



# SNS COLLEGE OF TECHNOLOGY

(An Autonomous Institution)

Approved by AICTE, New Delhi, Affiliated to Anna University, Chennai

Accredited by NAAC-UGC with 'A++' Grade (Cycle III) &

Accredited by NBA (B.E - CSE, EEE, ECE, Mech & B.Tech.IT)

COIMBATORE-641 035, TAMIL NADU



## UNIT I - Introduction To OOP

Object Oriented Programming concepts – Evolution of java – **Java Architecture** – Data Types – Variables and Operations – Environment setup – Command Line Arguments - Comments.

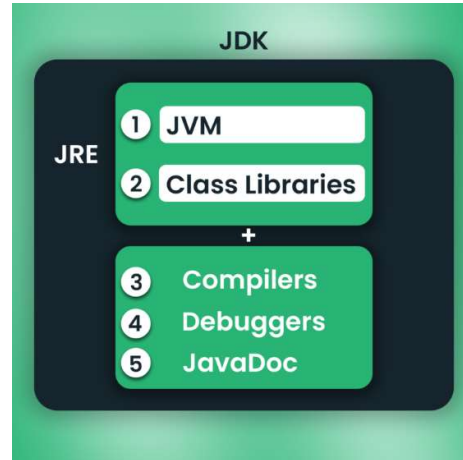
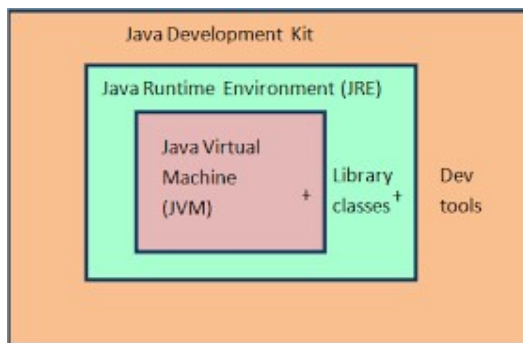
### Java Architecture

**Java Architecture** is a collection of components, namely **JVM (Java Virtual Machine)**, **JRE (Java Runtime Environment)**, and **JDK (Java Development Kit)**. It orchestrates the process of both interpretation and compilation, delineating all processes involved in creating a Java program.

#### Components of Java Architecture

Java architecture comprises three main components:

- **Java Virtual Machine (JVM)**
- **Java Runtime Environment (JRE)**
- **Java Development Kit (JDK)**



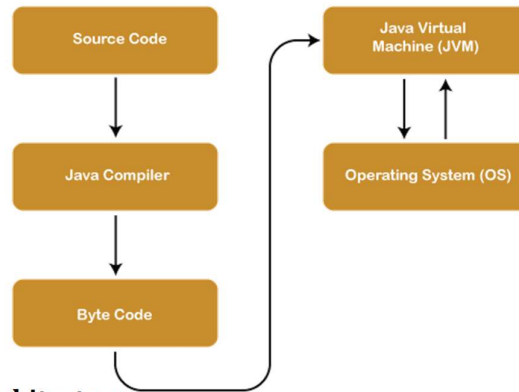
#### Java Architecture

**Java Architecture** is a collection of components, i.e., **JVM, JRE, and JDK**. It integrates the process of interpretation and compilation. It defines all the processes involved in creating a Java program. **Java Architecture** explains each and every step of how a program is compiled and executed.

**Java Architecture** can be explained by using the following steps:

- There is a process of compilation and interpretation in Java.
- Java compiler converts the Java code into byte code.

- After that, the JVM converts the byte code into machine code.
- The machine code is then executed by the machine.



### Components of Java Architecture

The Java architecture includes the three main components:

- Java Virtual Machine (JVM)
- Java Runtime Environment (JRE)
- Java Development Kit (JDK)

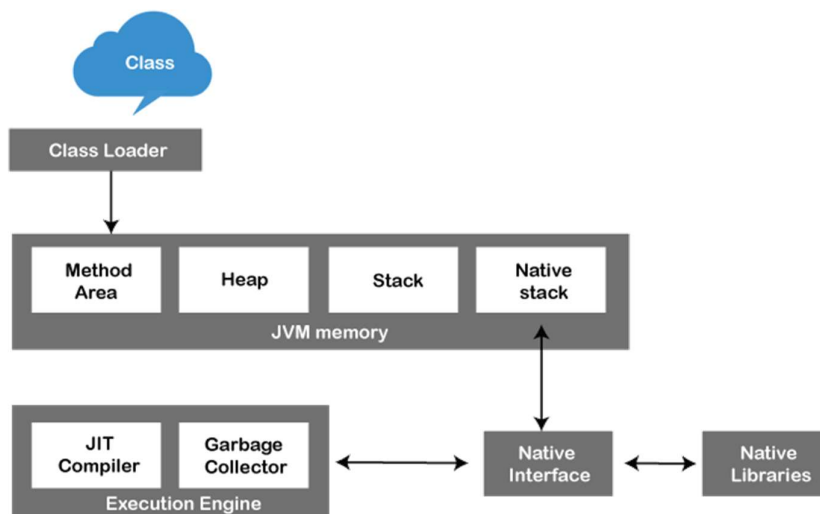
### Java Virtual Machine

The main feature of Java is **WORA**. WORA stands for **Write Once Run Anywhere**. The feature states that we can write our code once and use it anywhere or on any operating system. Our Java program can run any of the platforms only because of the Java Virtual Machine. It is a Java platform component that gives us an environment to execute java programs. JVM's main task is to convert byte code into machine code.

