<http://javarevisited.blogspot.sg/2011/11/hotspot-jvm-options-java-examples.html>

**Important Points about JVM Options:**

1)    Boolean JVM options can be  turned on with -XX:+ and can be turned off with -XX:-.

2)    Numeric JVM Options can be set with -XX:=. Numbers can include 'm' or 'M' for megabytes, 'k' or 'K' for kilobytes, and 'g' or 'G' for gigabytes (for example, 32k is the same as 32768).

3)    String JVM options can be set by using -XX:=, and usually used to specify a file, a path, or a list of commands.

**1) JVM memory options related to java heap size**

Following three JVM options are used to specify initial and max heap size and thread stack size while running Java programs.

**-Xms**        set initial Java heap size

**-Xmx**        set maximum Java heap size

**-Xss**>         set java thread stack size

**2) JVM option to print gc details**

**-verbose:gc** logs garbage collector runs and how long they're taking. I generally use this as my first tool to investigate if GC is a bottleneck for a given application.

**-XX:+PrintGCDetails** includes the data from -verbose:gc but also adds information about the size of the new generation and more accurate timings.

**-XX:-PrintGCTimeStamps**  Print timestamps at garbage collection.

**3) JVM parameters to specify Java Garbage collector**

**-XX:+UseParallelGC**      Use parallel garbage collection for scavenges

**-XX:-UseConcMarkSweepGC** Use concurrent mark-sweep collection for the old generation. (Introduced in 1.4.1)

**-XX:-UseSerialGC**        Use serial garbage collection. (Introduced in 5.0.)

**7) JVM options to change  Perm Gen Size**  
  
  
-XX:PermSize and MaxPermSize

-XX:NewRatio=2  Ratio of new/old generation sizes.

-XX:MaxPermSize=64m     Size of the Permanent Generation.

--------------------------------------------

<http://javarevisited.blogspot.com/2011/05/top-10-tips-on-logging-in-java.html>

**INFO** is more restricted than DEBUG java logging level and we should log messages which are informative purpose like Server has been started, Incoming messages, outgoing messages etc in **INFO level logging** in java.  
  
**WARN** is more restricted than INFO java logging level and **used to log warning sort of messages** e.g. Connection lost between client and server. Database connection lost, Socket reaching to its limit.  
  
**ERROR** is the more restricted java logging level than WARN and **used to log Errors and Exception**, you can also **setup alert on this java logging level** and alert monitoring team to react on this messages. ERROR is serious for logging in Java and you should always print it.  
  
**Another flexibility of log4j is that you can change logging level of your java application without restarting your java application**,   
  
-----------------------------------------------------------

<http://javarevisited.blogspot.com/2011/05/java-heap-space-memory-size-jvm.html>

<http://javarevisited.blogspot.com/2011/09/javalangoutofmemoryerror-permgen-space.html>

<http://javarevisited.blogspot.com/2011/08/enum-in-java-example-tutorial.html>