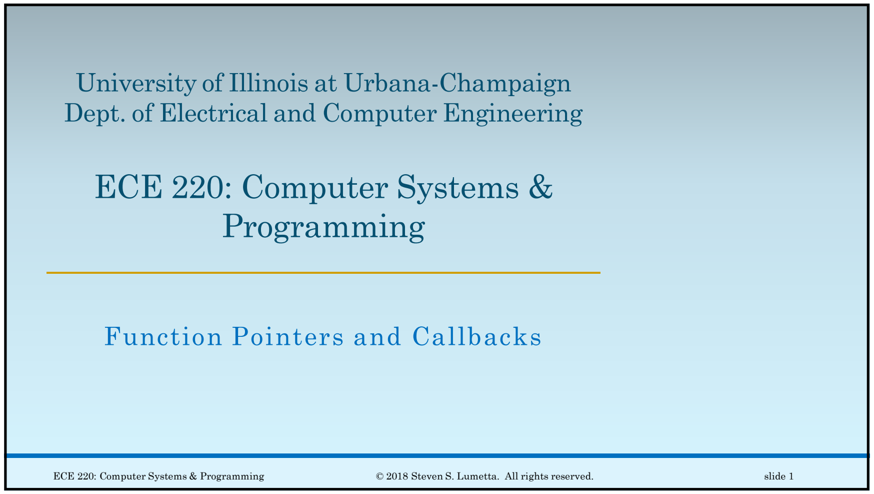
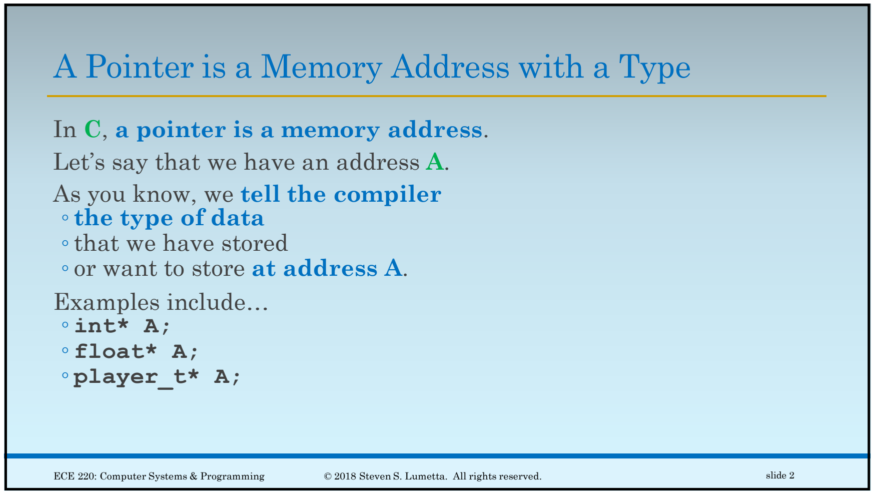


University of Illinois at Urbana-Champaign
Dept. of Electrical and Computer Engineering

ECE 220: Computer Systems & Programming

Function Pointers and Callbacks

1



A Pointer is a Memory Address with a Type

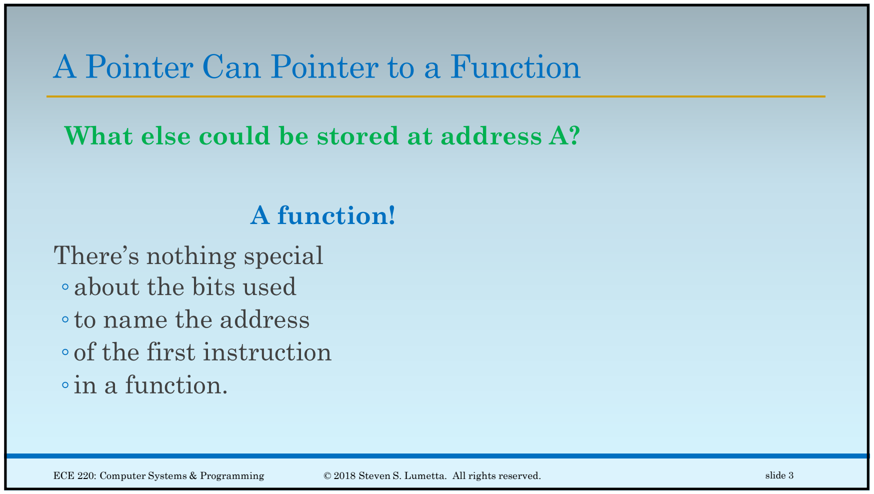
In **C**, **a pointer is a memory address**.

Let's say that we have an address **A**.

As you know, we **tell the compiler**
- **the type of data**
- that we have stored
- or want to store **at address A**.

Examples include…
- `int* A;`
- `float* A;`
- `player_t* A;`

2



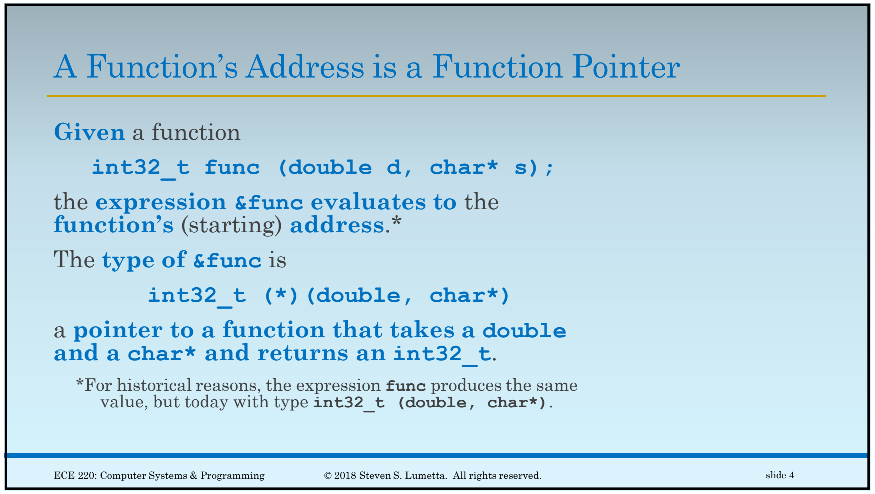
A Pointer Can Pointer to a Function

What else could be stored at address A?

A function!

There's nothing special
- about the bits used
- to name the address
- of the first instruction
- in a function.

3



A Function's Address is a Function Pointer

Given a function

```
int32_t func (double d, char* s);
```

the **expression `&func` evaluates to** the **function's** (starting) **address**.*

The **type of `&func`** is

```
int32_t (*)(double, char*)
```

a **pointer to a function that takes a double and a `char*` and returns an `int32_t`**.

*For historical reasons, the expression `func` produces the same value, but today with type `int32_t (double, char*)`.

4