Assignment

- Assignment: \[ A = B; \]
  - A must be a variable (or array element like array[i])
  - B can be a complicated expression with multiple operations
  - What happens: B is evaluated/computed, the value of B is copied into A:
    - \[ A \leftarrow \text{value of } B \]

Data Types

- For most operators, types of arguments must be compatible.

```cpp
bool array[100];
array[i] = ' ';  // What is wrong with the two examples above?
if (array[i] == NULL) ...  
```

Loop Processing

- Problem: write a function that will return true if all of the elements in an array are equal to 0.
- How to think about this problem:
  - ALL of them must be 0 to be true. I have to look at ALL of them before I can return true.
  - If any one of them is not 0, it is false. I need ONE bad example to return false.
Binary operations on classes

- Define operator== over a class AAA with member variables x, y, and z.
- It’s defined as a member function. It only takes one parameter for the other class (call it “that”).
- You must use x, y, and z in the function as the values for the object on the left hand side of the operator.

```cpp
bool operator==(AAA that) {
    return (x == that.x &&
            y == that.y &&
            z == that.z);
}
```

Or:

```cpp
bool operator==(AAA that) {
    return (this->x == that.x &&
            this->y == that.y &&
            this->z == that.z);
}
```

Practice Problem #1

- Write a small section of code that computes the maximum value in an integer array a[] of size N.

Practice Problems #2

- Write a function RemoveFirst() that removes the first occurrence of a given value x from an array a[] of size N. It is not known whether the value actually occurs in the array. For example, if a = { 2,4,5,6,4,7,2,3,4,2} then RemoveFirst( a , 4 ) produces a = {2,5,6,4,7,2,3,4,2}
  The interface for the function is:

```cpp
void RemoveFirst( int a[], int & N, int x )
//Removes first x from array a[], decrements // N if x is removed
```
Practice Problems #3

- Write a function RemoveLast() that removes the last occurrence of a given value x from a singly linked list. It is not known if the value is actually in the list. For example the RemoveLast(L, 5) applied to the list L: 3,5,4,2,5,7 modifies the list to be L: 3,5,4,2,7. Assume the declarations:

```c
struct node {
    int data
    node *link;
};
void RemoveLast( node* &L, int x);
// Removes last occurrence of x from L
```