**Difference between Stack and Heap?**

|  |  |
| --- | --- |
| **Stack** | **Heap** |
| 1. It is used for static memory allocation. For example, values (int, float, double, object reference varaible) are stored in stack | 1. It is used for dynamic memory allocation. For example, reference values (class, interfaces and actual object values) are stored in heap |
| 2. Let’s say, Customer c1=new Customer(). c1.ID=1, c1.name=pal. In this example c1 (object reference variable) is stored in stack | 2. in the same example, 101 and pal (actual object values) are stored in heap. |
| 3. Values stored in stack are destroyed immediately after code is lost | 3. Whereas, reference values stored in heap are not destroyed immediately after the code is lost instead those will remain in memory until garbage collector removes them. |
| 4. In this values are stored in LIFO (Last In First Out) format. | 4. In this values are stored in FIFO(First In First Out) format. |

**What is Garbage Collection?**

The main use of Garbage Collection is to free memory for an application. In DOTNET application, when an object is created the Common Language Runtime Environment (CLR) allocates memory from the heap. Once the memory is full, the CLR will initiates the Garbage Collector which removes the objects which ae no longer use in the application.

**There are three ways to initiate Garbage Collection:**

1. **Default:** The GC will be initiated automatically by CLR.
2. **Forced:** In this, GC will be initiated forcefully when there is a need to free memory immediately.
3. **Optimized:** In this, GC will check whether there is a need to remove objects or not. If there is enough memory to allocate for newer objects GC will not be initiated. But if there is not enough memory, GC will be initiated immediately.