Module 27: Arrays and Functions

Intro to Computer Science 1 - C++
Professor Scott Frees

Textbook

The following is covered in section 7.5 and 7.6 in the text

Arrays and Functions

We can write functions to accept and array as input (as a parameter)

```
void function(int x[], int count)
```

Typically we'll pass the count in, so we can walk through the array.

Pass by reference

Arrays may be big - so we always pass by reference

No special syntax - they are automatically passed by reference

Changing arrays in functions

If you edit elements in a function, those changes are reflected outside the function

If you *add or remove* elements - you probably need to pass `count` in by reference too!

You might also need to pass the *capacity* into the function.

Don’t change?

- What is we want to make sure the array is **not** changed?
- There is no way to stop arrays from being passed by reference, but we can mark parameters as “const”

```
void function(const int x[], int count)
```

It is now a compiler error for “function” to edit x in any way!

Returning arrays?

You **cannot** return arrays from functions.

However - remember you can pass an uninitialized array in as a parameter - and initialize it **inside** the function.

Since arrays are pass by reference, this produces the same effect.

Programming Example 31

If you have two lists, what makes them equal?

- Same numbers, same order?
- Same numbers, any order?

Exercise: Write a program that reads two lists

- Print out if they are exactly the same
- Print out if they are *equivalent*.
- For each number in the first list, make sure it appears the same number of times in the second list...