# JAVA

# WHAT IS JAVA?

- **Java** is a general-purpose, class-based, object-oriented programming language designed for having lesser implementation dependencies.
- It is a computing platform for application development. Java is fast, secure, and reliable, and it is widely used for developing Java applications in laptops, data centers, game consoles, scientific supercomputers, cell phones, etc.

# WHAT IS JAVA?

- Java is a multi-platform, object-oriented, and network-centric language. It is among the most used programming language. Java is also used as a computing platform.

# HISTORY OF JAVA PROGRAMMING LANGUAGE

- The Java language was initially called OAK.
- Originally, it was developed for handling portable devices and set-top boxes. Oak was a massive failure.
- In 1995, Sun changed the name to “Java” and modified the language to take advantage of the burgeoning www (World Wide Web) development business.
- Later, in 2009, Oracle Corporation acquired Sun Microsystems and took ownership of three key Sun software assets: Java, MySQL, and Solaris.

## WHAT IS JAVA PLATFORM?

- **Java Platform** is a collection of programs that help programmers to develop and run Java programming applications efficiently.
- It includes an execution engine, a compiler, and a set of libraries in it. It is a set of computer software and specifications.

# JAVA VERSIONS

- **The latest version of Java is Java 16** or JDK 16 released on March, 16th 2021.

| Java Versions | Release Date |
|---|---|
| JDK Alpha and Beta | 1995 |
| JDK 1.0 | 23rd Jan 1996 |
| JDK 1.1 | 19th Feb 1997 |
| J2SE 1.2 | 8th Dec 1998 |
| J2SE 1.3 | 8th May 2000 |
| J2SE 1.4 | 6th Feb 2002 |
| J2SE 5.0 | 30th Sep 2004 |
| Java SE 6 | 11th Dec 2006 |
| Java SE 7 | 28th July 2011 |
| Java SE 8 | 18th Mar 2014 |
| Java SE 9 | 21st Sep 2017 |
| Java SE 10 | 20th Mar 2018 |
| JAVA SE 11 | 25th Sep 2018 |
| JAVA SE 12 | 19th Mar 2019 |
| JAVA SE 13 | 17th Sep 2019 |
| JAVA SE 14 | 17th Mar 2020 |
| JAVA SE 15 | 15th Sep 2020 (latest Java Version) |

# COMPONENTS OF JAVA PROGRAMMING LANGUAGE

- A Java Programmer writes a program in a human-readable language called Source Code. Therefore, the CPU or Chips never understand the source code written in any programming language.
- These computers or chips understand only one thing, which is called machine language or code. These machine codes run at the CPU level. Therefore, it would be different machine codes for other models of CPU.
    - **Java Development kit (JDK)**
    - **Java Virtual Machine (JVM)**
    - **Java Runtime Environment (JRE)**

# JAVA DEVELOPMENT KIT (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.
- **Main reasons for 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.

# JAVA VIRTUAL MACHINE (JVM)

- Java Virtual Machine (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 the Java Run Environment (JRE).
- In other programming languages, the compiler produces machine code for a particular system. However, the Java compiler produces code for a Virtual Machine known as Java Virtual Machine
- **Main reasons for using JVM:**
  - JVM provides a platform-independent way of executing Java source code.
  - It has numerous libraries, tools, and frameworks.
  - Once you run a 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 faster than a regular application.

# JAVA RUNTIME ENVIRONMENT (JRE)

- JRE is a piece of software that is designed to run other software.
- It contains the class libraries, loader class, and JVM.
- In simple terms, if you want to run a 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.
- **Main reasons for using JRE:**
  - JRE contains class libraries, JVM, and other supporting files. It does not include any tool for Java development like a debugger, compiler, etc.
  - It uses important package classes like math, swing, util, lang, awt, and runtime libraries.
  - If you have to run Java applets, then JRE must be installed in your system.

# DIFFERENT TYPES OF JAVA PLATFORMS

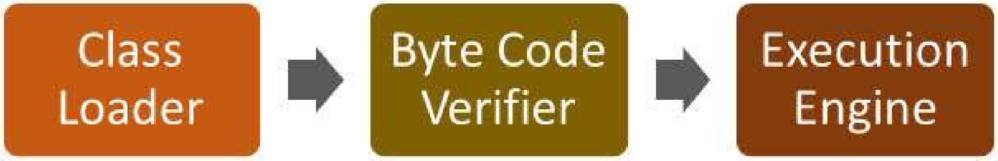
- **1. Java Platform, Standard Edition (Java SE):** Java SE's API offers the Java programming language's core functionality. It defines all the basis of type and object to high-level classes. It is used for networking, security, database access, graphical user interface (GUI) development, and XML parsing.
- **2. Java Platform, Enterprise Edition (Java EE):** The Java EE platform offers an API and runtime environment for developing and running highly scalable, large-scale, multi-tiered, reliable, and secure network applications.
- **3. Java Programming Language Platform, Micro Edition (Java ME):** The Java ME platform offers an API and a small-footprint virtual machine running Java programming language applications on small devices, like mobile phones.
- **4. Java FX:** JavaFX is a platform for developing rich internet applications using a lightweight user-interface API. It user hardware-accelerated graphics and media engines that help Java take advantage of higher-performance clients and a modern look-and-feel and high-level APIs for connecting to networked data sources.

