

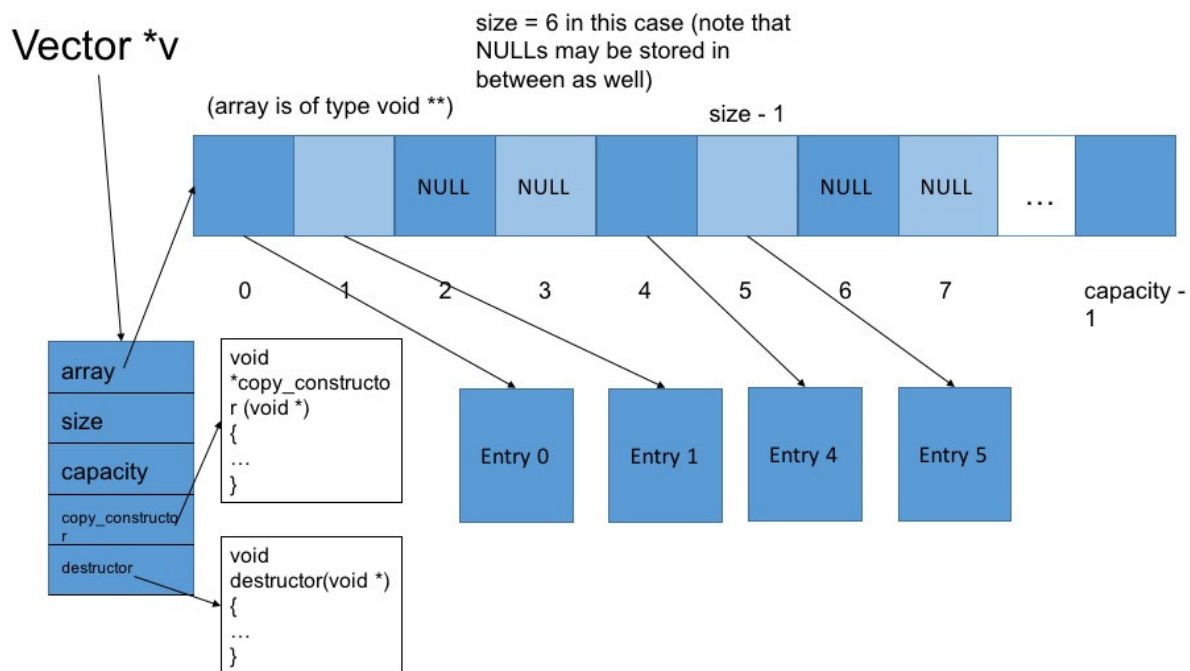
CS214: Recitation 2 : Vector

A vector is an array that grows and shrinks as a user adds and removes items from it. This is very similar to the Java ArrayList. In this recitation, you will be implementing your own vector - including constructors/destructors for the data structure.

Your implementation should go in `vector.c`, which is the only file that will be sent to your team lead for review. As an intern looking to become a full-time employee, you should create test cases in `vector_test.c` to show you are a responsible programmer. Your mentor has left notes in `vector.h` guiding your implementation.

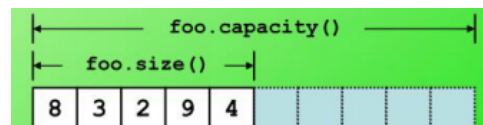
Since this vector is generic, it will have to call custom constructor and destructor functions when objects are added or removed. (How is this better than a single function which handles all possible types?) Thus, your vector structure will contain pointers to your constructor or destructor routines, and you can initialize the vector by passing pointers to these functions as parameters.

Here's an illustration:



What you'll end up with is a useful general-purpose vector, capable of dynamically expanding and shrinking. (No more fixed-sized buffers, guys!)

1. **vector is array-based (not node-based) sequence container**
2. **`size()`, `resize(n, value)`**
3. **`capacity()` and `reserve(n)`**



4. **if `size() > capacity()`, allocate new memory of `2*capacity()`, copy the original content, and delete the old memory**
5. **`void*` pointer**