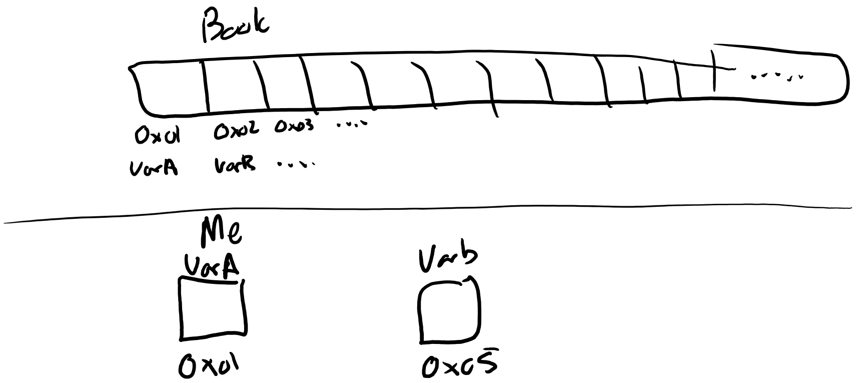
2019-11-12 Memory Addresses and Pointers

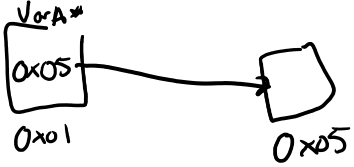
Tuesday, November 12, 2019 11:23 AM

17.3 Abstract representation of memory

1. How we draw memory on the board



- A pointer is just a variable that contains a memory address instead of an actual, usable value



- Basic pointers are useful when we need to indirectly modify a value that is not in our direct scope.

1. Creating basic pointers (w/o new) in C++

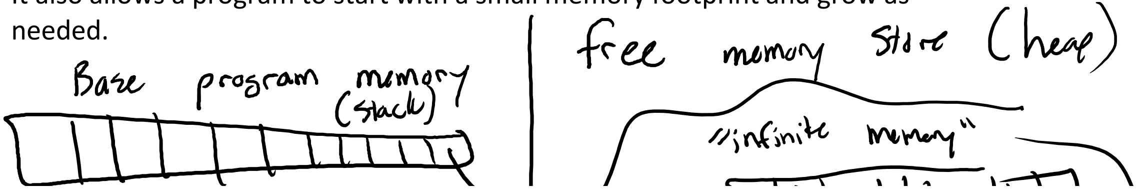
17.3.1

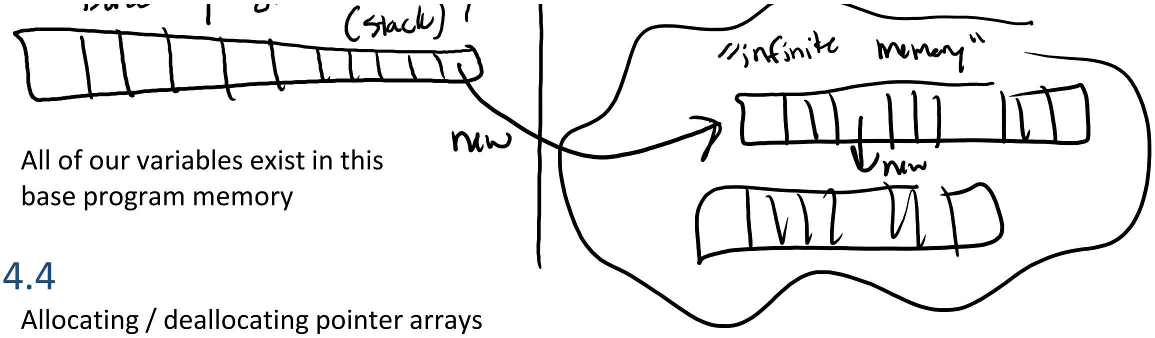
1. Sizeof command in C++

17.4.1

1. Free store allocation / deallocation using new/delete

- Real usage of pointers
- All programs inform the operating system of how much memory is required to initially start the program
    - In old days, the maximum memory allowed was 640k or lower
    - On the apple 2, this might have been 64k or less
    - The problem became: as the amount of RAM in a system increased, how do we allow programs to use more than that base level of memory?
- Using "new" allowed a program to use more than the max allowed memory.
- It also allows a program to start with a small memory footprint and grow as needed.





- All of our variables exist in this base program memory

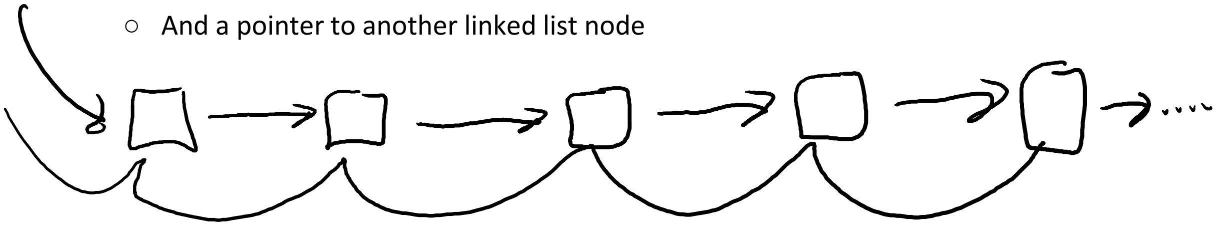
17.4.4

1. Allocating / deallocating pointer arrays

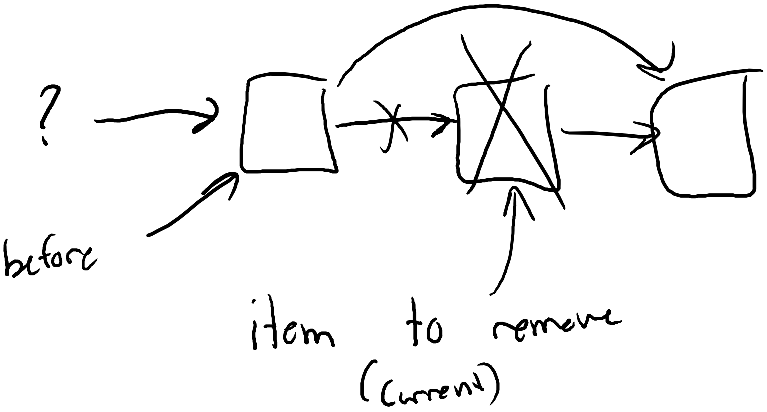
- The TRUE primary use of pointers
- Works exactly the same as new for single variables but uses array brackets

17.9.3 (Linked) Lists

- Just like a vector, a linked list is a continuously expanding collection of linear data
- In general, vectors are the better choice in real life
- Academia talks about linked lists because they require careful pointer management and some people really like doing that kind of thing.
    - Very important because most interviewers assume you've been hammered with linked list knowledge and thus will frequently ask you questions about linked lists
- The simplest linked list is an object that contains two bits of information:
    - A value
    - And a pointer to another linked list node



Removing an item from a linked list


?
before
item to remove
(current)