**Java Programming**

**What is Java.?**

**Java** is a class-based object-oriented programming language. Which is used to create Web and desktop application. Java is the most usable programming language by android development. Hence, Java can be used to create android mobile application. Java is fast, most secure and reliable; therefore, Java is used for developing computer applications, data centers, smartphones, gaming consoles, etc.

**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. James Gosling developed the Java platform at Sun Microsystems in 1995, and the Oracle Corporation later acquired it.

**Java Definition and Meaning**

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

A **computing platform** is a well-defined set of [hardware](https://simple.wikipedia.org/wiki/Computer_hardware) or [software](https://simple.wikipedia.org/wiki/Software) that allows application programs to run. Typical platforms include the hardware architecture, the [operating system](https://simple.wikipedia.org/wiki/Operating_system), as well as runtime libraries. The [programming language](https://simple.wikipedia.org/wiki/Programming_language) [Java](https://simple.wikipedia.org/wiki/Java_(programming_language)) is an example of Software Platform.

**What is Java used for.?**

* It is used for developing android applications
* Using java, we can create Enterprise software
* Wide range of mobile java applications
* Used for big data analytics
* Scientific computing applications
* Java programming and hardware devices
* Used for server-side technologies like Apache, JBoss, Glassfish, etc.

**History of Java programing 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.

**Java Versions**

|  |  |
| --- | --- |
| 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) |

**Java Features**

* It is one of the easy-to-use programming languages to learn.
* Write code once and run it on almost any computing platform.
* Java is platform-independent. Some programs developed in one machine can be executed in another machine.
* It is designed for building object-oriented applications.
* It is a multithreaded language with automatic memory management. i.e., Multithreading is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU.
* It is created for the distributed environment of the Internet.
* Facilitates distributed computing as its network-centric.

**Component of Java Programing 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](https://www.guru99.com/best-programming-language.html).

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.

However, you need to worry about the machine code, as programming is all about the source code. The machine understands this source code and translates them into machine understandable code, which is an executable code.

All these functionalities happen inside the following 3 Java platform components:

**Java Development Kit (JDK)**

JDK is a Java development envirnment which is used for making Applets and Java applications. JDK stands for Java Development Kit. Java developers can use it on multiple platforms like windows, macOS, Solaris, and Linux. JDK helps them to code and run the Java program. It is possible to install more than one JDK version on same computer.

**Why use JDK.?**

* JDK contains tools required write Java programs and JRE to execute them
* It includes Java compiler, Java Application launcher and Applet viewer.
* Compiler convert code written in Java into Byte code.
* Java application launcher opens a JRE. Load the necessary class files, and executes its main method.

**Java Virtual Machine (JVM)**

* Java virtual machine is an engine that provides a runtime envirnment to drive Java code or program or application
* It converts Java Byte code into machine language
* JVM is a part of Java Runtime Envirnment (JRE)
* In other programming language, Compiler produces the machine code for a particular computer. However, Java compiler produce Java Byte code for Java virtual Machine.

**Why JVM**

* JVM provided platform-Independent envirnment to Java program or application. i.e., once the Java code is run it can be run on multiple machine
* Java virtual Machine comes with JIT (Just-in-time) compiler that convert Java source code into low-level machine language. Hence Java application is faster than regular apps
* JVM provide numbers of libraries, tools, and frameworks

**Architecture of JVM**

First the code is compiled and it converted into Byte code**.** then the Byte code interpreted on different machines.

Between host system and Java source, Bytecode is an intermediary language.

JVM in Java is responsible for allocating memory space.



Now in this JVM tutorial, let's understand the Architecture of JVM. JVM architecture in Java contains class loader, memory area, execution engine etc.



The class loader is used to load the class. Files into the RAM. Class loader performs three important functions via. Loading, linking and initialisation.

**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 class structures like metadata, the constant runtime pool, and the code for methods.

**3) Heap**

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

**4) JVM language Stacks**

Java language Stacks store local variables, and it’s partial results. Each thread has its own JVM stack, created simultaneously as the thread is created. A new frame is created whenever a method is invoked, and it is deleted when method invocation process is complete.

**5)  PC Registers**

PC register 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 is written in another language instead of Java.

**7) Execution Engine**

It is a type of software used to test hardware, software, 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.

