**Stack & Heap:**

.NET frame work stores items in memory mainly in two locations as your code executes. i.e. stack memory and heap memory. Depends on the data type, variable is allocated to that type of memory.

Primitive data types(Value types) are not complex and this will be stored in stack. Value types are types which hold both data and memory on the same location. Stack is responsible for keeping track of the running memory needed for your application. Stack is responsible for keeping track of what’s executing in our code. In stack memory allocation and de-allocation is done at the top of the Stack using LIFO. It is self-maintaining .

Complex data types(Reference type) will be stored in Heap. When we have create a object. It creates a pointer on the stack and the actual object is stored in another memory location .That is called Heap. It doesn’t track of running memory. It’s just a collection of objects which can be reached at any time. It’s used for dynamic memory allocation. A reference type has a pointer which points to the memory location. Pointers are managed by the CLR. Heap is responsible for keeping track of our objects. Garbage Collection which deals with how to keep the Heap clean.

**Garbage Collection:**

Garbage Collector acts like a automatic memory manager. It allocates the objects efficiently in heap and enables us to develop a application without having to free memory and keeps the memory available for future allocations. Provides memory safety by making sure that an object cannot use the content of another object. During the process of garbage collection, it checks for dead objects and the objects which are no longer used then it adjusts the space for live objects and tries to free more memory. Allocates objects on the managed heap efficiently. It is managed by different Generations

Generation Zero: This generation holds short lived objects

Generation One: This generation is the buffer between short-lived and long lived objects.

Generation Two: This generation holds long lived objects like a static and global variable, that needs to be persisted for a certain amount of time.

There is no specific timings for GC to get triggered, GC automatically start operation. When virtual memory is running out of space and when allocated memory is suppressed acceptable threshold. We can also explicitly trigger garbage collector.