Never Look Up Precedence Rules!

Another task for you:

Evaluate the C expression: `10 / 2 / 3`

Did you get 1.67?

Is it a friend's birthday?

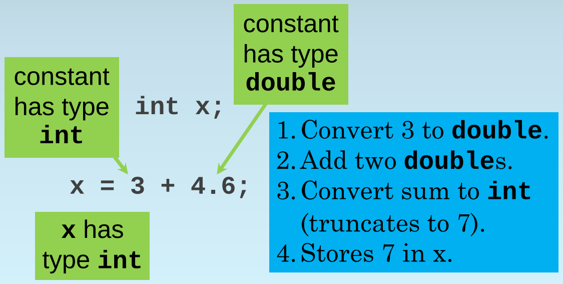
Perhaps it causes a divide-by-0 error?

Or maybe it's … 1? (10 / 2) / 3, as **`int`**

If the order is not obvious,
- Do NOT look it up.
- **Add parentheses**!

Compiler Silently Auto-Converts … Sometimes

What does this code do?



```
int x;
x = 3 + 4.6;
```

1. Convert 3 to **`double`**.
2. Add two **`double`**s.
3. Convert sum to **`int`** (truncates to 7).
4. Stores 7 in x.

Be Careful with Auto-Conversion

How does auto-conversion work?

When there's a choice, into the "larger" type.

What does that mean? Nothing obvious.

Integers convert to floating-point.

```
unsigned a = 10;
int b = -20;
if (a + b < 0) {
    printf ("ok");
}
```

What does the code to the left print?

Nothing.
As you'd expect?

Be Careful with Auto-Conversion

Auto-conversion happens silently: no errors, and no warnings.

For anything unclear (anything with a choice), avoid auto-conversion, or use explicit conversions (example to right).

```
unsigned a = 10;
int b = -20;
if (((int)a) + b < 0) {
    printf ("ok");
}
```