Java Fundamentals

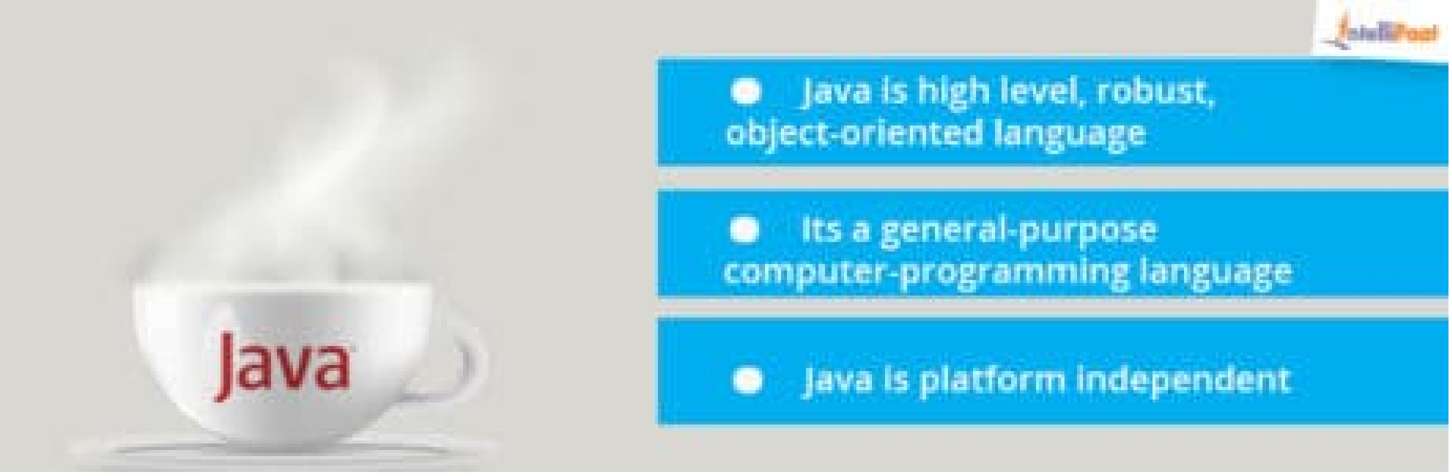
Regardless of whether one thinks that Java is now growing, or inactive, or dying, one has to admit that Java had truly revolutionised and had defined the programming language. Java had combined and brought to the tons of useful programming language features that were previously available only separately or in various languages. Java is a programming language and platform released by **Sun Microsystems** in 1995. Java is a secure, fast, and reliable programming language. Java is everywhere from PCs to Mobile phone, satellites, other electronic devices. There are many applications and websites that will not run without Java installation on your machine.

Watch this Video on How to Install Java on Windows 10

Table of Content:

What is Java?

Java is a programming language which produces software that can be used for gaming, web, business, desktop, database, and other applications. It exhibits most of its syntax from C/C++.



- Java is high level, robust, object-oriented language
- Its a general-purpose computer-programming language
- Java is platform independent

Java is a simple, fast, object-oriented, robust, secure, multi-threaded, and platform-independent programming language. Java platform consists of several programs, each of which provides a portion of its overall capabilities.
Basically, Java consists of two portions-

- **Code**
- **Platform**

Watch this Java Video:

Code consists of source code and byte code. Source Code is what a programmer writes in the text editor, then Java Compiler compiles that and produces byte code.
Java platform includes an execution engine, a compiler, and a set of libraries that help to develop programs.

Java History

Java was designed by Sun Microsystems in the early 1990s to solve the problem of connecting many household machines together. In 1991 java was started initially it was called as “Oak”. Later Java became popular for developing web internet applications. Java was developed by James Gosling and a small team of engineers. James Gosling is considered as “FATHER OF JAVA”. The main concept of java is “Write Once, Run Anywhere”. Java first publication 1.0 was released in 1995 by Sun Microsystems. Now, Sun Microsystem is the part of Oracle Corp.

