

# JDK vs JRE vs JVM: Key Differences

By James Hartman ⌚ Updated October 6, 2021

## What is JDK?

JDK is a software development environment used for making applets and Java applications. The full form of JDK is Java Development Kit. Java developers can use it on Windows, macOS, Solaris, and Linux. JDK helps them to code and run Java programs. It is possible to install more than one JDK version on the same computer.

In this tutorial, you will learn:

- [What is JDK?](#)
- [What is JRE?](#)
- [What is JVM?](#)
- [Why use JDK?](#)
- [Why use JRE?](#)
- [Why JVM?](#)
- [Features of JDK](#)
- [Features of JRE](#)
- [Features of JVM](#)
- [How JDK Functions?](#)
- [How JRE Functions?](#)
- [How JVM Functions?](#)
- [Difference between JDK, JRE and JVM](#)

## What is JRE?

JRE is a piece of a software which is designed to run other software. It contains the class libraries, loader class, and JVM. In simple terms, if you want to run Java program you need JRE. If you are not a programmer, you don't need to install JDK, but just JRE to run

Java programs. Though, all JDK versions comes bundled with Java Runtime Environment, so you do not need to download and install the JRE separately in your PC. The full form of JRE is Java Runtime Environment.

## What is 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. The full form of JVM is Java Virtual Machine.

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.

## KEY DIFFERENCES

- JDK is a software development kit whereas JRE is a software bundle that allows Java program to run, whereas JVM is an environment for executing bytecode.
- The full form of JDK is Java Development Kit, while the full form of JRE is Java Runtime Environment, while the full form of JVM is Java Virtual Machine.
- JDK is platform dependent, JRE is also platform dependent, but JVM is not platform independent.
- JDK contains tools for developing, debugging, etc. JRE contains class libraries and other supporting files, whereas software development tools are not included in JVM.
- JDK comes with the installer, on the other hand, JRE only contains the environment to execute source code whereas JVM bundled in both software JDK and JRE.

## Why use JDK?

Here are the important reasons of using JDK:

- JDK contains tools required to write Java programs, and JRE to execute them.
- It includes a compiler, Java application launcher, Appletviewer, etc.
- Compiler converts code written in Java into byte code.
- Java application launcher opens a JRE, loads the necessary class, and executes its main method.

## Why use JRE?

Here are the important reasons of using JRE:

- JRE contains class libraries, JVM, and other supporting files. It does not contain any tool for Java development like a debugger, compiler, etc.
- It uses important package classes like math, swingetc, util, lang, awt, and runtime libraries.
- If you have to run Java applets, then JRE must be installed in your system.

## Why JVM?

Here are the important reasons of using JVM:

- JVM provides a platform-independent way of executing Java source code.
- It has numerous libraries, tools, and frameworks.
- Once you run Java program, you can run on any platform and save lots of time.
- JVM comes with JIT(Just-in-Time) compiler that converts Java source code into low-level machine language. Hence, it runs more faster as a regular application.

## Features of JDK

Here are the important features of JDK:

- It enables you to handle multiple extensions in a single catch block.
- JDK includes all features that JRE has.
- It contains development tools such as a compiler, debugger, etc.
- JDK provides the environment to develop and execute Java source code.
- It can be installed on Windows, Unix, and Mac operating systems.
- Diamond operator can be used in specifying a generic type interface instead of

writing the exact one.

## Features of JRE

Here are the important features of JRE:

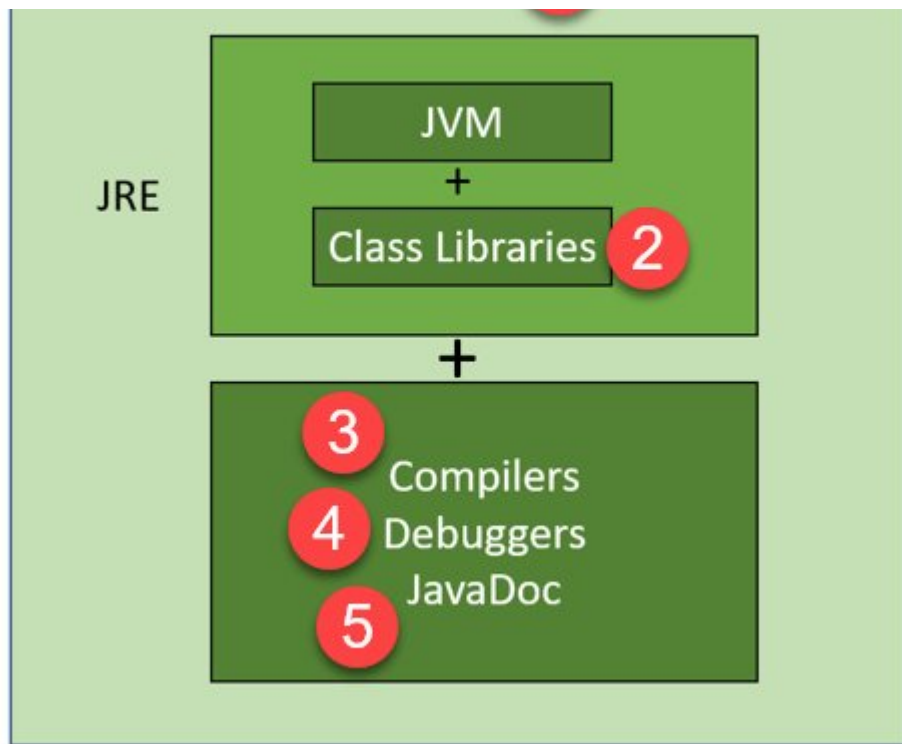
- Java Runtime Environment is a set of tools using which the JVM actually runs.
- JRE contains deployment technology, including Java Web Start and Java Plug-in.
- Developers can easily run the source code in JRE, but he/she cannot write and compile the Java program.
- It includes integration libraries like Java Database Connectivity (JDBC), Remote Method Invocation (RMI), Java Naming and Directory Interface (JNDI), and more.
- JRE has JVM and Java HotSpot virtual machine client.

