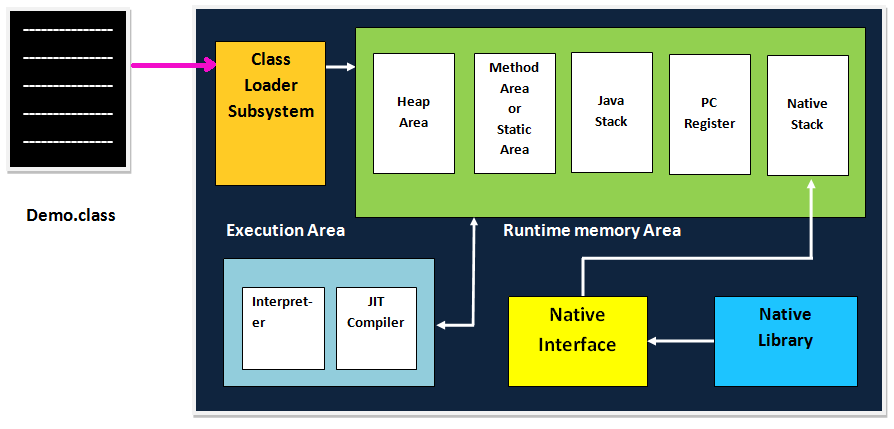
**JVM Architecture in Java**

**JVM (Java Virtual Machine)** is a software. It is a specification that provides Runtime environment in which java bytecode can be executed.

[](http://www.sitesbay.com/java/images/basic-java/JVM-Architecture.png)

**Operation of JVM**

JVM mainly performs following operations.

* Allocating sufficient memory space for the class properties.
* Provides runtime environment in which java bytecode can be executed
* Converting byte code instruction into machine level instruction.

JVM is separately available for every Operating System while installing java software so that **JVM is platform dependent**.

**Note:**Java is platform Independent but JVM is platform dependent because every Operating system have different-different JVM which is install along with JDK Software.

**Class loader subsystem:**

Class loader subsystem will load the **.class** file into java stack and later sufficient memory will be allocated for all the properties of the java program into following five memory locations.

* Heap area
* Method area
* Java stack
* PC register
* Native stack

**Heap area:**

In which object references will be stored.

**Method area**

In which static variables non-static and static method will be stored.

**Java Stack**

In which all the non-static variable of class will be stored and whose address referred by object reference.

**Pc Register**

Which holds the address of next executable instruction that means that use the priority for the method in the execution process?

**Native Stack**

Native stack holds the instruction of native code (other than java code) native stack depends on native library. Native interface will access interface between native stack and native library.

**Execution Engine**

Which contains **Interpreter** and **JIT compiler** whenever any java program is executing at the first time interpreter will comes into picture and it converts one by one byte code instruction into machine level instruction JIT compiler (just in time compiler) will comes into picture from the second time onward if the same java program is executing and it gives the machine level instruction to the process which are available in the buffer memory.

**Note:**The main aim of JIT compiler is to speed up the execution of java program.

**What is JIT and Why use JIT**

JIT is the set of programs developed by SUN Micro System and added as a part of JVM, to speed up the interpretation phase.

In the older version of java compilation phase is so faster than interpretation phase. Industry has complained to the SUN Micro System saying that compilation phase is very faster and interpretation phase is very slow.

So solve this issue, SUN Micro System has developed a program called JIT (just in time compiler) and added as a part of JVM to speed up the interpretation phase. In the current version of java interpretation phase is so faster than compilation phase. Hence java is one of the highly interpreted programming languages.