**Java Runtime Envirnment (JRE)**

JRE is a piece of software that is designed to run other software. JRE contains class libraries, loader class, and JVM. If you want run Java program you need JRE. JRE is just run Java program.

**Why use JRE**

* JRE is only a software that contains class libraries, loader class, JVM and other supporting files it helps to run Java code.
* It does not include any development tools like a debugger, compiler, etc.
* It uses important package classes like math, swing, util, lang, awt, and runtime libraries.
* If you want to run Java applet. JRE must installed in your computer

**How Java Virtual Machine Works:**

By using **Java Virtual Machine**, this problem can be solved. But how it works on different processors and O.S. Let's understand this process step by step.

**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.

While working on your PC or browsing the web, whenever you see either of Java icons, be assured the java virtual machine is loaded into your RAM. But what makes Java lucrative is that code, once compiled, can run not only on all PC platforms but also on mobiles or other electronic gadgets supporting Java.

Hence, Java is Platform independent programming language

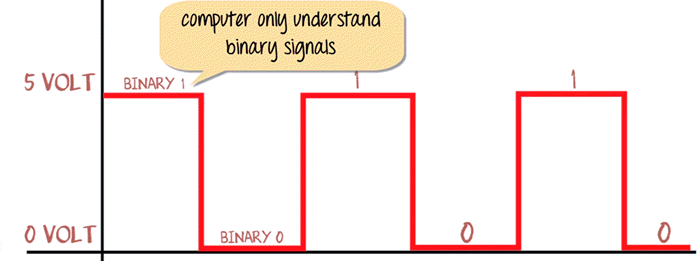
**Different Types of Java Platforms OR Editions**

There are four different types of Java programming language platforms:

1. **Standard Edition (Java SE):** Java SE’s API offers the Java programming language’s core functionality. It defines all basis of type and object to high-level classes. It is used for networking, security, development, database access, graphical user interface (GUI)
2. **Enterprise Edition (Java EE):** Java EE platform offers an API and Runtime envirnment for developing large scale, high-level, multi-tiered, reliable and highly secure applications.
3. **Micro Edition (Java ME):** Java ME edition offers an API and runtime envirnment and small foot-print virtual machine running Java programming applications on small devices like mobiles and smartphones
4. **Java FX:** JavaFX is a platform for developing rich internet applications using a lightweight user-interface API. Its 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.

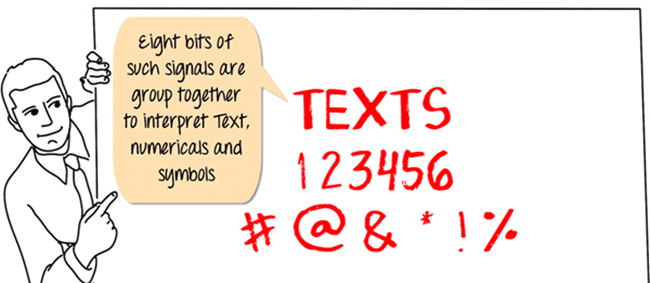
**What is Assembly language.?**

The computer is an electronic device, and it can only understand electronic signals or binary signals. For example, the 5-volt electronic signal may represent binary number 1, while 0 volts may represent binary number 0. So, your PC is continuously bombarded with these signals.



Eight bits of such signals are grouped together to interpret Text, numerical, and symbols.

For example, the # symbol is identified by the computer as 10101010. Similarly, the pattern for adding a function is represented by 10000011.



This is known as 8-bit computing. Current day processor is capable of decoding 64-bit time. But what is the relation of this concept with the programming language JAVA? Let understand these as an example.

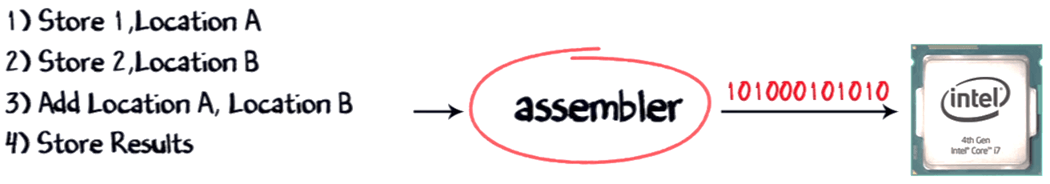
We are going to give the command to a computer in this format, as shown below. Your code to add two numbers in this language would be in this order.