## Features of JVM

Here are the important features of JVM:

- It enables you to run applications in a cloud environment or in your device.
- Java Virtual Machine converts byte code to the machine-specific code.
- It provides basic java functions like memory management, security, garbage collection, and more.
- JVM runs the program by using libraries and files given by Java Runtime Environment.
- JDK and JRE both contain Java Virtual Machine.
- It can execute the java program line by line hence it is also called as interpreter.
- JVM is easily customizable for example, you can allocate minimum and maximum memory to it.
- It is independent from hardware and the operating system. So, you can write a java program once and run anywhere.

## How JDK Functions?

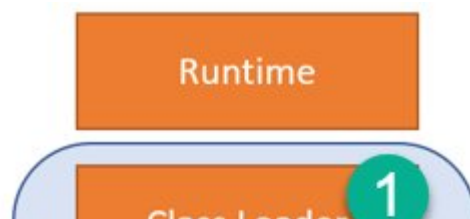


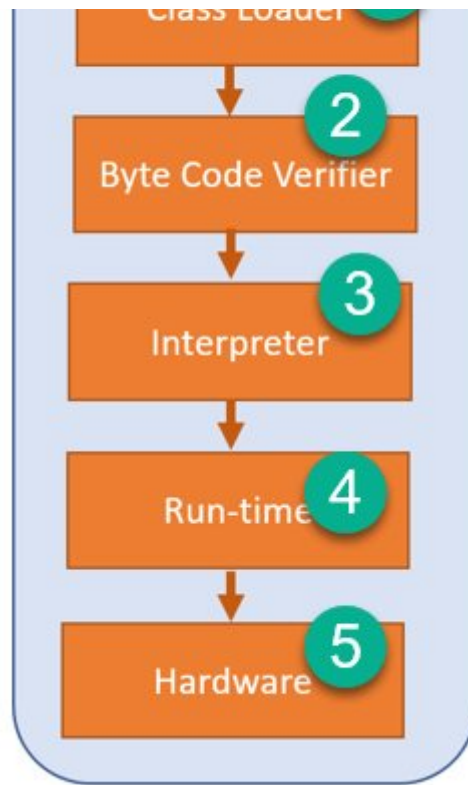
JDK Functionality

Here are the important components of JDK:

- **JDK and JRE:** The JDK enables programmers to create core Java programs that can be run by the JRE, which included JVM and class libraries.
- **Class Libraries:** It is a group of dynamically loadable libraries that Java program can call at run time.
- **Compilers:** It is a Java program that accepts text file of developers and compiles into Java class file. It is the common form of output given by compiler, which contains Java byte code. In Java, the primary compiler is Javac.
- **Debuggers:** Debugger is a Java program that lets developers test and debug Java programs.
- **JavaDoc:** JavaDoc is documentation made by Sun Microsystems for the Java. JavaDoc can be used generating API documentation in HTML file from the source program

## How JRE Functions?





JRE Functionality

JRE has an instance of JVM with it, library classes, and development tools. Once you write and compile Java code, the compiler generates a class file having byte code.

