ECAP615

Programming in Java

Harjinder Kaur

Assistant Professor

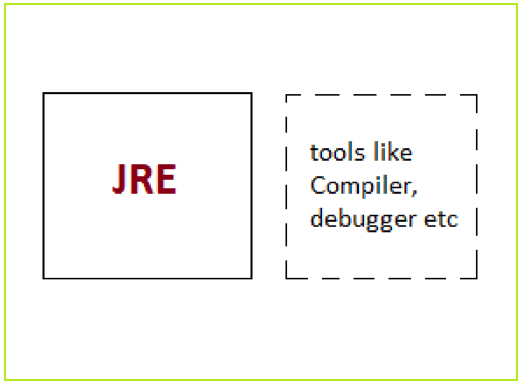
Learning Outcomes

After this lecture, you will be able to

- Learn the basics of JDK,JRE and JVM
- Understand the working of JDK,JRE and JVM

What is JDK

- JDK stand for java development kit .
- It is used to build and develop the java program.
- It internally contains JRE.



JDK - Java Development Kit

- It contains the compiler and debugger.

What is JDK?

- It contains all the related set of libraries and files to build and compile the java program.
- Without JDK we cant build any java program

Basic JDK Tools

The following are the basic tools that becomes the foundation of Java Development Kit.

- Javac
- Java
- Javadoc
- Appletviewer
- Jar

Javac

- Javac is the compiler for the Java programming language.
- It is used to compile .java file.
- It creates a class file which can be run by using java command.

Example: c:javac TestFile.java

Java & Javadoc

- When a class file has been created, the java command can be used to run the Java program.

Example: c:java TestFile.class

- JavaDoc is an API documentation generator for the Java language, which generates documentation in HTML format from Java source code.

Appletviewer & jar

- Appletviewer run and debug applets without a web browser, its standalone command-line program to run Java applets.
- The jar is (manage Java archive) a package file format that contains class, text, images and sound files for a Java application or applet gathered into a single compressed file.

What is JVM

It is:

- A specification where working of Java Virtual Machine is specified. But implementation provider is independent to choose the algorithm. Its implementation has been provided by Oracle and other companies.

What is JVM

- An implementation Its implementation is known as JRE (Java Runtime Environment).
- Runtime Instance Whenever you write java command on the command prompt to run the java class, an instance of JVM is created.

What is JVM

- Mostly in other Programming Languages, compiler produce code for a particular system but Java compiler produce Bytecode for a Java Virtual Machine.
- When we compile a Java program, then bytecode is generated that can be used to run on any platform.
- It is the medium which compiles Java code to bytecode which gets interpreted on a different machine and hence it makes it Platform/Operating system independent.

Platform Independent

- Java is called platform independent because of Java Virtual Machine.
- JVM is the main component of Java architecture.
- As different computers with the different operating system have their JVM, when we submit a .class file to any operating system, JVM interprets the bytecode into machine level language.

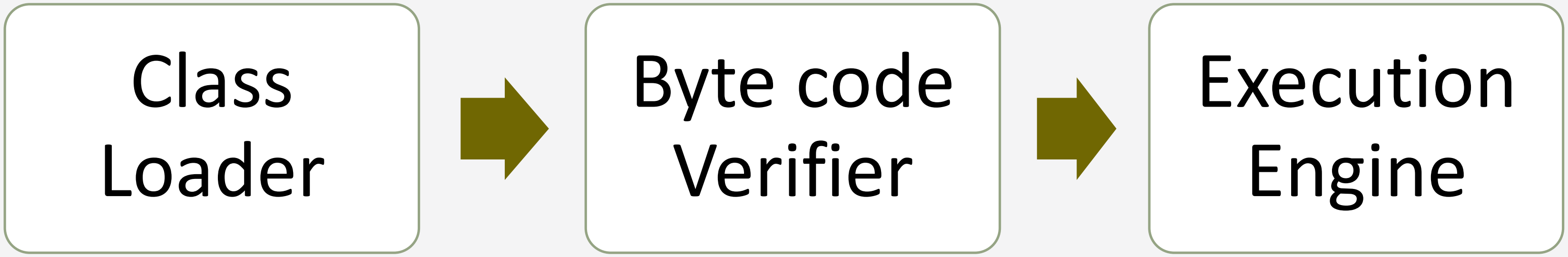
Platform Independent

- A program of JVM is written in C Programming Language, and JVM is Operating System dependent.
- JVM is responsible for allocating the necessary memory needed and also responsible for deallocating the memory space not required by the Java program.

