

Add a Function Pointer for Printing a Citation

```
struct reference_t {
    double_list_t link;
    char* author_list;
    char* title;
    int32_t year;
    void32_t(print_citation)
        (reference_t* ref);
};
```

21

Function Pointers Allow Partial Customization

In any `paper_t`, the `print_citation` field points to `print_paper_citation`.

Similarly, in any `book_t`, the `print_citation` field points to `print_book_citation`.

And so forth.

Note that **we do not have to specialize**

- for example, we might choose to have
- `print_citation` point to `print_book_citation` in any `textbook_t`.

22

To Print All, Loop Over All Elements and Call

Now to print our bibliography...

```
double_list_t* elt;
reference_t* ref;
for (elt = biblio.next;
     &biblio != elt;
     elt = elt->next) {
    ref = (reference_t*)elt;
    (*ref->print_citation) (ref);
}
```

Iterate over all elements in the bibliography.

Function called now depends on actual type of `*ref`.

23

Cost Increases if We Customize Many Functions

As you saw,

- the **code is a little cleaner**
- but **allows customization of functionality by type**.

What's the cost?

One pointer per structure variable ...

... per function customized.

Can we reduce this cost?

24