

Class Loader:

The class loader subsystem is used for loading/reading the **.class** files and saving the bytecode in the JVM method area. It performs three basic activities in following order

- Loading (imports the data)
- Linking (performs verification)
- Initialization (Invokes class variables)

Class / Method Area:

This component holds the class level data of each **.class** file such as metadata, static variables, the code for the methods etc.

Heap Area:

This component is a part of JVM memory where all the objects and its corresponding instance variables and arrays are stored.

Stack Area:

While running a method, it needs some more memory to store the data and results. This memory is allotted on Java Stacks. [This area plays an important role during the method invocation and returns.]

PC Registers:

This component holds the address of the JVM instruction which is currently executing. Each thread in Java has its own PC register to hold the address of the currently executing instruction

Native Method Stacks:

This component is written in a different language and holds the native method information.

Execution Engine:

This component executes the bytecode which is assigned to the runtime data areas and has two major sub-components i.e.:

- **Interpreter:** This component reads the bytecode instructions and executes them in a sequential manner.
- **Garbage Collection:** This component is a part of execution engine which frees up the memory by collecting and removing the unreferenced objects

Native Method Interface:

This allows the Java code to call or be called by the libraries and the native applications (i.e. the programs specific to the hardware and the OS of a system).

Native Method Libraries:

This component is a collection of native C, C++ libraries which are required by the execution engine.

JVM vs. JRE vs. JDK

Java Virtual Machine (JVM):

JVM is a virtual machine which provides a runtime environment for executing the Java bytecode

Java Runtime Environment (JRE):

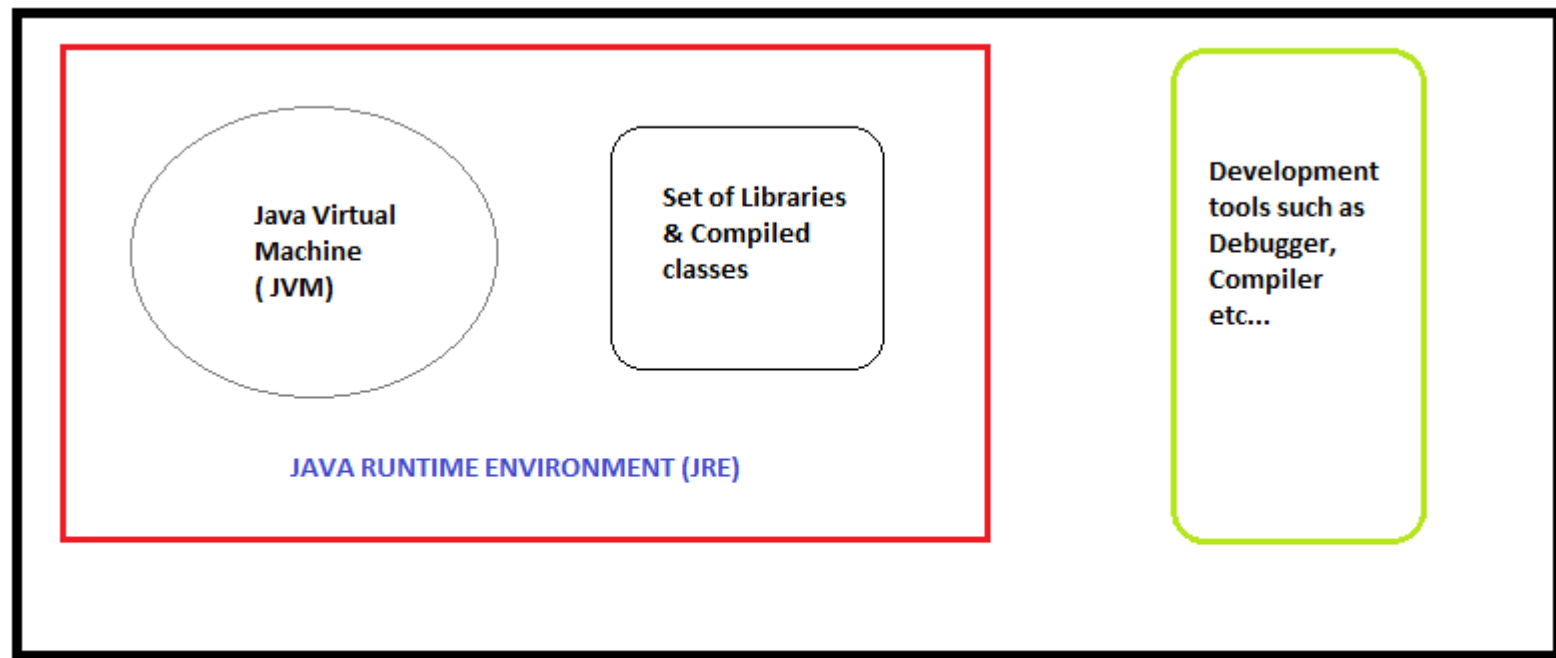
JRE is an environment within which the JVM runs and has class libraries and other files that Java Virtual Machine uses at the time of execution.

JRE = Java Virtual Machine (JVM) + Libraries to run the application

Java Development Kit (JDK):

JDK is the parent set of the JRE and has everything that JRE contains along with the development tools such as a compiler, debugger etc.

JDK = Java Runtime Environment (JRE) + Development tools



JAVA DEVELOPMENT KIT (JDK)

The Java Buzzwords

- The key considerations were summed up by the Java team in the following list of buzzwords:
 - ❖ Simple
 - ❖ Object-oriented
 - ❖ Robust
 - ❖ Multithreaded
 - ❖ Architecture-neutral
 - ❖ Interpreted & High performance
 - ❖ Distributed
 - ❖ Dynamic
 - ❖ Secure
 - ❖ Portability

CLASS

- Class is blue print or an idea of an Object
- From One class any number of Instances can be created
- It is an encapsulation of attributes and methods