# WHAT IS JVM?

- **Java Virtual Machine (JVM)** is a engine that provides runtime environment to drive the Java Code or applications. It converts Java bytecode into machines language. JVM is a part of Java Runtime Environment (JRE). In other programming languages, the compiler produces machine code for a particular system. However, Java compiler produces code for a Virtual Machine known as Java Virtual Machine.

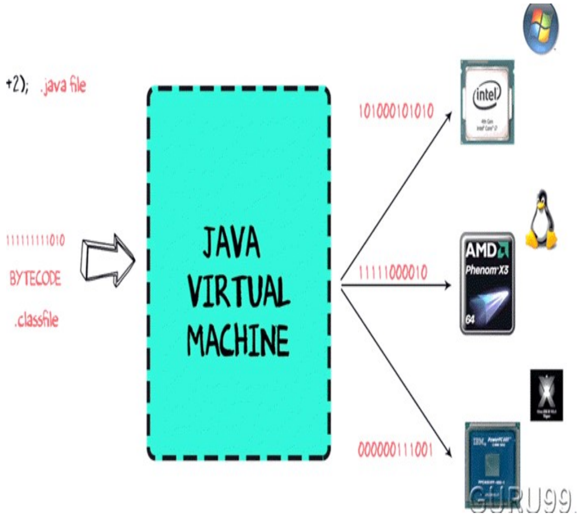
# HOW JAVA VIRTUAL MACHINE WORKS?

- **Here is how JVM works**
- First, Java code is compiled into bytecode. This bytecode gets interpreted on different machines
- Between host system and Java source, Bytecode is an intermediary language.
- JVM in Java is responsible for allocating memory space.



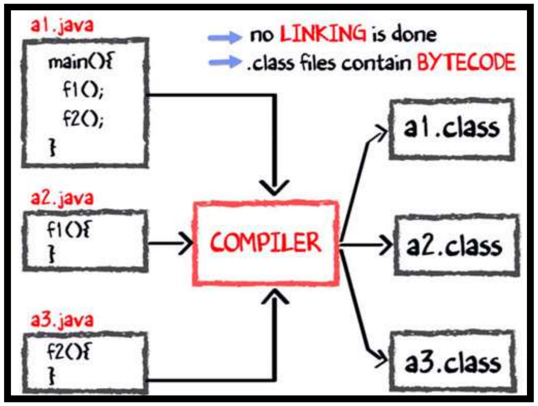
# HOW JAVA VIRTUAL MACHINE WORKS?

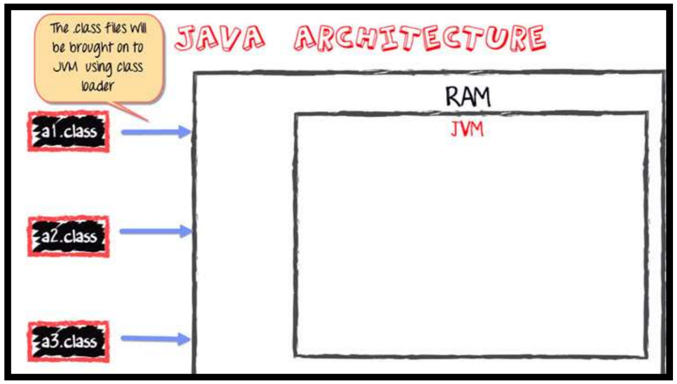
- **Step 1)** The code to display the addition of two numbers is System.out.println(1+2), and saved as a .java file.
- **Step 2)** Using the java compiler the code is converted into an intermediate code called the **bytecode.** The output is a **.class file.**
- **Step 3)** This code is not understood by any platform, but only a virtual platform called the **Java Virtual Machine.**
- **Step 4)** This Virtual Machine resides in the RAM of your operating system. When the Virtual Machine is fed with this bytecode, it identifies the platform it is working on and converts the bytecode into the native machine code.

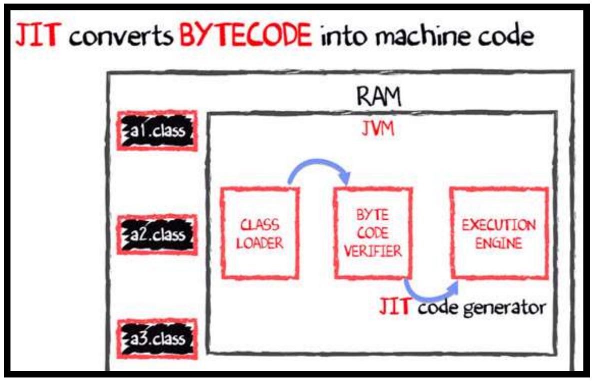


# JAVA CODE COMPILATION AND EXECUTION IN JAVA VM

- During execution, using the class loader the class files are brought on the RAM. The BYTE code is verified for any security breaches.
- The execution engine will convert the Bytecode into Native machine code. This is just in time compiling.




The .class files will be brought on to JVM using class loader
JAVA ARCHITECTURE
RAM
JVM
a1.class
a2.class
a3.class



# WHY IS JAVA BOTH INTERPRETED AND COMPILED LANGUAGE?

- A **compiler** is a program which converts a program from one level of language to another.
- The java compiler converts high-level java code into bytecode.
- An **interpreter** is a program which converts a program at one level to another programming language at the **same level.**
- In Java, the Just In Time Code generator converts the bytecode into the native machine code which are at the same programming levels.
- Hence Java is both compiled as well as interpreted language.