JVM: provides ***runtime environment***in which java bytecode can be executed. It loads ***code, verifies, and executes***.  
JRE: Used for developing ***Java apps***. It is implementation of ***JVM***. Has physically ***structure and contains set of other libraries***.

JIT: it converts ***byte code file to machine code***. Java In Time

JDK: It contains JVM and development tools and libraries, needed for java apps.

Multitasking:

Process of executing multiple tasks simultaneously.

1. Multithreading
2. Multiprocessing
3. Multithreading:
   1. Threads share same address space
   2. Thread is lightweight.
   3. If any thread occurs exception the other one runs.
4. Multiprocessing:
   1. Each process allocates separate memory
   2. Process is heavyweight.

Thread Method:

start -> starts the execution of the thread

run -> used to do an action for thread

sleep -> sleeps a thread for specified amount of time

join -> waits for a thread to die

stop -> used to stop a thread

Lifecycle of Thread:

New

Active

Blocked/Waiting

Timed Waiting

Terminated

Garbage Collection:

Memory efficient because removes unreferenced objects from heap.

Automatically done by GC

Log4J

Log4j is a fast, reliable and flexible logging framework which is written in java.

Logging is a powerful aid for understanding and debugging the runtime behavior of the programs.

Quick Debugging, Easy Maintenance, Cost and Time Savings.

Deadlock:

Deadlock can occur in a situation when a thread is waiting for an object lock, that is acquired by another thread and second thread is waiting for an object lock that is acquired by first thread.

Multi-Threading:

It is done by extending Threads class or implemented using Runnable Interface.

Thread Sync:

If we have declared 2 threads the thread 2 will be waiting for thread 1 to complete its execution in sync block (which could be method or func).