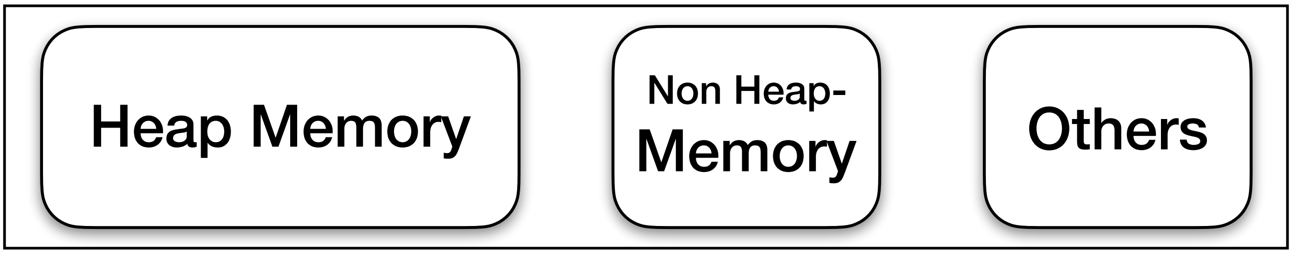
Java Memory Management

- Java virtual machine: Is a abstract computer machine that's enable computer to run program

1. Java (JVM) memory structure: It is divided into 3 parts



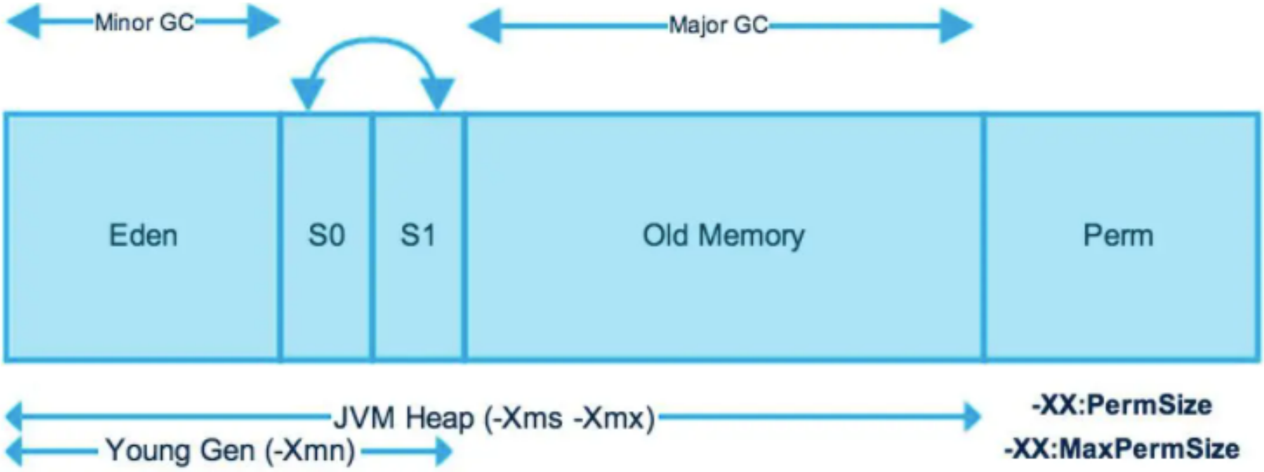
Heap Memory

Non Heap-Memory

Others

- Heap Memory: Its a runtime data area which store the memory of all the class instance and all arrays. It is created by JVM when the program runs and the size of the head can be increase or decrease . The size can be specified using **Xms VM** option and set using **Xmx** options. The heap can be fixed of variable size depending on garbage collection strategy . The default size of heap size is 64Mb
- Non heap Memory:It is created when the JVM Start-up . Its stores per-class structure such as runtime pool, method data , fields ,constructor as well as Internal String . The size can be change using -XX: MaxPermSize.
- Others: Jvm uses other code memory to store the JVM code itself.

2. Java (JVM) Heap Memory structure:



JVM heap is divide in to 2 parts new Generation(nursery) and old generation .

The nursery is a part of heap allocation for new objects. When the nursery is full , garbage is collected by young collection, and the objects lived long enough in the new generation are move to old generation and make space available for new objects. The Nursery is further divided into **2** types . **Edan** and **Survivor** Memory.

Key features of Nursery :

- The new object created are stored in Edam memory
- When Edam is fill . Small Garbage collection is performed and all survivor object are move to survivor space
- The Minor gc also check the object moved or not .
- * The object that are survived many GC cycle are moved to old Gen. Usually done the setting Threshold age .

When Old generation became fill the we perform GC and its call old collection (**MAJOR GC**) . It contain those object which lived long and survived many GC cycles.
The young generation free the amount of memory more fast the old GEN.

(In new Version the are is new are called as Keep area and it is reserved .It store most recent objects created and not perform garbage collection until next young GEN)

JAVA MEMORY MODEL

Permanent Generation(Prem Gen):Prem Gen contain application met data required by JVM to describe the class and method of the application. It is populated when the JVM a runtime based on class and method. Objects are Garbage collected in full Garbage collection.(Java Heap).Its a Fixed Size .

MetaSpace: There is no perm gen in java 8. Metaspace is not a part of heap .Allocation of the class metadata are now out of native memory. Metaspace can by default auto increase its size(depend on os). As long as the class is alive meta data remain alive in Metaspace and can’t be free .

Code Cache: The java code compile in a tiered manner. First tier, It uses client compiler (C1) in order to compile the code .The Profiling data is used in the second tier (C2) for the server compiler, to optimize the code in optimised manner.This this by default in java 8.

The Java-In-Time(JIT) compile store the compile code in the code cache. This is the special heap that hold the compile code. This is flushed when the the size exceeds a threshold.

Method Area: It's a perm Gen Part used to store the class Structure and code method and constructor.

Memory pool: It is created by JVM memory manager to create pool the immutable objects.Its a part of heap or Prem gen depend on JVM mamory manager.

Java Stack Memory:It is used for the execution of a thread. They contain method-specific value that are short live span and reference to other objects in haep.

Garbage Collection: Its a process of freeing space in the heap for the allocation of new objects. Its a backend program and java by default call the garbage collection .It find out all the unreferenced objects and are deleted and space in reclaimed for the allocation of new objects. Basic steps of GC:

- **Marking**: It identify all the object that are in use and which are not in use.
- **Normal Detection**: GC remove all the unreferenced objects and reclaims the free space.
- **Detection and compacting**: For better performance, after deletion all the surved objects are then moved to be together.

Mark and sweep Model of GC:

It perform the Gc for the whole heap .It consist of 2 Phases , the mark and the sweep phase.

During marking process it marks all the objects that are reachable from java thread, native handler and other root sourc and rest are considered as garbage .

During sweep phase , the head is traversed to find all the gaps between the objects. These gaps are marked in the free list and available to store objects.

Types of Gc:

- Serial GC
- Parallel GC
- Parallel old gc
- Concurrent marks and sweep collector
- G1 Garbage Collector