd.)

- Java bytecode is the **intermediate representation** of your Java program that **contains instructions that Java Virtual Machine will execute**.
- Thus, the output of javac is **not code that can be directly executed.**

REFERENCE

- Herbert Schildt, Java: The Complete Reference, 8/e, Tata McGraw Hill, 2011.