CSC209 Lecture 2: Arrays and Pointers

David Liu, Department of Computer Science, University of Toronto

*Navigation tip for web slides: press **?** to see keyboard navigation controls.*

Loose ends from last class

chmod and numeric modes

```
$ chmod 750 hello
```

What does the mode `750` mean?

	Owner user	**Group**	**Other users**
	`7`	`5`	`0`
Binary representation	`111`	`101`	`000`
Permissions	`rwx`	`r-x`	`---`

Compiling C programs

```
$ gcc -Wall -g -std=gnu99 -o hello hello.c
```

- `gcc` is the **compiler** program
- `hello.c` is the **source code** (being compiled)
- command-line arguments to `gcc`:

Argument	Description
`-Wall`	report (almost all) compiler warnings
`-g`	used for debugging, more later on
`-std=gnu99`	specify version of C being used
`-o hello`	specify the name of the executable generated

Arrays and Pointers!

- An **array** is a contiguously allocated set of objects of a fixed type.
  - Example: `int nums[4];`
  - Warning: unlike many other programming languages, C arrays do not "store" their length.
- A **pointer** is an object whose value provides a reference (or memory address) to an object of a different type.
  - Example: `int *num_pt;`
- Two key operators:
  - `*` (dereference). When `x` is a pointer to type `T`, `*x` evaluates to the object of type `T` referenced by `x`.
  - `&` (address of). When `x` has type `T`, `&x` evalutes to the object of type "pointer to `T`" containing the memory location of `x`.