**History of Java**

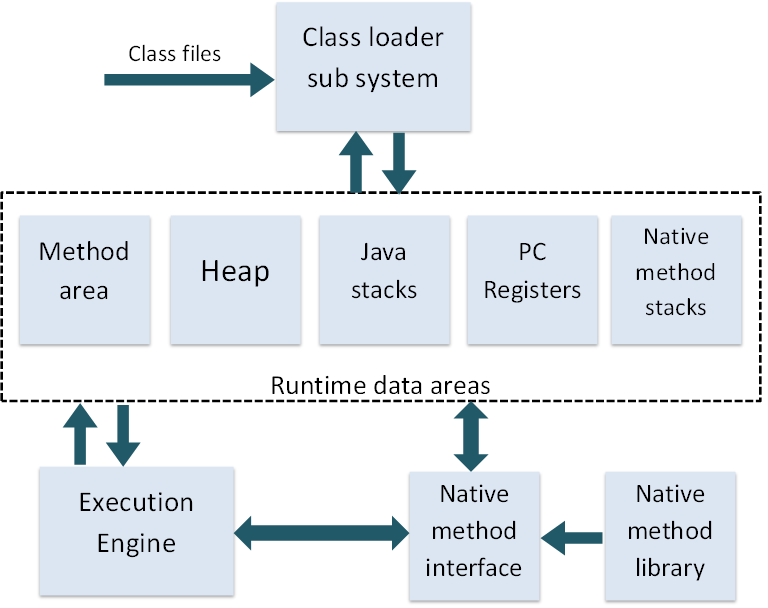
* JAVA was originated at Sun Microsystems Inc., in 1991.
* The initial name was **Oak** but it was renamed to **Java** in 1995 as OAK was a registered trademark of another Tech company. It was conceived by James Gosling and Patrick Naughton.
* Simple object oriented programming language to write, compile and debug a program easily.
* Java is fast, secure, and reliable, therefore. It is widely used for developing various applications.
* Java Editions are Java SE , Java EE , Java ME & Java Fx.
* Helps to create Developing Desktop, Web Applications, Mobile Operating System and Embedded Systems Robotics and games etc.

**Why Java...?**

* Simple
* Object – Oriented
* Platform independent
* Secured
* Portable
* Distributed
* Robust
* High Performance
* Multithreaded

**JDK, JRE & JVM**

JVM – loads, verifies and executes the code and provides the runtime environment



**JRE:**



**JDK:**



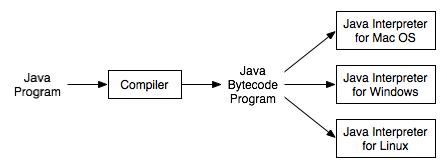
**HelloWorld.java:**

**Class** HelloWorld {

**Public** **static** **void** main (String args []) {

   System.out.println ("Hello Java");      }

}



**Class & Object in Java**

A class is a template for creating objects which define its state and behavior. A class contains field and method to define the state and behavior of its object.

**Syntax:**

<Access Modifier> class <Class Name> extends <Super\_Class\_Name> implements <Interface Name>

[**ACCESS MODIFIER**:](https://javabeginnerstutorial.com/core-java-tutorial/access-modifier-in-java/)  Defines who in the Java world can access this class and members of the class.

**CLASS\_NAME**: Unique name for the class in a specific package.

**SUPER\_CLASS\_NAME:**  Name of the class which given class extends. (Extends keyword is used for this purpose)

**INTERFACE\_NAME**: Name of an Interface which above class implements. (Implements keyword is used for this purpose)

**Object:**

There is often confusion about the difference between Class and Object. A class is the creation of a prototype and Object is the actual implementation of that prototype. Technically, Class is a template that describes what state and behavior an instance of this class can have. Object implements the state and behavior in the form of variables and methods and requires some memory allocated

**Syntax:**

<Class Name>   ClassObjectReference = new <Class Name>();