**Garbage Collection in Java**

Garbage collection relieves programmers from the burden of freeing allocated memory.

Giving this job to the JVM has several advantages.

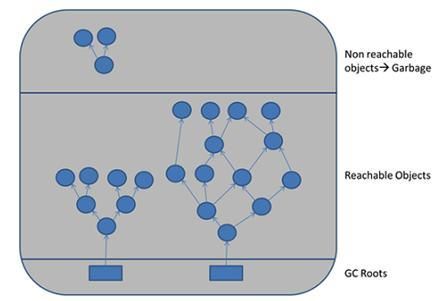
A garbage collector is responsible for

* Allocating memory
* Ensuring that any referenced objects remain in memory, and
* Recovering memory used by objects that are no longer reachable from references in executing code.

**Garbage-Collection Roots — The Source of All Object Trees**

Every object tree must have one or more root objects. As long as the application can reach those roots, the whole tree is reachable.

But when are those root objects considered reachable? Special objects called garbage-collection roots (GC roots; see Figure below) are always reachable and so is any object that has a garbage-collection root at its own root.



**Heap Generations for Garbage Collection in Java**

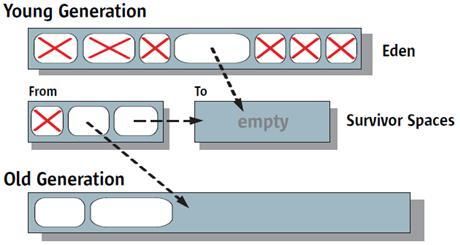
Java objects are created in Heap and Heap is divided into three parts or generations for sake of garbage collection in Java, these are called as

* **Young generation,**
* **Tenured or Old Generation**
* **Perm Area/** **Permanent generation of heap**.

Young/New Generation again 3 parts :

* eden - When an object first created in heap its gets created in new generation inside Eden space.
* Survivor 1 - And after subsequent Minor Garbage collection if object survives its gets moved to survivor 1.
* Survivor 2 - And then Survivor 2 before Major Garbage collection moved that object to Old or tenured generation.

Permanent generation of Heap or Perm Area of Heap is somewhat special and it is used to store **Meta data related to classes and method in JVM**, it also hosts **String pool** provided by JVM.



**Summary**

* Java Heap is divided into three generation for sake of garbage collection. These are young generation, tenured or old generation and Perm area.
* New objects are created into young generation and subsequently moved to old generation.
* String pool is created in Perm area of Heap, garbage collection can occur in perm space but depends upon JVM to JVM.
* Minor garbage collection is used to move object from Eden space to Survivor 1 and Survivor 2 space and Major collection is used to move object from young to tenured generation.
* Whenever Major garbage collection occurs application threads stops during that period which will reduce application’s performance and throughput.
* There are few performance improvement has been applied in garbage collection in java 6 and we usually use JRE 1.6.20 for running our application.
* JVM command line options –Xmx and -Xms is used to setup starting and max size for Java Heap. Ideal ratio of this parameter is either 1:1 or 1:1.5 based upon my experience for example you can have either both –Xmx and –Xms as 1GB or –Xms 1.2 GB and 1.8 GB.
* There is no manual way of doing garbage collection in Java