A **heap** is a specialized **tree-based data structure** that satisfies the heap property:

If A is a parent node of B then the key of node A is ordered with respect to the key of node B with the same ordering applying across the heap.

Heaps are crucial in the sorting algorithm heapsort.

**Heap Memory in Java**

* Java objects reside in an area called the heap.
* The heap is created when the JVM starts up and may increase or decrease in size while the application runs.
* Java Virtual Machine gets some memory from Operating System and uses this memory for all its need and part of this memory is call java heap memory.
* When the heap becomes full, garbage is collected. During the garbage collection, objects that are no longer used are cleared, thus making space for new objects.
* The heap is sometimes divided into two areas (or generations) called the **nursery (or young space) and the old space.** The nursery is a part of the heap reserved for allocation of new objects. When the nursery becomes full, garbage is collected by running a special young collection, where all objects that have lived long enough in the nursery are promoted (moved) to the old space, thus freeing up the nursery for more object allocation. When the old space becomes full garbage is collected there, a process called an old collection.
* Default size of Heap space in Java is 128MB
* Another way to find default heap size of JVM is to start an application with default heap parameters and monitor in using JConsole which is available on JDK 1.5 onwards, on VMSummary tab you will be able to see maximum heap size.
* If your application is large and lots of object created you can change size of heap space by using JVM options -Xms and -Xmx. Xms denotes starting size of Heap while -Xmx denotes maximum size of Heap in Java.
* PermGen Space of heap is used to store classes and Meta data about classes in Java.
* Increase heap size in JBOSS and Tomcat
* JBoss will use the JAVA\_OPTS environment variable to include additional JVM arguments, you could specify it in the /bin/run.conf.bat
* Add JAVA\_OPTS in your catalina.bat or Catalina.sh of bin folder

**Garbage Collection** is the process of freeing space in the heap or the nursery for allocation of new objects.

* The Mark and Sweep Model
* Dynamic and Static Garbage Collection Modes
* Generational Garbage Collection
* Compaction – fragmentation and defragmentation.

GC is a daemon thread. A daemon thread runs behind the application. It is started by JVM. The thread stops when all non-daemon threads stop.

What happens if an uncaught exception is thrown from during the execution of the finalize() method of an object?

The exception will be ignored and the garbage collection (finalization) of that object terminates.

JConsole from Windows command prompt

1. Mark and sweep
2. Scavenge

