**MEMORY MANAGEMENT:**

How is memory allocated 🡪with new keyword in heap

What is heap memory🡪main pool of memory in java app. Accessible to whole of the application.

What happens if there is no enouh memory to create objects?

JVM tries to claim some from heap by Garbage collection. Then it throws OUTofMEMORY error and jvm exists.

SECTIONS in MEMORY

Survivor space

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| eden | S1 | S2 | Old generation | Permanent generation or metaspace in java 8  Contains java.lang.instances |

Young generation

Not GC

Rarely garbage collected

Often garbage collected

<-------------------------------Heap------------------------------------->

Difference between Metaspace and PermGen

|  |  |
| --- | --- |
| Default increases size to the one offered by OS  Class may be loaded or unloaded | Has fixed size  Classes are loaded |

String Pool memory chnage after Java 8 🡪 String Pool moved from PermGen to Heap

What to do when you get OUTofMEMORY ?

1. Check if created unnecessary object and decrease its size
2. Increase heap memory

java Xms- 256n -Xmx 1g Main.java

java.lang.OutofMemoryError PermGen/Metaspace 🡪Increase size of meta

**STOP THE WORLD:**

During the GC process all the threads are stopped to reclaim memory.

During GC, promotions to different generations, grouping object together🡪Compaction

Incremental GC 🡪 perform GC in few phases and main program is executed between phases and sometimes during some phaces

concurrent G 🡪does not stop the main pgm at all.

**OBJECT FINALIZATION:**

Apart from GC clean up, dev should clean up used resources network connection and file handlers.

System.runFinalization () – request to JVM.

|  |  |
| --- | --- |
| Finalize | Finally |
| No control over when called as JVM decides when to execute it. So many resources may remain open | Finally can close them for sure. |
|  |  |

**CLASS LOADER:**

🡪when we compile java pgm it is converted to byte code which is platform independent and it is stored as .CLASS file on file system.

JVM loads the file into memory. Bringing byte code for class definition in memory of the running JVM is class loading.

3 types :

Bootstrap 🡪 loads JDK internal classes

Extension 🡪 jdk extension library

System 🡪 load classes from CLASSPATH