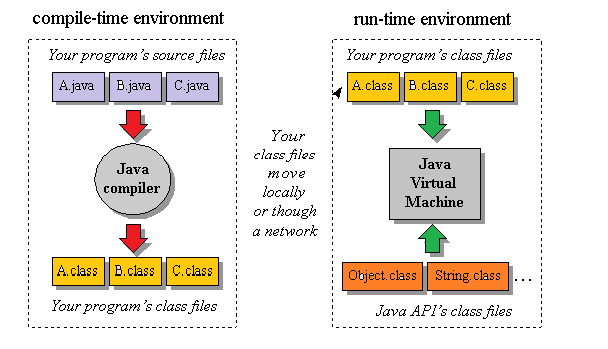
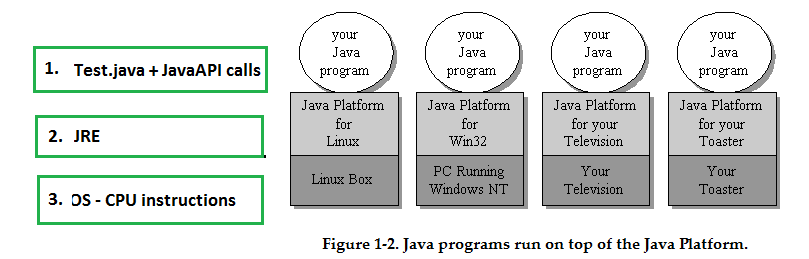
**The Architecture**

* .class
* API
* JVM

When you write your java program, you access system resources (such as I/O, for example) by calling methods in the classes of API.



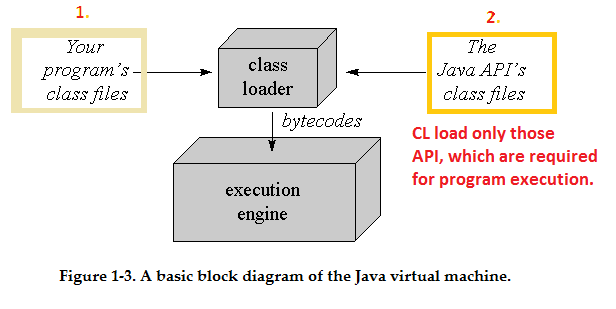
****

**The Java virtual machine** is an abstract computer.

Its specification defines certain features , but leaves many choices to the designers of each implementation.

For example –

1. All Java virtual machines must be able to execute Java bytecodes, they may use any technique to execute them.
2. The specification is flexible, Java virtual machine to be implemented **either completely in software or in hardware.**
3. The flexible nature of the JVM's specification enables it to be **implemented on a wide variety of computers and devices.**
4. A Java virtual machine's main job is to load class files and execute the byte codes.
5. Java virtual machine contains a class loader, which loads class files from both the program and the Java API. Only those class files from the Java API that are actually needed by a running program are loaded into the virtual machine. The byte codes are executed in an execution engine.



execution engine

the simplest kind of execution engine just interprets the bytecodes one at a time.

one that is faster but requires more memory, is a just-in-time compiler.

: