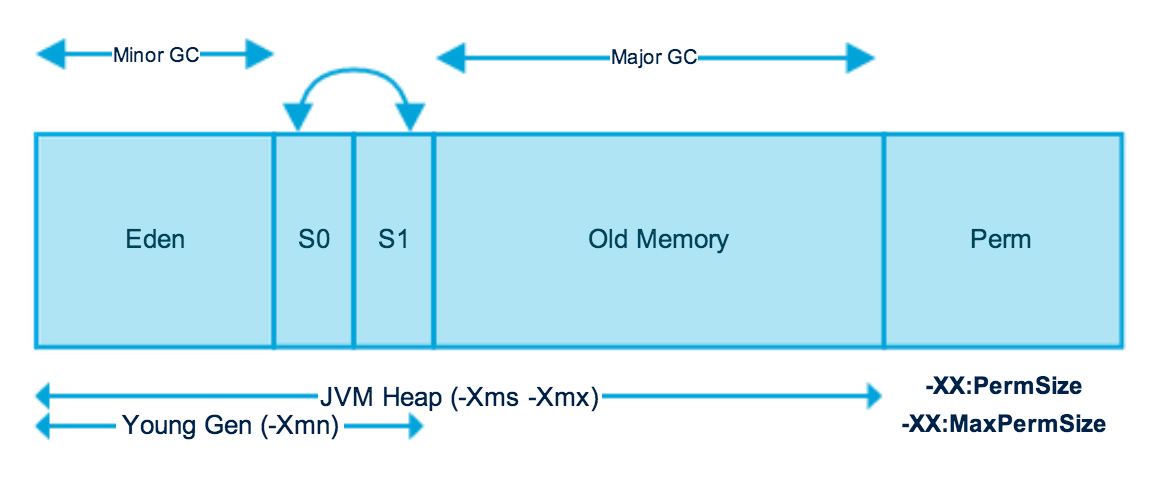
Java Memory Model

Heap



**Young Generation**: Where all new objects are allocated and aged. When the young generation fills up, this causes a minor garbage collection. A young generation full of dead objects is collected very quickly. Some surviving objects are aged and eventually move to the old generation.

* Eden Space : When object created using new keyword memory allocated on this space.
* Survivor Space : This is the pool which contains objects which have survived after java garbage collection from Eden space.

**Old Generation**:  This pool basically contains tenured and virtual (reserved) space and will be holding those objects which survived after garbage collection from Young Generation.

* Tenured Space: This memory pool contains objects which survived after multiple garbage collection means object which survived after garbage collection from Survivor space.

**Permanent Generation:** This memory pool as name also says contain permanent class metadata and descriptors information so PermGen space always reserved for classes and those that is tied to the classes for example static members

JVM divides memory into:

* Heap (objects)
* Stack (methods, local variables,reference variables)
* Code (byte code)
* Static (static data/ methods)

Heap is divided into:

* Old generation
* Young memory
  + Eden Space
  + Survivor Space 1
  + Survivor Space 2

Thread stack is temporary memory allocated related to methods and local variables within those methods. As these methods are executed, the local variables are terminated as well. When an object is created, it is stored in the heap.

Stack memory used for static memory allocation and the execution of a thread.

How to increase heap space:

* Xmx specify maxheapsite
* Xms to specify initial heap size
* Xss to set the stack size
* Xmxlg
* Xms64m