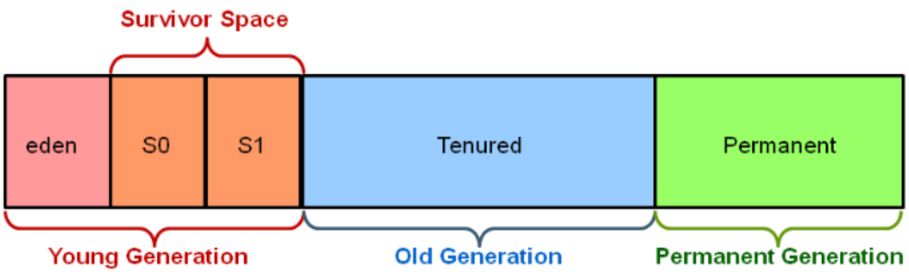
Garbage Collector and Heap area

Heap Area:



Permanent Generation --> Metaspace where class data is loaded, If too many classes are loaded we might get PermGen OutOfMemoryError
Solution:

1) Increase JVM Size

It is possible to increase heap size allocated by the JVM by using these command line options:

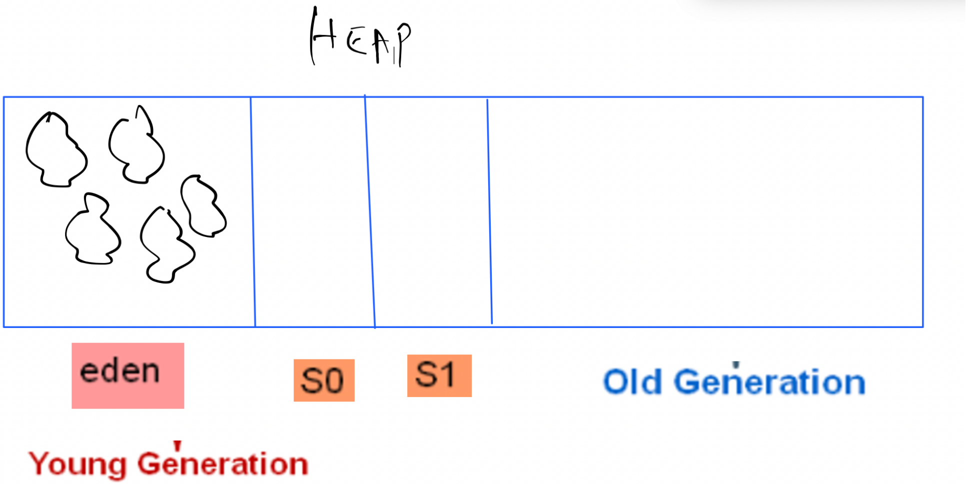
```
-Xms<size>       set initial Java heap size
-Xmx<size>       set maximum Java heap size
-Xss<size>       set java thread stack size
```

In the following example, minimum heap size is set to 16mb, and the maximum to 64mb:

```
java -Xms16m -Xmx64m ClassName
```

2) Also try to use Anonymous classes / lambda where ever possible

Memory for newly created objects is allocated in “EDEN” area / Young Generation

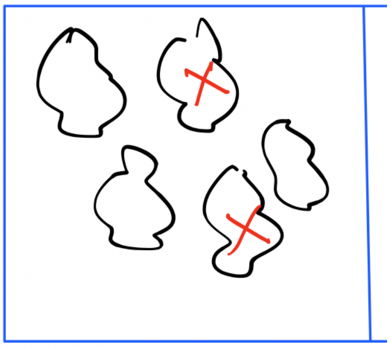


Garbage Collector is a low priority thread [Unit of Work] whose job is to clear unused memory

a) Short term GC / Scavenger
b) Long term GC

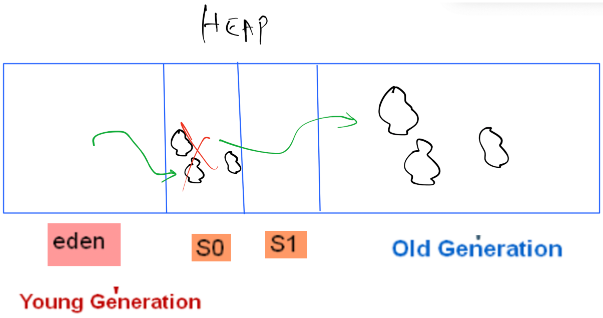
Short term GC

1) Mark objects which are not reachable / no references to the object

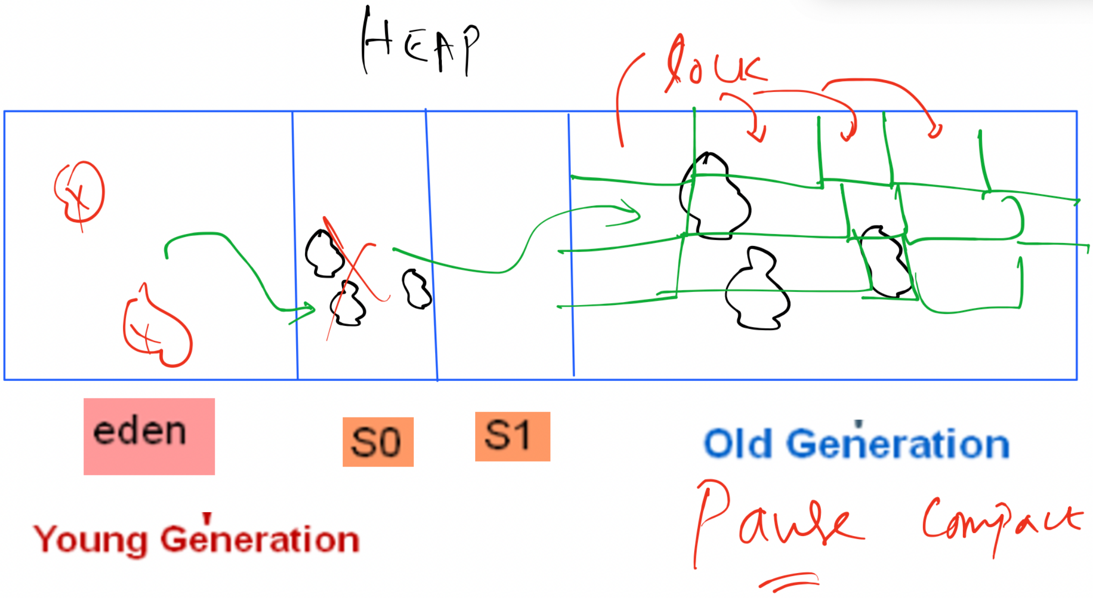


2) Clear them → release memory
3) After 3 cycles of GC [can be configured]

If objects survive 3 cycles of short term GC, they will be moved to Old Generation making use of S0 or S1



Long Term GC executes once after 3 cycles of Short term GC, this is responsible for releasing memory for old generation



Why we can't use Pointers?
Object doesn't reside in single memory.