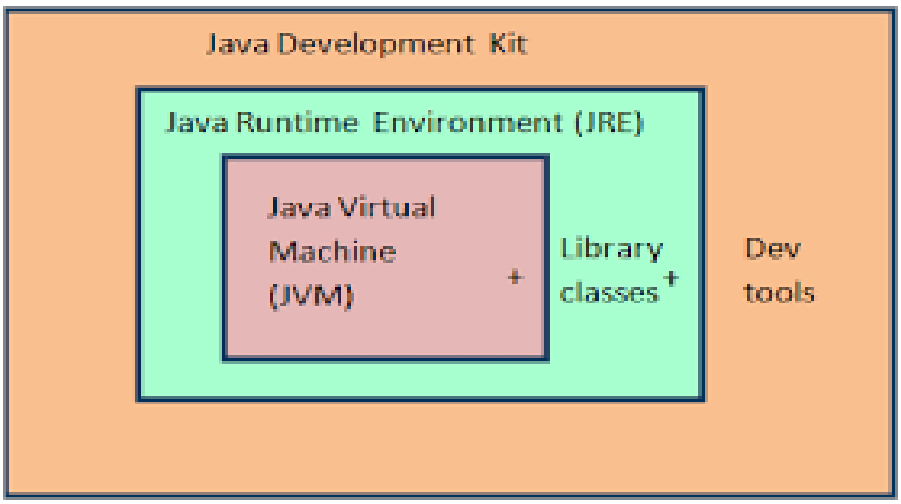
JAVA ARCHITECTURE

- **Java Architecture** is a collection of components, i.e.,
- **JVM**
- **JRE**
- **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.

COMPONENTS OF ARCHITECTURE



JAVA DEVELOPMENT KIT (JDK)

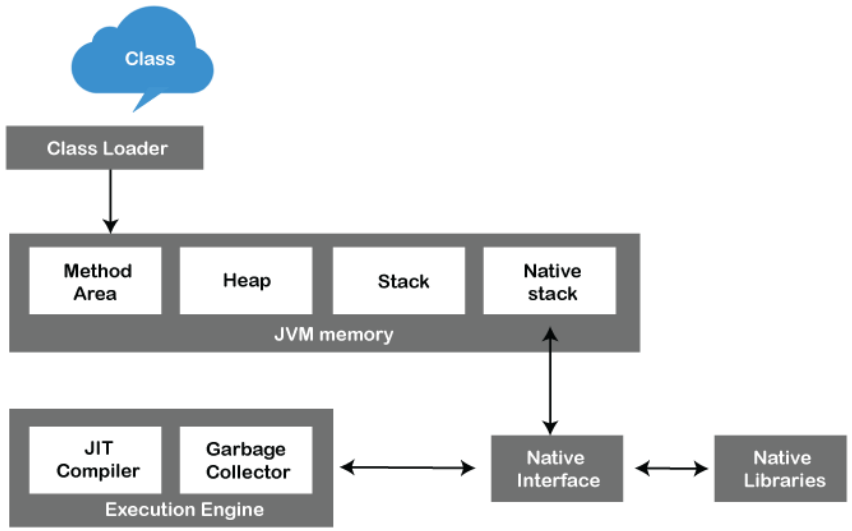
- It is a software development environment used in the development of Java applications and applets.

- Java Development Kit holds JRE, a compiler, an interpreter or loader, and several development tools in it.

JAVA VIRTUAL MACHINE(JVM)

- 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 ARCHITECTURE



- **Class Loader:** Class Loader is a subsystem used to load class files. Class Loader 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 increase 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.

- **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**.

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

JAVA RUNTIME ENVIRONMENT(JRE)

- 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.

THANK YOU