Working of JVM

The JVM performs following operation:

- Loads code
- Verifies code
- Executes code
- Provides runtime environment

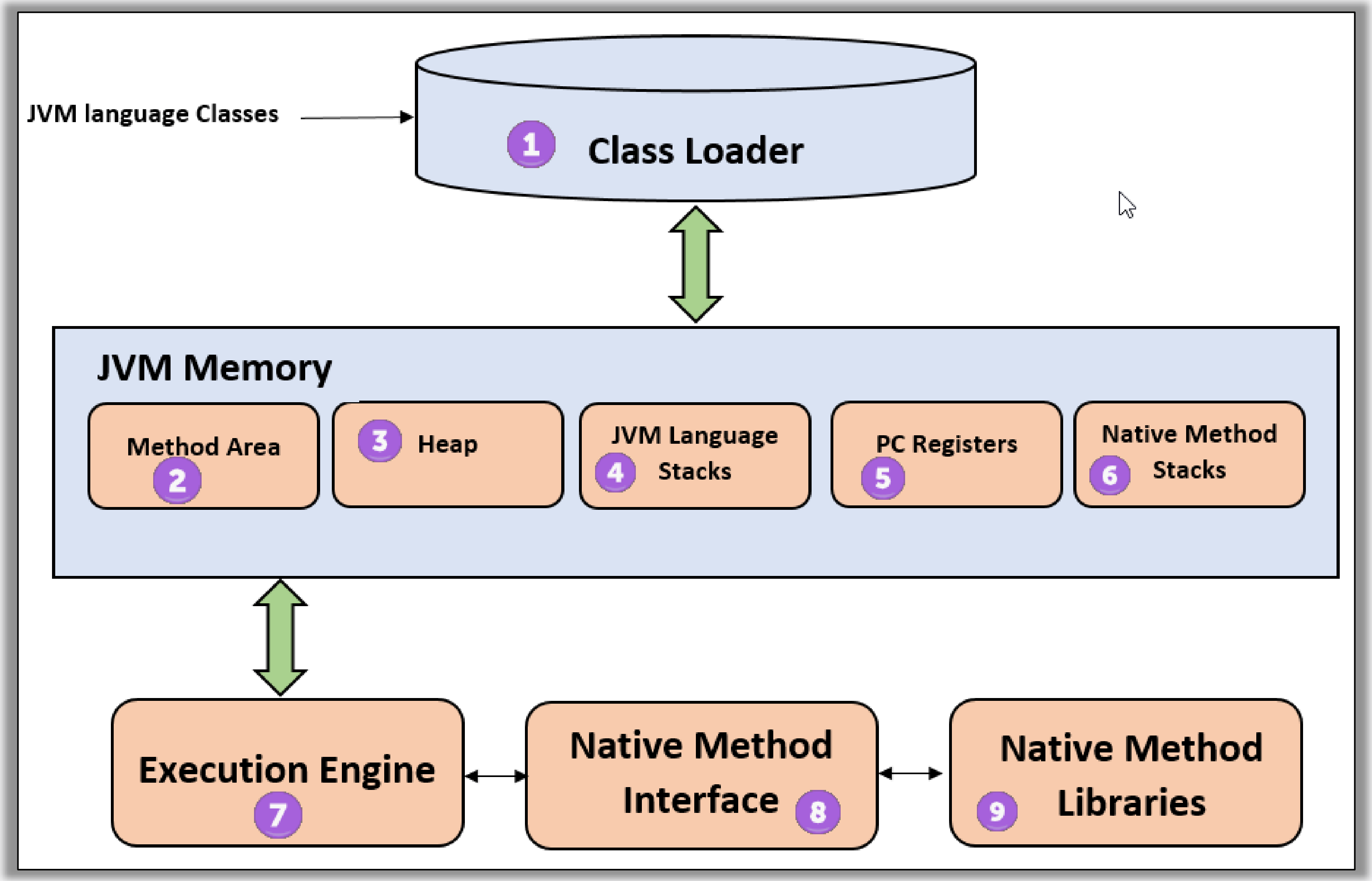


Class Loader

Byte code Verifier

Execution Engine

JVM Architecture



JVM Architecture

1) Class loader

The class loader is a subsystem used for loading class files. It performs three major functions viz. Loading, Linking, and Initialization.

2) Method Area

JVM Method Area stores class structures like metadata, the constant runtime pool, and the code for methods.

JVM Architecture

3) Heap

- All the Objects, their related instance variables, and arrays are stored in the heap.
- This memory is common and shared across multiple threads.

JVM Architecture

4) JVM language Stacks

- Java language Stacks store local variables, and it’s partial results.
- Each thread has its own JVM stack, created simultaneously as the thread is created.
- A new frame is created whenever a method is invoked, and it is deleted when method invocation process is complete.

JVM Architecture

5) PC Registers

- PC register store the address of the Java virtual machine instruction which is currently executing.
- In Java, each thread has its separate PC register.

6) Native Method Stacks

- Native method stacks hold the instruction of native code depends on the native library.
- It is written in another language instead of Java.

JVM Architecture

7) Execution Engine

- It is a type of software used to test hardware, software, or complete systems.
- The test execution engine never carries any information about the tested product.

JVM Architecture

8) Native Method interface

- The Native Method Interface is a programming framework that Java code which is running in a JVM to call by libraries and native applications.

9) Native Method Libraries

- Native Libraries is a collection of the Native Libraries(C, C++) which are needed by the Execution Engine.

What is JRE?

- A Java runtime environment (JRE) is a set of components to create and run a Java application.
- A JRE is part of a Java development kit (JDK).
- A JRE is made up of :

✓Java virtual machine (JVM)

