**Pointers and Dynamic Memory**

* **Pointers**
  + Can change what it points to
  + Can be used as an array
  + Doesn't need to initialized
  + Points to a variable
  + References to a variable
  + Passing pointers passes the value of the objects
* **References**
  + Can't change what it references to
  + initialized first
  + Passing objects make deep copy of the value of the object.
* **Arrow Operator**
  + Takes an object and used it to point to a method name
  + Points to data members or methods of that value
* **New**
  + Create new objects on the heap
* **Delete**
  + Delete objects and data from the heap
* **Heap**
  + Not structured, and cannot predict where the values will go
  + Programmers will free memory from the heap
  + Values on the heap doesn't get deleted, only the pointers on the stack gets deleted
  + Memory leaks occur when removing the pointers from the stack without also removing values from the heap, such as out of scope stack popping.
  + Dynamic memory
  + Create and delete memory when they're done
  + Variables on the heap can't predict where it is, not on the stack.