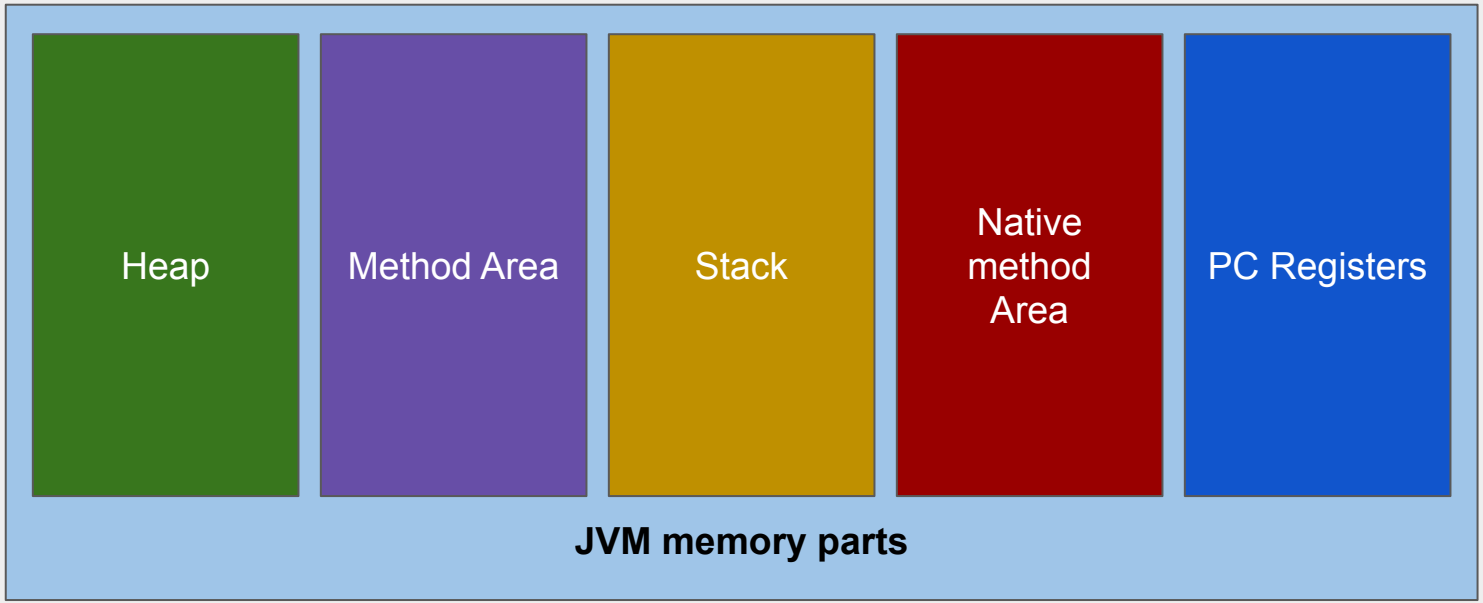
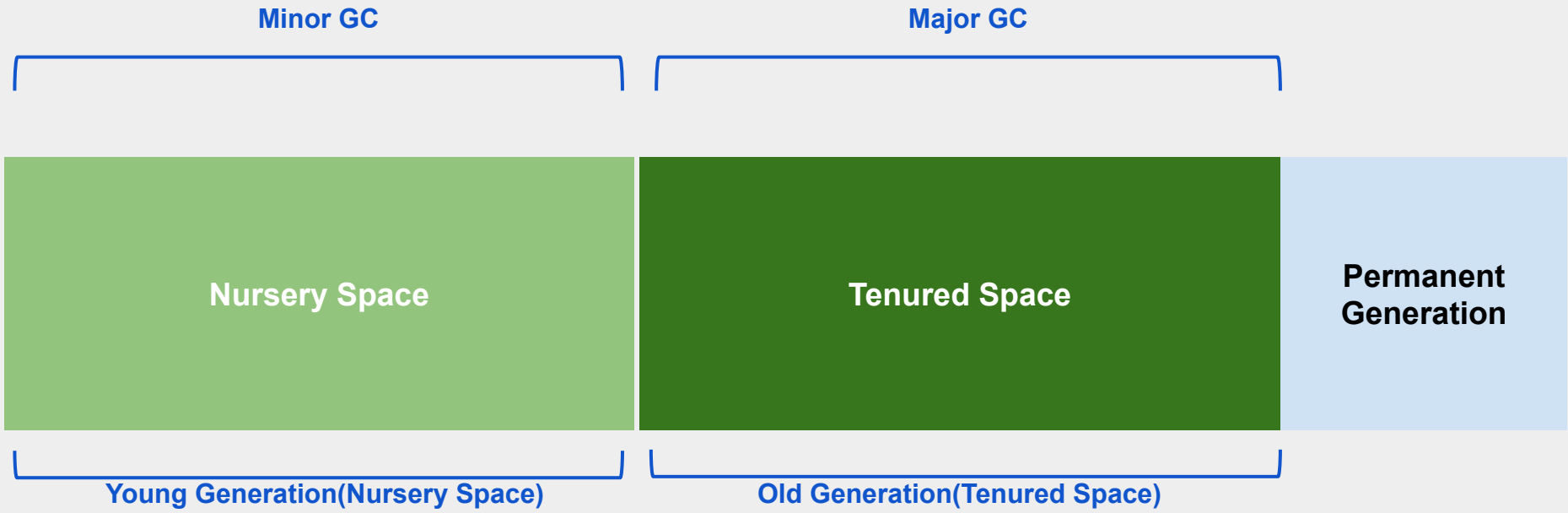
Heap Memory and Method Area