Here are the important components of JRE:

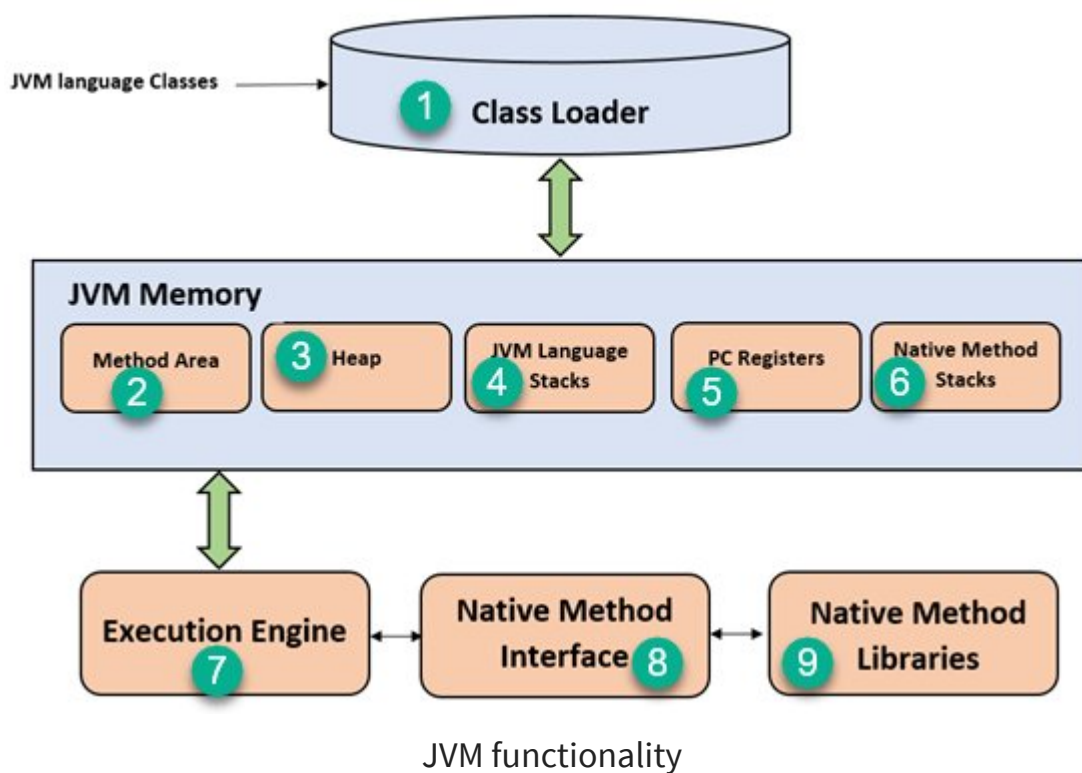
- **Class loaders:** The class loader loads various classes that are necessary for running a Java program. JVM uses three class loaders called the bootstrap class loader, extensions class loader, and system class loader.
- **Byte code verifier:** Byte code verifier verifies the bytecode so that the code doesn't

disturb the interpreter.

- **Interpreter:** Once the classes get loaded, and the code is verified, the interpreter reads the code line by line.
- **Run-time:** Run-time is a system used mainly in programming to describe time period during which a particular program is running.
- **Hardware:** Once you compile Java native code, it runs on a specific hardware platform.

In this way, the Java program runs in JRE.

## How JVM Functions?



Here are the important components of JVM:

### 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 structure of class like metadata, the code for Java methods, and the constant runtime pool.

### **3) Heap**

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

### **4) JVM language Stacks**

Java language Stacks store local variables, and its partial results. Each and every thread has its own JVM language stack, created concurrently as the thread is created. A new frame is created when method is invoked, and it is removed when method invocation process is complete.

### **5) PC Registers**

PC registers 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 allocates memory on native heaps or uses any type of stack.

### **7) Execution Engine**

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

### **8) Native Method interface**

The Native Method Interface is a programming framework. It allows 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.

## **Difference between JDK, JRE and JVM**

Here are the major differences between JDK vs. JRE vs. 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 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.
JDK is platform dependent.	JRE is also platform dependent.	JVM is highly platform dependent.
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.
It is the superset of JRE	It is the subset of JDK.	JVM is a subset of JRE.
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.
JDK comes with the installer.	JRE only contain environment to execute source code.	JVM bundled in both software JDK and JRE.

## You Might Like:

- [SQL Vs PL/SQL Vs T-SQL: Key Differences](#)
- [32 bit vs 64 bit: Key Differences](#)
- [SAT vs ACT: Key Differences | Which one should you take?](#)
- [Telnet vs SSH: Key Differences](#)
- [Wix vs. Squarespace: Must know Differences \(2021\)](#)



## About

[About Us](#)[Advertise with Us](#)[Write For Us](#)[Contact Us](#)

## Career Suggestion

[SAP Career Suggestion Tool](#)[Software Testing as a Career](#)

## Interesting

[eBook](#)[Blog](#)[Quiz](#)[SAP eBook](#)

## Execute online

[Execute Java Online](#)[Execute Javascript](#)[Execute HTML](#)[Execute Python](#)