Weeks 5-6
- Pointers and Arrays
  - Basic pointer type
  - Pointers and Arrays
  - Address arithmetic
  - Pointer Arrays
- User-defined data types
  - Structures
  - Unions
  - Pointers and arrays of user-defined data types
- Next two weeks
  - Memory allocation and dynamic structures
  - Complex structures with pointers, structures, etc.
  - Pointers functions

Basic Pointer
- Variable, memory storage and addresses

Basic Pointer -- continued
- Pointers and references

Parameter Passing with Pointers
- Still Pass-by-value with pointers
- However the memory content can change
  - Implications:
    - Be aware of the argument type when passing parameters
    - Be aware of the content is changing
    - When you attempt to change pointer argument values, ask yourself twice, typically a no-no.

Parameter Passing -- Example

Pointers and Arrays
- Essentially, pointers and array identifier are memory addresses.
- Array name, in particular, is the address of the first element
Address Arithmetic

- Pointers can have arithmetic operations!
  
  ```c
  int *ip1, *ip2, a[8];
  ip1 = &a[7];
  ip2 = &a[0];
  
  Then:
  ip2 - ip1 + 1?
  ip1 == ip2 ?
  ip1 == NULL?
  ip1 > ip2 ?
  ```

Pointer Arrays

- Comparisons with multi-dimensional arrays
  - A typical usage, command line arguments
    
    ```c
    int *ip[3], a[3][4], i=0;
    ip[0] = &a[0][0];
    ip[1] = &a[1][0];
    ip[2] = &a[2][0];
    ```

Sample Usage

-- command-line arguments

```c
int main (int argc, char * argv[]) {
    int i=0;
    while (i++ < argc) {
        printf("%s
", argv[i]);
    }
}
```