

# Allocating Memory

Adam Montgomery

2/16/19

## Why do we need to allocate memory

Arrays are useful tools in C programming however, their size needs to be specified in their declaration. This is an example of a time where dynamic memory allocation could be useful, where variables or objects can be created at runtime, not compile time.

## Dynamic Programming

Dynamic programming is programming where the runtime can change. Instead of using recursion to solve problems, we can program linearly while dynamically handling inputs.

## Stack vs. Heap

## Functions to use

**malloc()** m(emory) alloc(ation) takes a size parameter, and returns a **void** pointer  
**calloc()** c(ontiguous) alloc(ation) takes a number of elements, and size. Returns a **void** pointer  
**realloc()** re alloc(ation) takes a pointer, and a new size. Returns a **void** pointer.  
**free()** anytime we use these functions, we need to tell the system when we are done with the memory using **free()**.

## Examples

```
int *p = malloc(sizeof(int));
int *a = calloc(10, sizeof(int));
//now we need more space in a
a = realloc(a, 12 * sizeof(int));

//don't forget to free the memory
free(p);
free(a);
```