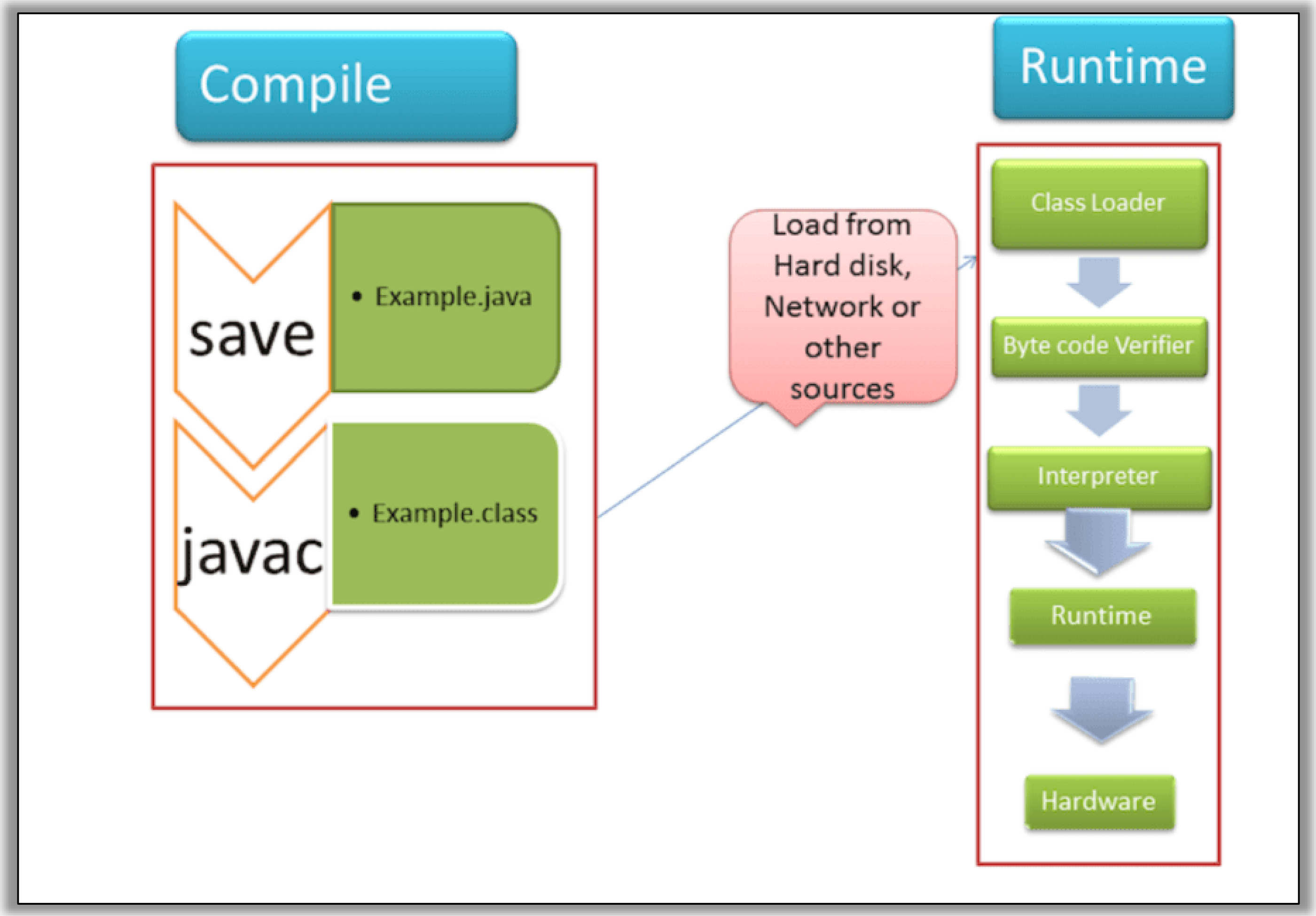
✓ Java class libraries

✓Java class loader.

Working of JRE

- Once we write any source code, we have to save it with the .java extension.
- Next, we compile our program using the “javac” command in command prompt.
- It translates the code to the bytecode which is platform-independent.
- When we compile our program, it will generate a .class which has the bytecode for the equivalent java file. And that bytecode is executed on any platform having the JRE.

Working of JRE



Working of JRE

The workflow is explained below.

ClassLoader

It loads the various classes dynamically in the JVM (Java Virtual Machine), which are essential for executing the program. After JVM started following three class loader are used:

- Bootstrap class loader
- Extension class loader
- System class loader

Working of JRE

Byte code verifier

- It verifies the byte code during the run time so that the code doesn’t make any disturbance for the interpreter.
- The codes are interpreted, only if it passes the tests of the bytecode verifier that check the formatting and for illegal code.

Working of JRE

Interpreter

- When class loaded and code gets verified, then it reads the assembly code line by line and process the following two functions:
- It executes the Byte Code.
- Make appropriate calls to the integrated hardware.

Difference Between JDK,JRE and JVM

JDK	JRE	JVM
The full form of JDK is Java Development Kit.	The full form of JRE is Java Runtime Environment.	The full form of JVM is Java Virtual Machine.
JDK is platform dependent.	JRE is also platform dependent.	JVM is platform-independent.
JDK is a software development kit to develop applications in Java.	It is a software bundle which provides Java class libraries with necessary components to run Java code.	JVM executes Java byte code and provides an environment for executing it.
It is the superset of JRE	It is the subset of JDK.	JVM is a subset of JRE.
JDK comes with the installer.	JRE only contain environment to execute source code.	JVM bundled in both software JDK and JRE.

Difference Between JDK,JRE and JVM

JDK	JRE	JVM
The JDK enables developers to create Java programs that can be executed and run by the JRE and JVM.	The JRE is the part of Java that creates the JVM.	It is the Java platform component that executes source code.
It contains tools for developing, debugging, and monitoring java code.	It contains class libraries and other supporting files that JVM requires to execute the program.	Software development tools are not included in JVM.

That’s all for now…