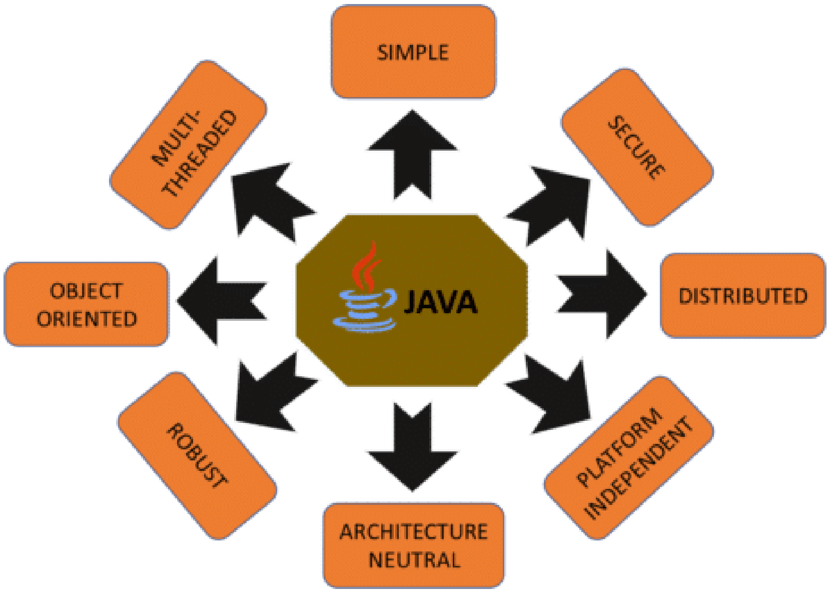
Looking for Advanced Courses?
Certification in Big Data Analytics
In partnership with E&ICT, IIT Guwahati
KNOW MORE
400+ Hrs of Instructor-led Training | Job Assistance

Features of Java

Java provides very good features which are better as compared to C/C++. These features are:

- **Simple**: Java is easy to learn because most of the complex features of C/C++ like pointers, operator overloading, etc. are not provided in Java.
- **Secure**: Java programs run within the JVM which protects from unauthorised access to system resources.
- **Platform Independent/Portable**: Java program can be executed on any kind of machine containing any CPU or operating system.

- **Architecture Neutral**: Since it can run on any operating system, so it is architecture-neutral.
- Robust: Java is robust because of having strong memory management, no pointers, exception handling, type checking mechanism, platform-independent.
- **Multi-threaded**: Java allows us to write a program that can do many things simultaneously.
- **Object-Oriented**: Java follows OOPs model that helps to break the complex code into easy to understand objects.



Go for this in-depth job-oriented Java Training Course now!

Installation of Java

As you know the features of Java, let’s move into the steps to install Java.

Step 1: Install JDK:

- Visit site @ http://www.oracle.com/technetwork/java/javase/downloads/index.html, choose JDK according to your operating system & download.

Step 2: Set PATH to include JDK’s directory

- **Set PATH permanently in System Environment Variable**

On Windows: Use the following command:-
Right click on My Computer ->Properties ->Advanced System setting ->Environment Variable ->new button in the user variable
Provide the required info:

- Variable name- PATH
- Variable value- C:\jdk1.8.o\bin

For Example:

If it contains ‘C:\Windows\System32′ then change your path by ‘C:\Windows\System32; C:\Program Files\java\bin’.

Click **OK** button

On Linux, UNIX, Solaris

Use the following command:

export PATH=$PATH:/home/jdk1.6.01/bin/

Where JDK is installed in the home directory under Root (home)

- Set path temporary in command line

Say you have saved a program in D drive, then

D:\set PATH=%PATH%; C:\jdk1.8.0\bin;

Java First Program & JVM, JRE, JDK Story:

After Java installation, write the below hello program in any text editor, save your program with the same name as that of the class, compile and execute.

