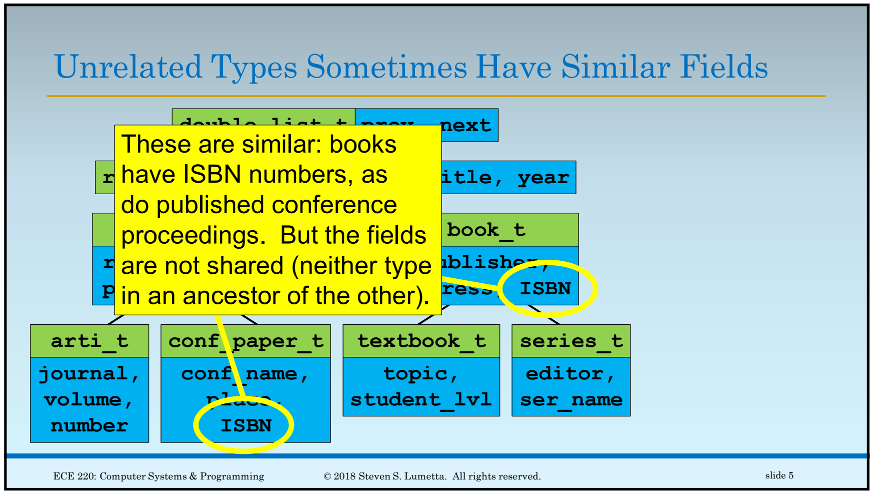


Unrelated Types Sometimes Have Similar Fields

double_list_t prev, next

r itle, year

book_t

r ublisher,

p ress, ISBN

arti_t

journal, volume, number

conf_paper_t

conf_name, place, ISBN

textbook_t

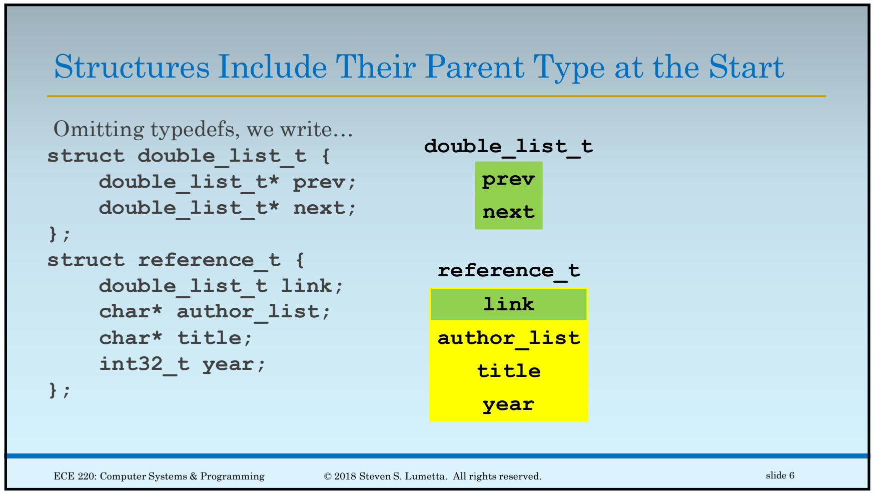
topic, student_lvl

series_t

editor, ser_name

These are similar: books have ISBN numbers, as do published conference proceedings. But the fields are not shared (neither type in an ancestor of the other).

5



Structures Include Their Parent Type at the Start

Omitting typedefs, we write…

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

double_list_t

prev

next

reference_t

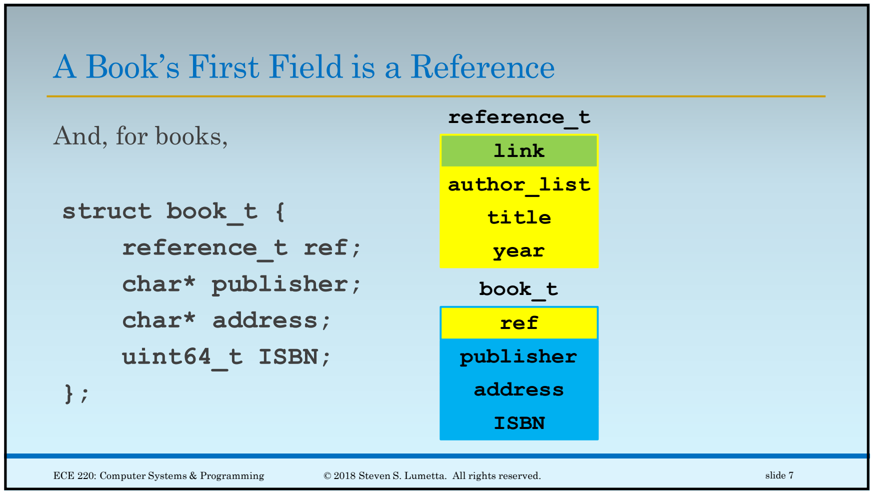
link

author_list

title

year

6



A Book's First Field is a Reference

And, for books,

```
struct book_t {
    reference_t ref;
    char* publisher;
    char* address;
    uint64_t ISBN;
};
```

reference_t

link

author_list

title

year

book_t

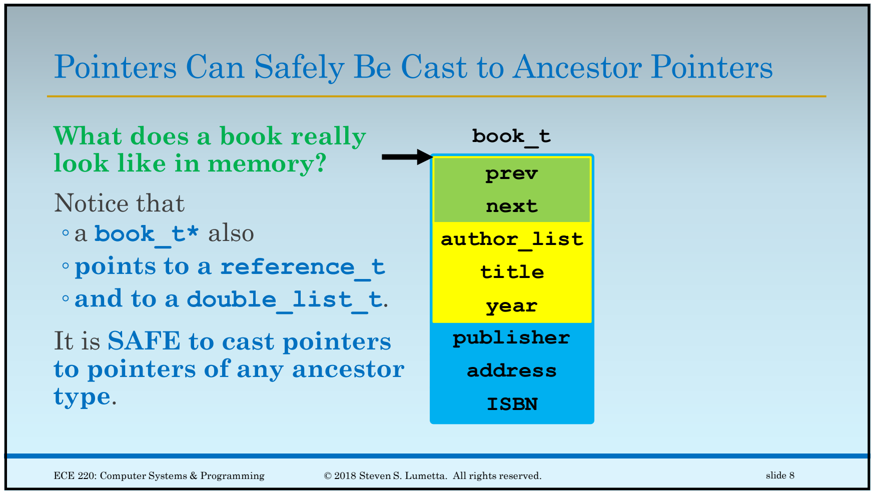
ref

publisher

address

ISBN

7



Pointers Can Safely Be Cast to Ancestor Pointers

What does a book really look like in memory?

Notice that

- a **book_t*** also
- **points to a reference_t**
- **and to a double_list_t.**

It is **SAFE to cast pointers to pointers of any ancestor type.**

book_t

prev

next

author_list

title

year

publisher

address

ISBN

8