JVM Memory Structure

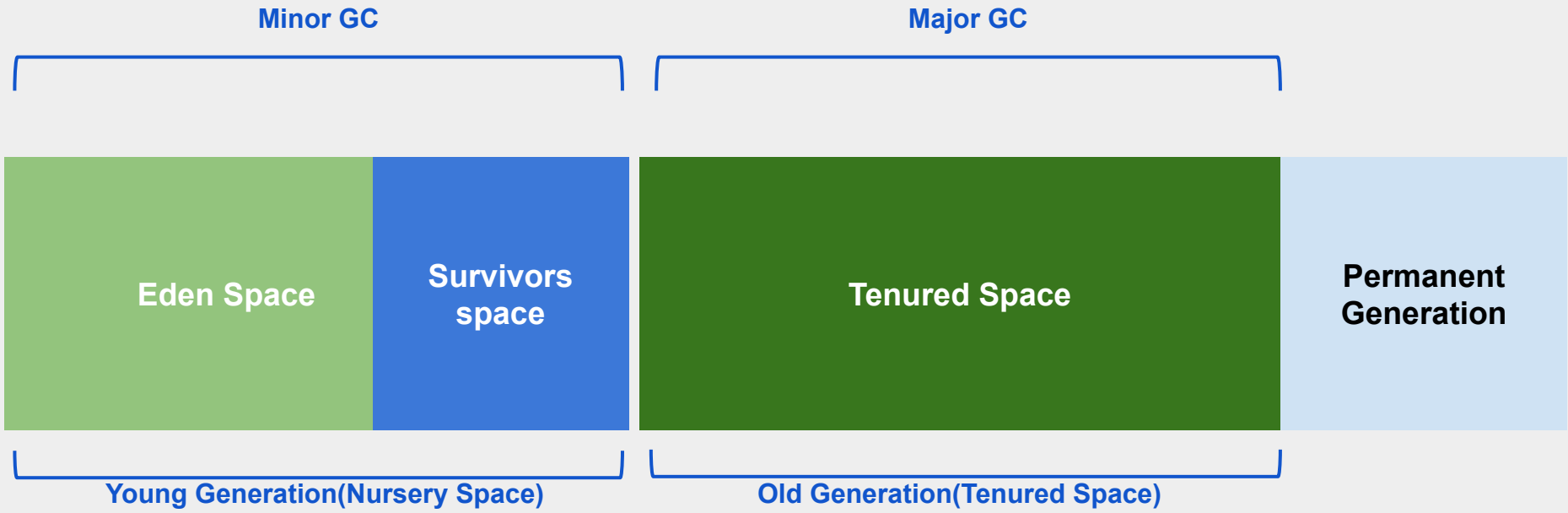


Note: The heap area is one of the most important memory areas of JVM. All the java objects are stored in heap area.