* Store number 1 at memory location say A
* Store number 2 at memory location say B
* Add contents of Location A & B
* Store results

But how are we going to do this? Back in the 1950s, when computers were huge and consumed a great deal of power, you would convert your assembly code into corresponding machine code to 1 and 0's using mapping sheets. Later, this code will be punched into the machine cards and feed to the computer. The computer will read these codes and execute the program. This would be a long process then until ASSEMBLER came to help.

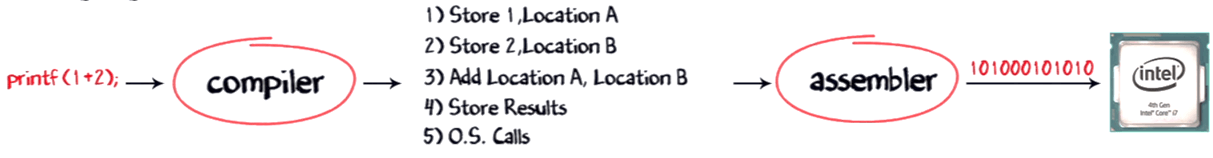
**What is Assembler & Compiler.?**

With the advancement in technology, I/o devices were invented. You could directly type your program into the PC using ASSEMBLER. It converts it into the corresponding machine code (110001...) and feeds it to your processor. Coming back to our example addition of (1+2), the assembler will convert this code into machine code and output.



That apart, you will also have to make calls to create Operating System provided functions to display the code's output.

But alone the assembler is not involved in this process; it also requires the compiler to compile the long code into a small chunk of codes. With the advancement in software development languages, this entire assembly code could shrink into just one line **print f 1+2 A** with the software called COMPILER. It is used to convert your c language code into assembly code. The assembler converts it into corresponding machine code. This machine code will be transmitted to the processor. The most common processor used in PC or Computers are the Intel processor.



Though present-day compilers come bundled with assembler can directly convert your higher language code into machine code.

As a developer, I want my software program to work on all platforms to maximize my revenues. So, I would have to buy separate compilers that convert my print f command into the native machine code.



But compilers come expensive, and there is a chance of compatibility issues. So, buying and installing a separate compiler for different OS and processor is not feasible. So, what can be an alternative solution? Enter Java language.

**How Java is Platform Independent.?**

Like the C compiler, the Java compiler does not produce native executable code for a particular machine. Instead, Java produces a unique format called bytecode. This JVM allows programmers to run a code on multiple computers. Therefore, Java is a platform-independent language.

Bytecode is understandable to any JVM installed on any OS. In short, the java source code can run on all operating systems.

Programming Language are classified as

* Higher Level Language Ex. C++, Java
* Middle-Level Languages Ex. C
* Low-Level Language Ex Assembly
* finally, the lowest level as the Machine Language.

A **compiler** is a program which converts a program from one level of language to another. Example conversion of C++ program into machine code.

The java compiler converts high-level java code into bytecode (which is also a type of machine code).

An **interpreter**is a program which converts a program at one level to another programming language at the **same level.** Example conversion of Java program into C++

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.

# **Summary:**

* Java is a multi-platform, object-oriented, and network-centric programming language Java is a general-purpose, class-based, object-oriented programming language.
* Java Platform is a collection of programs that help programmers to develop and run Java applications efficiently.
* Meaning of Java: Java is a multi-platform and network-centric programming language.
* It is mainly used for developing Android Apps and Enterprise Software.
* 2009, Oracle Corporation acquired Sun Microsystems and took ownership of three key Sun software assets: Java, Solaris, and MySQL.
* The latest version of Java released on September 15th, 2020
* The best feature of the Java is that it is one of the easiest programming languages to learn.
* Four types of Java Programming language platforms are:
  + 1) Java Platform, Standard Edition (Java SE)
  + 2) Java Platform, Enterprise Edition (Java EE)
  + 3) Java Platform, Micro Edition (Java ME)
  + 4) JavaFX
* A computer is an electronic device capable of performing computations.
* The computer only understands electronic signals or binary signals.
* Assembler is an advanced technology that converts source core to corresponding machine code (110001...) and feeds to your processor