JVM, first of all, loads the code into memory and verifies it. After that, it executes the code and provides a runtime environment. Java Virtual Machine (JVM) has its own architecture, which is given below:

### JVM Architecture

JVM is an abstract machine that provides the environment in which Java bytecode is executed. The falling figure represents the architecture of the JVM.



**ClassLoader:** ClassLoader is a subsystem used to load class files. ClassLoader first loads the Java code whenever we run it.

**Class Method Area:** In the memory, there is an area where the class data is stored during the code's execution. Class method area holds the information of static variables, static methods, static blocks, and instance methods.

**Heap:** The heap area is a part of the JVM memory and is created when the JVM starts up. Its size cannot be static because it increases or decrease during the application runs.

**Stack:** It is also referred to as thread stack. It is created for a single execution thread. The thread uses this area to store the elements like the partial result, local variable, data used for calling method and returns etc.

**Native Stack:** It contains the information of all the native methods used in our application.

**Execution Engine:** It is the central part of the JVM. Its main task is to execute the byte code and execute the Java classes. The execution engine has three main components used for executing Java classes.

- **Interpreter:** It converts the byte code into native code and executes. It sequentially executes the code. The interpreter interprets continuously and even the same method multiple times. This reduces the performance of the system, and to solve this, the JIT compiler is introduced.
- **JIT Compiler:** JIT compiler is introduced to remove the drawback of the interpreter. It increases the speed of execution and improves performance.
- **Garbage Collector:** The garbage collector is used to manage the memory, and it is a program written in Java. It works in two phases, i.e., **Mark** and **Sweep**. Mark is an area where the garbage collector identifies the used and unused chunks of memory. The Sweep removes the identified object from the **Mark**

### **Java Native Interface**

Java Native Interface works as a mediator between Java method calls and native libraries.

### **Java Runtime Environment**

It provides an environment in which Java programs are executed. JRE takes our Java code, integrates it with the required libraries, and then starts the JVM to execute it

### **Java Development Kit**

It is a software development environment used in the development of Java applications and applets. Java Development Kit holds JRE, a compiler, an interpreter or loader, and several development tools in it.

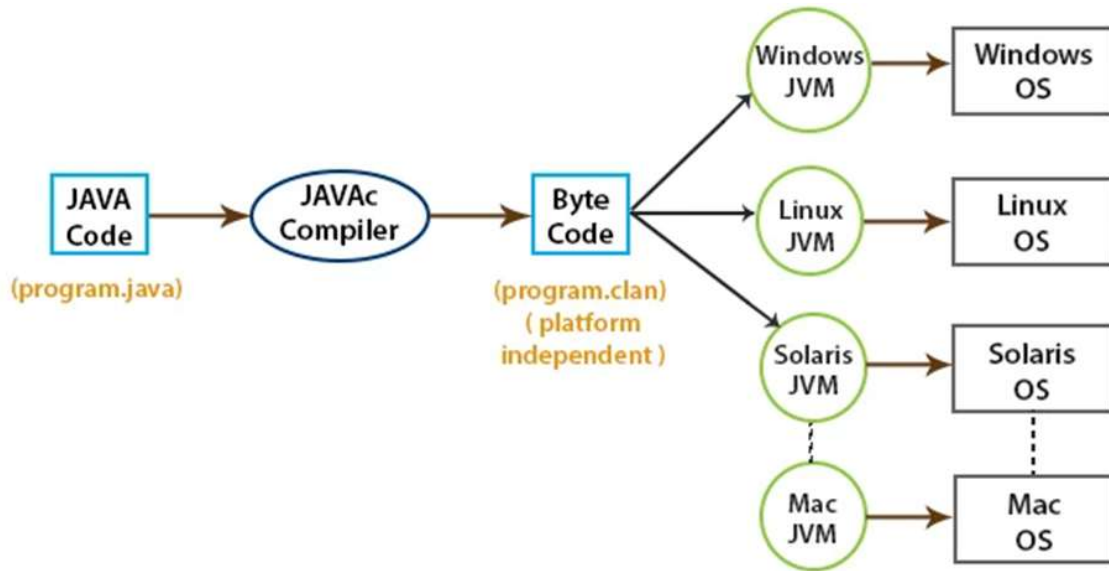
These are three main components of Java Architecture. The execution of a program is done with all these three components.

### **How is Java platform independent?**

The meaning of Java platform-independent is that the Java compiled code(byte code) can run on all operating systems in java the bytecode produced by the javac compiler may be run on a variety of Operating Systems.

In Java, the compiler converts the program into bytecode, which is not executable code. Now, any other platform or operating system may run this bytecode. When we execute its program or file, we obtain a .class file, which is a non-executable file. We would require a virtual machine to

execute a .class file. Java JVM is undoubtedly platform-independent architecture. Similar to how it varies depending on the surroundings. For instance, MAC and Linux have a different JVM than Windows. It supports portability as one of its key features since its code's portable bytecode makes it platform-independent.



### 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. JVM	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.
JDR is platform dependent.	JRE is also platform dependent.	JVM is platform- independent.
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