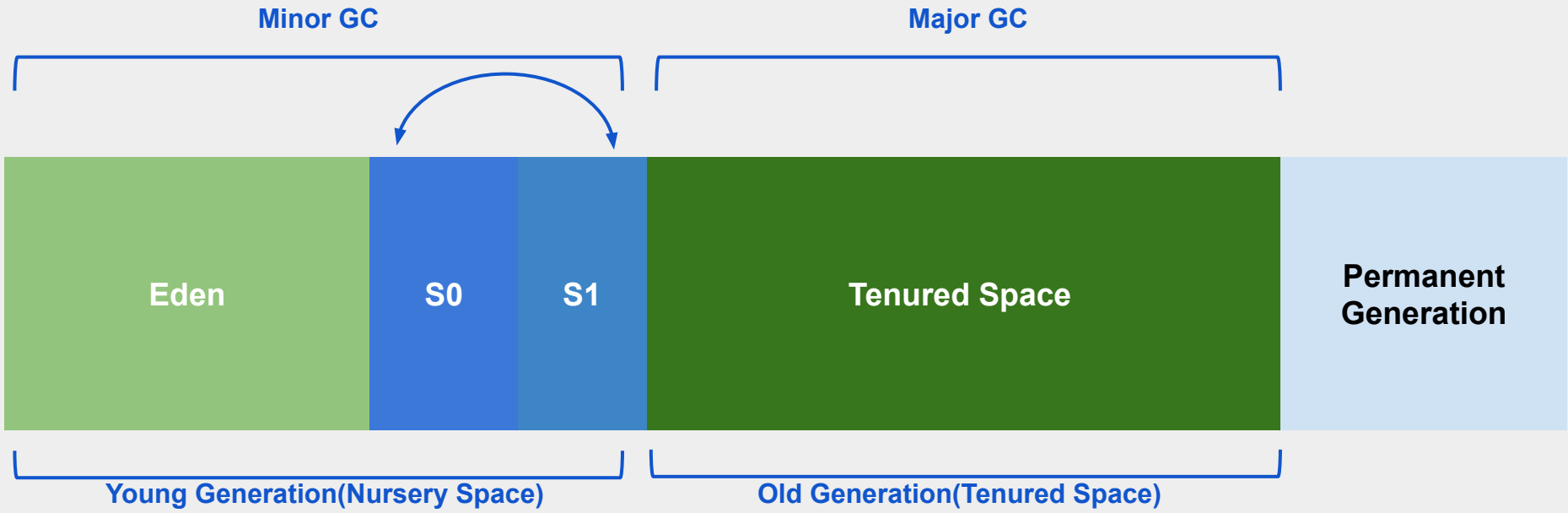
Heap Memory


Minor GC
Major GC
Nursery Space
Tenured Space
Permanent Generation
Young Generation(Nursery Space)
Old Generation(Tenured Space)

Heap Memory


Minor GC
Major GC
Eden Space
Survivors space
Tenured Space
Permanent Generation
Young Generation(Nursery Space)
Old Generation(Tenured Space)

Heap Memory


Minor GC
Major GC
Eden
S0
S1
Tenured Space
Permanent Generation
Young Generation(Nursery Space)
Old Generation(Tenured Space)

Nursery Space

The JVM heap is physically divided into two parts (or generations) nursery (or young space/young generation) and old space (or old generation). The nursery is a part of the heap reserved for the allocation of new objects. When the nursery becomes full, garbage is collected by running a special young collection, where all the objects that have lived long enough in the nursery are promoted (moved) to the old space, thus freeing up the nursery for more object allocation. This garbage collection is called Minor GC. The nursery is divided into three parts – Eden Memory and two Survivor Memory spaces.

Important points about the nursery space

- Most of the newly created objects are located in the Eden Memory space
- When Eden space is filled with objects, Minor GC is performed and all the survivor objects are moved to one of the survivor spaces
- Minor GC also checks the survivor objects and moves them to the other survivor space. So at a time, one of the survivor space is always empty
- Objects that have survived many cycles of GC, are moved to the old generation memory space. Usually, it is done by setting a threshold for the age of the nursery objects before they become eligible to promote to the old generation

Tenured Space

When the old generation becomes full, garbage is collected there and the process is called as old collection. Old generation memory contains the objects that are long-lived and survived after many rounds of Minor GC. Usually, garbage collection is performed in Old generation memory when it's full. Old generation garbage collection is called as Major GC and usually takes longer. The reasoning behind a nursery is that most objects are temporary and short-lived. A young collection is designed to be swift at finding newly allocated objects that are still alive and moving them away from the nursery. Typically, a young collection frees a given amount of memory much faster than an old collection or a garbage collection of a single-generational heap (a heap without a nursery).

Permanent Generation

Permanent Generation or **“Perm Gen”** contains the application metadata required by the JVM to describe the classes and methods used in the application. Perm Gen is populated by JVM at runtime based on the classes used by the application. Perm Gen also contains Java SE library classes and methods. Perm Gen objects are garbage collected in a full garbage collection.

Metaspace

With Java 8, there is no Perm Gen, which means there is no more “java.lang.OutOfMemoryError: PermGen” space problems. Unlike Perm Gen which resides in the Java heap, Metaspace is not part of the heap. Most allocations of the class metadata are now allocated out of native memory. Metaspace by default auto increases its size (up to what the underlying OS provides), while Perm Gen always has fixed maximum size. The theme behind the Metaspace is that the lifetime of classes and their metadata matches the lifetime of the classloaders. That is, as long as the classloader is alive, the metadata remains alive in the Metaspace and can’t be freed.

Method Area

Method Area is part of **meta space** and used to store class structure (runtime constants and static variables) and code for methods and constructors.

When Method Area garbage collected ?

Interview Questions