

## Perhaps Useful to Write as Separate Functions

But maybe the citation printing code

- is lengthy, or
- is useful elsewhere.

In that case, write...

```
print_paper_citation
    (paper_t* paper);
print_book_citation
    (book_t* book);
and so forth.
```

17

## Each Case is Then Just a Function Call

Then, in `print_citation...`

```
switch (ref->type) {
    case TYPE_PAPER:
        print_paper_citation
            ((paper_t*)ref);
        break;
    // ...
}
```

Safe to cast: we know that `ref` points to a `paper_t`.

18

## A Function Pointer Has a Specific Signature

But if every case is just a function call...

...why not just add a function pointer to `reference_t` instead?

One problem: the functions don't have the same signatures...

```
print_paper_citation
    (paper_t* paper);
print_book_citation
    (book_t* book);
```

19

## Solution: Use a Pointer to the Known Ancestor Type

Let's

- **replace** all pointers with `reference_t*`
- and **cast** to the appropriate type in each function (we know it's safe in each case).

For example,

```
print_book_citation
    (reference_t* ref) {
    book_t* book = (book_t*)ref;
    // ...
```

20