Heap memory

The heap memory is the runtime data area from which the Java VM allocates memory for all class instances and arrays. The heap may be of a fixed or variable size. The garbage collector is an automatic memory management system that reclaims heap memory for objects.

* **Eden Space**: The pool from which memory is initially allocated for most objects.
* **Survivor Space**: The pool containing objects that have survived the garbage collection of the Eden space.
* **Tenured Generation**: The pool containing objects that have existed for some time in the survivor space.

Non-heap memory

Non-heap memory includes a method area shared among all threads and memory required for the internal processing or optimization for the Java VM. It stores per-class structures such as a runtime constant pool, field and method data, and the code for methods and constructors. The method area is logically part of the heap but, depending on the implementation, a Java VM may not garbage collect or compact it. Like the heap memory, the method area may be of a fixed or variable size. The memory for the method area does not need to be contiguous.

* **Permanent Generation**: The pool containing all the reflective data of the virtual machine itself, such as class and method objects. With Java VMs that use class data sharing, this generation is divided into read-only and read-write areas.
* **Code Cache**: The HotSpot Java VM also includes a code cache, containing memory that is used for compilation and storage of native code.