Hello Java Program

```
class Hello{public static void main(String args[]){
/* multi-line comment */
System.out.println("Hello Intellipaat"); //single line comment
}
}
```

```
javac Hello.java
java Hello
```

Output: Hello Intellipaat

Program Explanation

The above program contains the following main parts which are important before going into the main topics:

- **class** definition**:** class is a keyword that tells that a new class has created. “Hello” is used as an identifier to declare a class name. Entire class starts with a curly brace { and ends with } curly brace.
- **main** method**:** Execution of each program starts with the main method. Each program must have a main method. Syntax of the main method
- **public static void main(String args[])** or **static public void main(String args[])**
    - **public:** public so that JVM can execute the program from anywhere
    - **static:** so that main method can be called without creating object.
    - **void:** method main doesn’t return anything.
    - **String []args:** to pass arguments from command line as String array elements
- **System.out.print()**: Statements inside () in double quotation marks will be printed as it is, and without double quotation marks, values will be printed. Example:
    - **System.out.println(“Hello Intellipaat”);** will print Hello Intellipaat and **System.out.println(10);** will print 10.
- **Comments**: Java supports two types of comments to have a better understanding of the program.
    - **Single Line Comment**: It is used to comment on a single line. The statement in the above code- **//**, is used to write single line comment.
    - **Multi-Line Comment**: It is used to comment on multiple lines. The statement in the above code- **/* <statements> */**, is used to write multi-line comments.

Learn more about Java Certification from this insightful Java Certification blog!

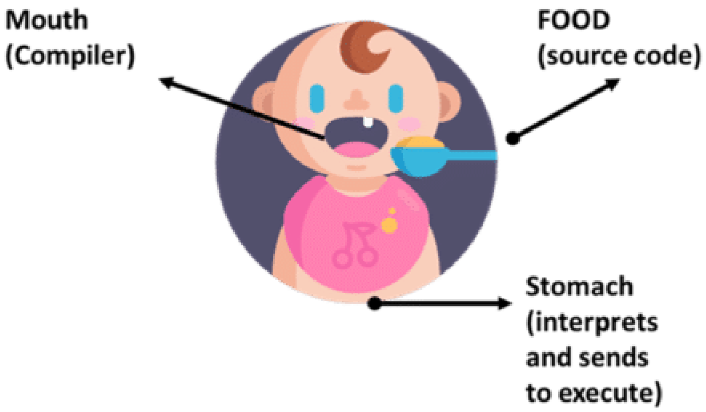
JVM, JDK, JRE

In the previous topic, you learned about the basic java program. But the question is **what is helping Java to compile and execute this code?**

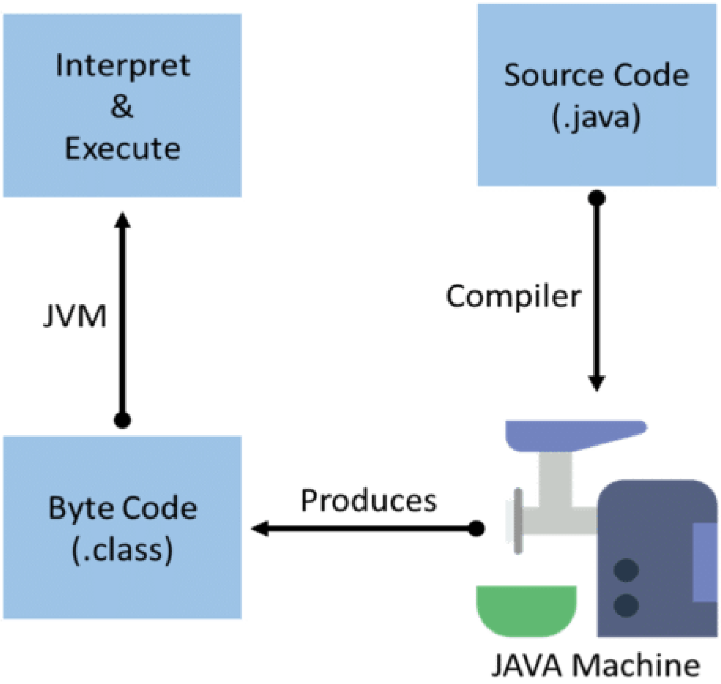
There is something in java that helps in the compilation and execution of a program.

