1. **How Java Is Platform independent?**

* When java program is compiled successfully it creates .class files which are error free. Then the JVM runs the .class file with JIT compiler and communicate with the operating system. This is why we can compile the java code once and run it anywhere.
* But in case of C/C++ etc, it is directly compiled into binary code, so it is depended on the machine architecture that it is compiled into.

1. **Interpreted Language vs Compiled Language:**

* Interpreted language is translated line by line into binary, while compiled language whole program is translated once.
* Interpreted language runs for every line, while compiler run only once.
* If there is any error, compiler wont compile. But interpreter will run program till the error happen.

1. **Virtual machine**: it’s a software simulation of a machine, which can perform operation like a physical machine. Two types of virtual machine:
2. **Hardware/system Based**: Within the same physical computer we can provide several logical machines with strong isolation from each other.

**Examples** cloud computing, VMware, kvm (kernel based virtual machine), xen etc.

Advantages-

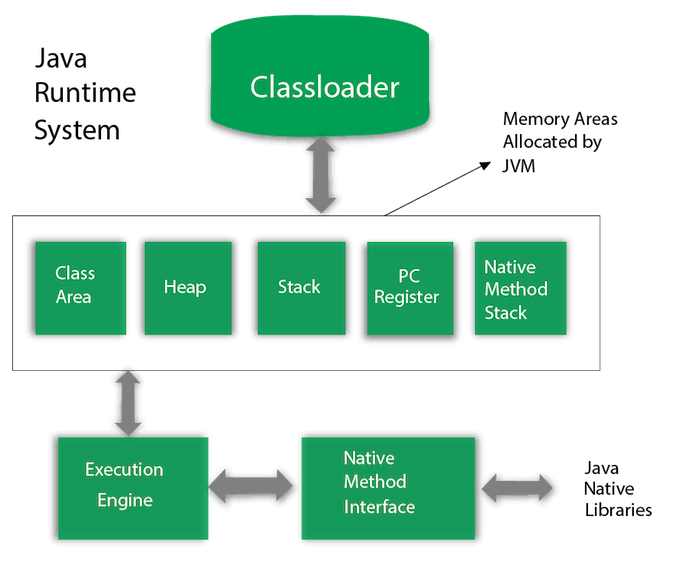
* improving the utilization of the hardware resources,
* Sharing hardware resources.

These are responsibilities for admins/network related. As a programmer, our responsibility is about software/process based virtual machine.

1. **Software/process Based**: These are runtime engines to run a particular program.

**Examples** JVM (for java), PVM (for Perl), CLR (common language runtime for .net) etc.

**JVM:** It is a runtime engine to run java applications. JVM is part of JRE, it is responsible for 2 activities- to load and run .class files.



From the pictures we can see, there is total of three modules

1. **Class loader subsystem**: It responsible for loading, linking and initialization.
   1. First step is to read and load the .class file into JVM method area. The information it loads – fully qualified name of class and its parent, method and variable information, constructor information, modifiers information, constant pool information etc.
   2. Second step is loading object of type class to represent the information of the class we loaded in the method area in the heap area. These can provide us the complete information (information about methods, constructors, variables etc.) of the class to programmers.
2. Memory areas
3. Execution engine: