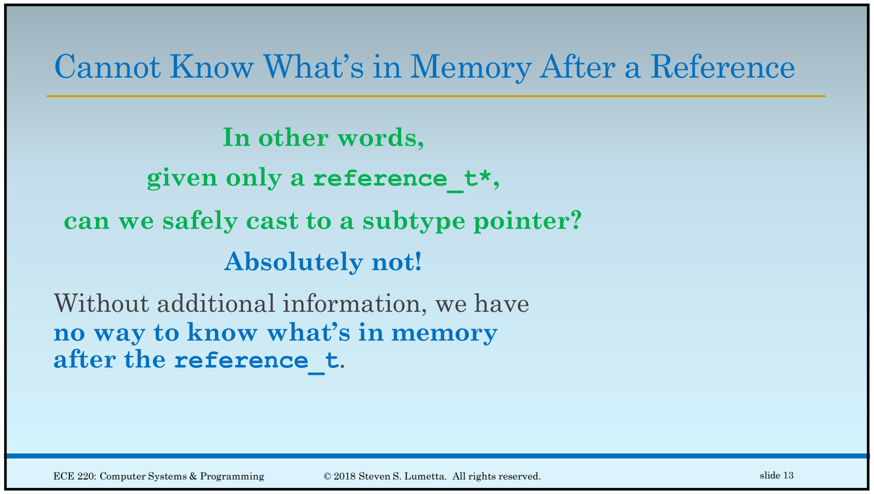


Cannot Know What's in Memory After a Reference

In other words,

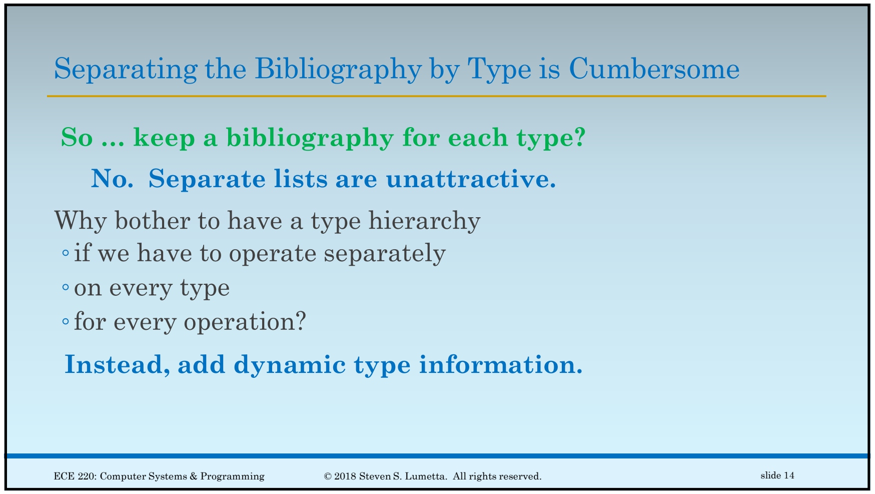
given only a `reference_t*`,

can we safely cast to a subtype pointer?

Absolutely not!

Without additional information, we have **no way to know what's in memory after the `reference_t`.**

13



Separating the Bibliography by Type is Cumbersome

So … keep a bibliography for each type?

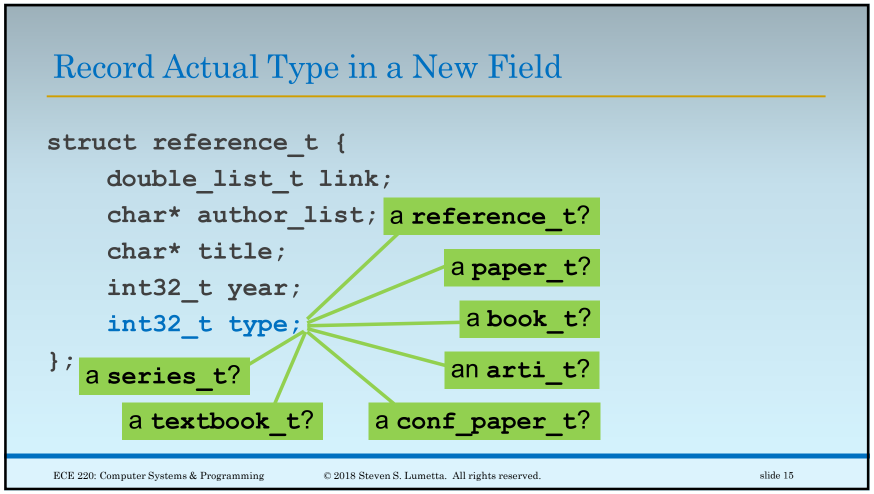
No. Separate lists are unattractive.

Why bother to have a type hierarchy

- if we have to operate separately
- on every type
- for every operation?

Instead, add dynamic type information.

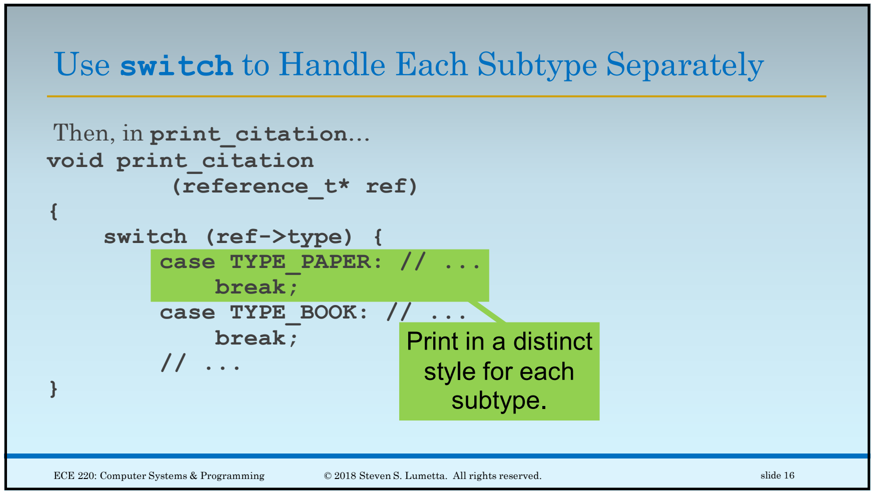
14



Record Actual Type in a New Field

```
struct reference_t {
    double_list_t link;
    char* author_list;
    char* title;
    int32_t year;
    int32_t type;
};
```

15



Use `switch` to Handle Each Subtype Separately

Then, in `print_citation`…

```
void print_citation
        (reference_t* ref)
{
    switch (ref->type) {
        case TYPE_PAPER: // ...
            break;
        case TYPE_BOOK: // ...
            break;
        // ...
}
```

Print in a distinct style for each subtype.

16