Let’s start with source code and byte code. The program which you write in the text editor is called as source code. The source code is saved with .java extension. The written code is converted by the java compiler into byte code with extension .class. This byte code can be run into any system that contains Java software. After that, java machine interprets the code and executes the program.

This is just like human digestive system.

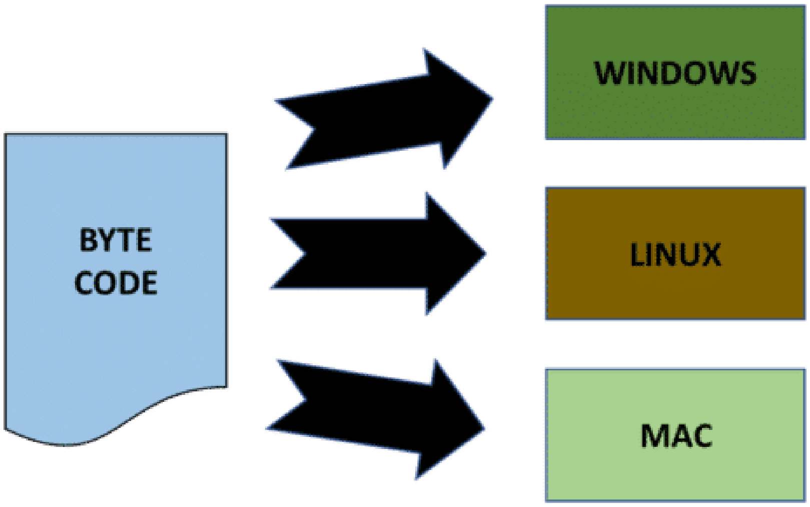


The food you eat is the source code. Mouth acts as the compiler and converts the food by chewing into the byte code so that the food can pass through the throat.
Stomach stores the food and digests it which can be considered as interpret and execute.



Important Points:

- **Source code**: the one user saves with .java extension like Hello.java
- **Byte code**: java compiler compiles the source code and generates byte code. This byte code can be run on any kind of Operating System having java software. That is why it is called as platform-independent. Byte code contains the .class file.



JVM (JAVA INTERPRETER)

- JVM (Java Virtual Machine) acts as a run-time engine to run java applications. JVM is the part of JRE.
- JVM calls the main method after the execution of the program starts.
- JVM contains JIT(Just In Time) that converts byte code into machine code just before the execution of the program.

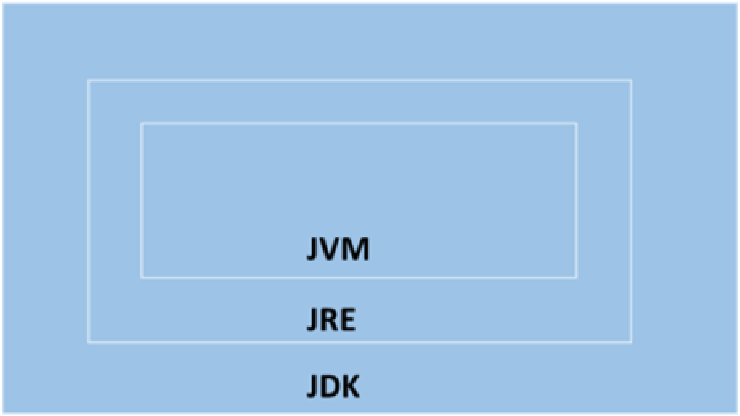
JRE

- JRE stands for Java Runtime Environment and can be also written as Java RTE.
- JRE contains JVM, other classes, and libraries that are required during the program execution.

JDK (JAVA COMPILER)

- JDK allows developers to create Java programs that can be executed and run by JVM and JRE.
- JDK contains JRE, java compiler, and other java tools.

See the picture to have a better understanding of JVM, JRE, JDK



JDK = JRE + Development tools

JRE= JVM + Library classes