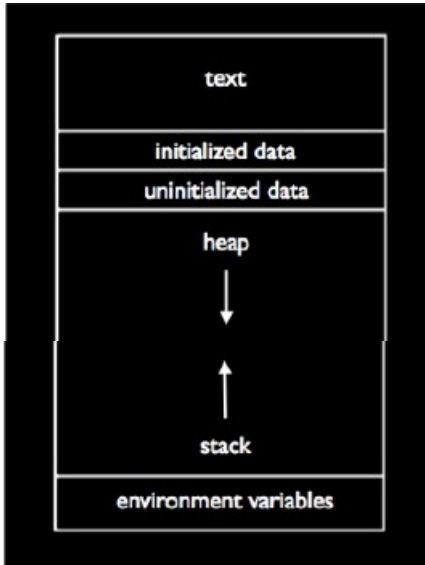
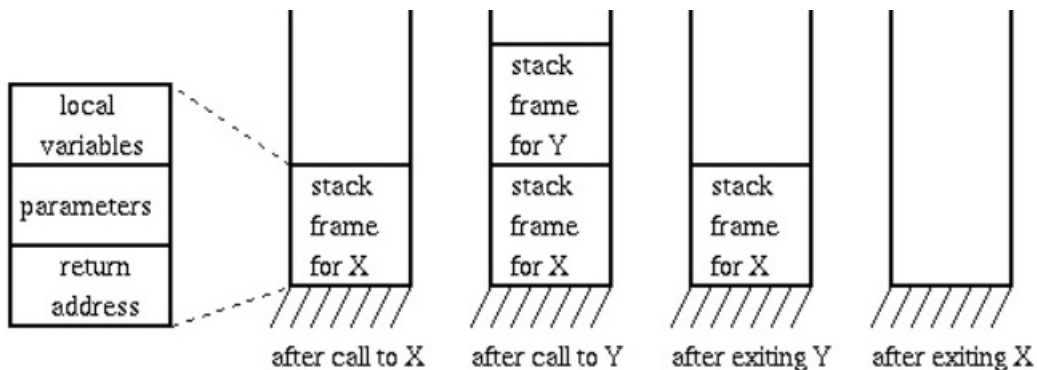


Runtime Stack vs. Memory Heap

When you run a C++ program, the memory you use is divided into the runtime stack and the heap. Each grow and shrink as you call/finish functions and allocate/deallocate memory.



Every time you call a function, an activation record (also called stack frame) is pushed on the stack. It holds memory used for parameters, local variables, address where to return, etc. When a function finishes, the stack frame is popped off the stack, releasing the memory for the next call. If you run out of stack memory, e.g., infinite recursion, you get a stack overflow error.



The memory heap is used when you do a "new" to allocate dynamic memory. When you use "delete" to deallocate the memory, it is freed up to be used again.

