



CMSC 106

Lecture Set #12 – Dynamically Allocated Arrays

Set Started:
Wednesday, November 30, 2011



Pointers and Arrays

- Name of an Array = Address of 0th element
- Can use pointers to indicate other elements of the array
- Can use pointer arithmetic to traverse an array
- When an array is passed as the argument to a function – only the address indicated is passed



Statically allocated array

- `char name[] = "Jandelyn";`
- `int sizes[5] = {4,6,2,4,8};`
- `double scores[10];`
- Name of array is a constant pointer

~~`scores++;`~~ ~~`name+=2;`~~ `*(name+2) = 'z';`



Filling statically allocated array

- Size of array must be known at COMPILE time
- Always stays the same size (can use only a portion of the allocated space if you wish)
- Must make sure you stay "in bounds"



Dynamically allocating space

- calloc
 - First argument must indicate the number of items you want in the array
 - Second argument must indicate the size of one of those array items
 - Allocates a single sequential and contiguous block of memory of the size needed
 - Returns NULL if the request can't be fulfilled
 - Returns the Address of that space if it can be



Rules

- Space is allocated from a place called the heap
- Once allocated, the size is fixed (can't change size and remain at that location in memory)
- The space is sequential and contiguous so can still use array index or pointer method (exactly the same as a statically allocated array)
- The name of the Array is NOT CONSTANT – but be careful not to lose the location of the beginning of the array
- Once allocated and you are done using it, the space must be freed
