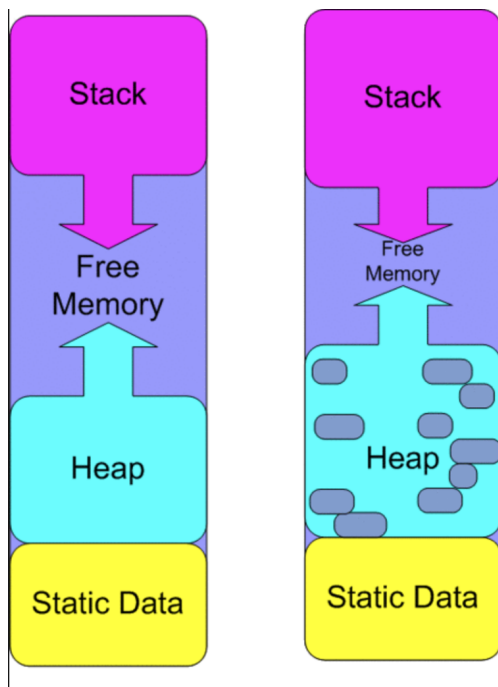


# [0402] The Stack vs The Heap

Programs have four "chunks" of memory available to them when they run - the Stack, the Heap, the Fixed Data area, and the Free Space area. It is important for programmers to understand how these memory areas work.



(These simplified diagrams show how memory is actually used in a typical C/C++ program. C#'s managed memory is laid out differently, however, from a conceptual standpoint, these diagrams are close enough for most programs.)

## Here are the Two Key Things to Remember:

1. Local variables and parameters are allocated from the Stack.
2. Objects and other things that are created by "new" are allocated from the Heap.

Other important things to know:

- The Heap is Garbage Collected.
- The Stack grows and shrinks automatically as new "scopes" are entered and exited.
- The Static Data area contains all of the strings and literals contained in the program. It never changes.
- The heap grows throughout the life of the program as objects are allocated. Freed objects are typically just left as "holes" in the heap until the program needs more memory. At that point, the garbage collector reorganizes the heap to eliminate the holes and create more contiguous free space.
- If the stack and heap grow large enough to use all of the free memory space - and garbage collection cannot reclaim more space from the heap - the program crashes because it is out of memory.

If you'd like a more complete description of things, this 4-part article explains memory usage in more detail:

<https://www.c-sharpcorner.com/article/C-Sharp-heaping-vs-stacking-in-net-part-i/> [\\_ \(https://www.c-sharpcorner.com/article/C-Sharp-heaping-vs-stacking-in-net-part-i/\)](https://www.c-sharpcorner.com/article/C-Sharp-heaping-vs-stacking-in-net-part-i/)