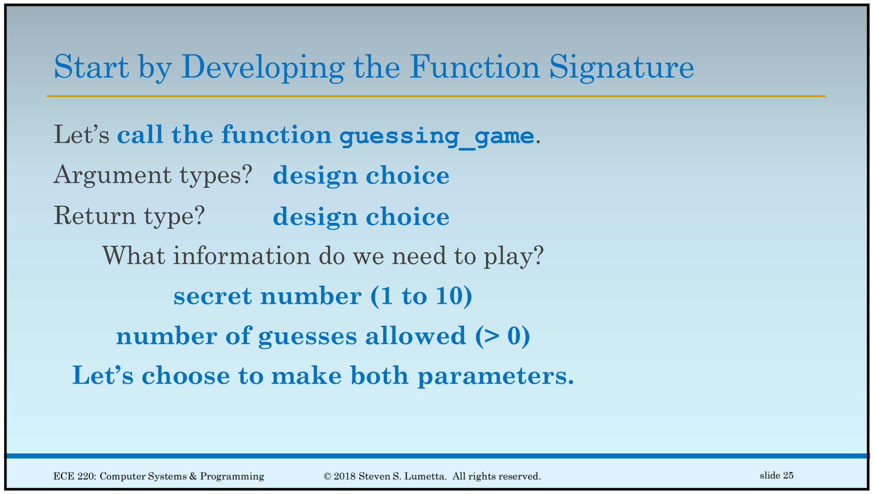


Start by Developing the Function Signature

Let's **call the function `guessing_game`**.

Argument types? **design choice**

Return type? **design choice**

What information do we need to play?

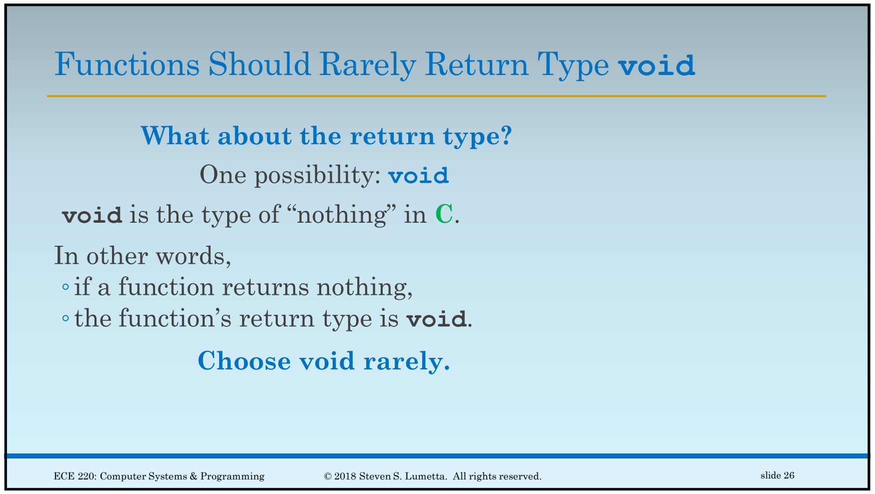
secret number (1 to 10)

number of guesses allowed (> 0)

Let's choose to make both parameters.

ECE 220: Computer Systems & Programming © 2018 Steven S. Lumetta. All rights reserved. slide 25

25



Functions Should Rarely Return Type `void`

What about the return type?

One possibility: **`void`**

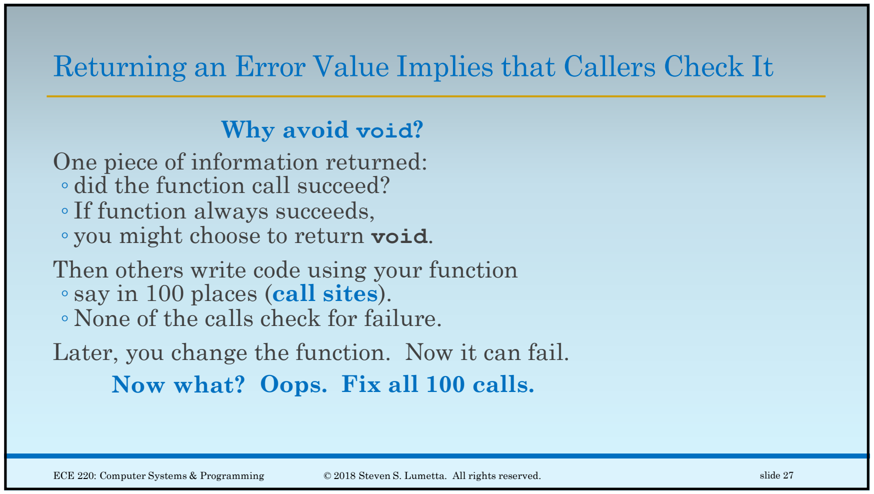
`void` is the type of "nothing" in **C**.

In other words,
- if a function returns nothing,
- the function's return type is `void`.

Choose void rarely.

ECE 220: Computer Systems & Programming © 2018 Steven S. Lumetta. All rights reserved. slide 26

26



Returning an Error Value Implies that Callers Check It

Why avoid `void`?

One piece of information returned:
- did the function call succeed?
- If function always succeeds,
- you might choose to return `void`.

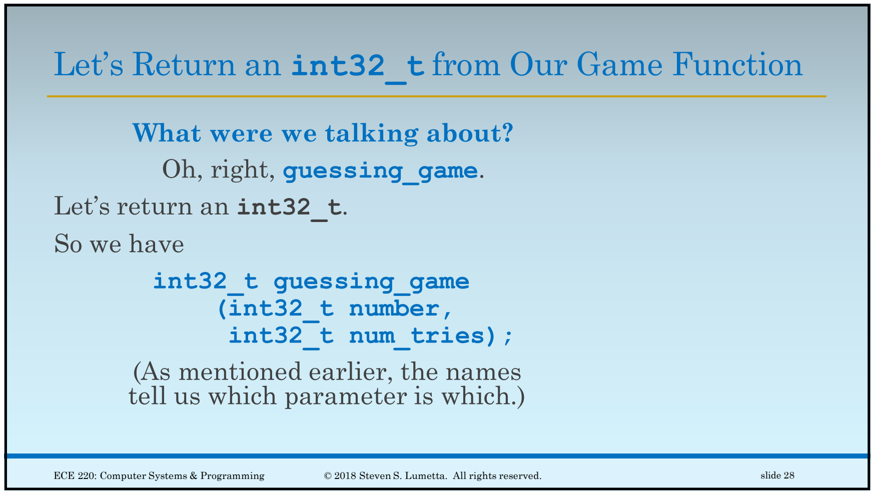
Then others write code using your function
- say in 100 places (**call sites**).
- None of the calls check for failure.

Later, you change the function. Now it can fail.

Now what? Oops. Fix all 100 calls.

ECE 220: Computer Systems & Programming © 2018 Steven S. Lumetta. All rights reserved. slide 27

27



Let's Return an `int32_t` from Our Game Function

What were we talking about?

Oh, right, **`guessing_game`**.

Let's return an `int32_t`.

So we have

```
int32_t guessing_game
     (int32_t number,
      int32_t num_tries);
```

(As mentioned earlier, the names tell us which parameter is which.)

ECE 220: Computer Systems & Programming © 2018 Steven S. Lumetta. All rights reserved. slide 28

28