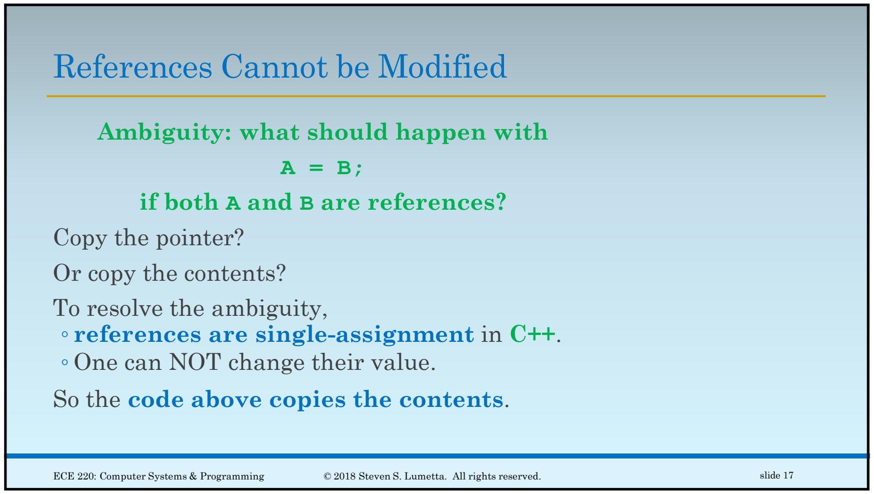


References Cannot be Modified

Ambiguity: what should happen with

`A = B;`

if both `A` and `B` are references?

Copy the pointer?

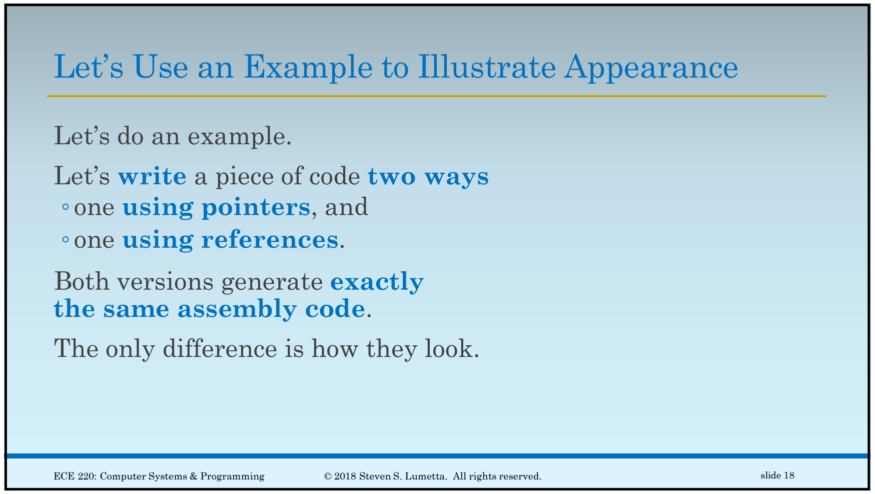
Or copy the contents?

To resolve the ambiguity,

- **references are single-assignment** in **C++**.
- One can NOT change their value.

So the **code above copies the contents**.

17



Let's Use an Example to Illustrate Appearance

Let's do an example.

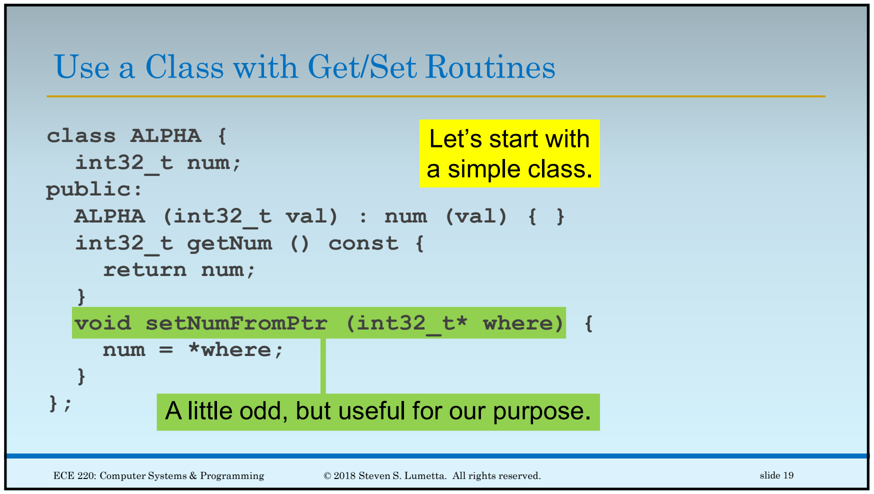
Let's **write** a piece of code **two ways**

- one **using pointers**, and
- one **using references**.

Both versions generate **exactly the same assembly code**.

The only difference is how they look.

18



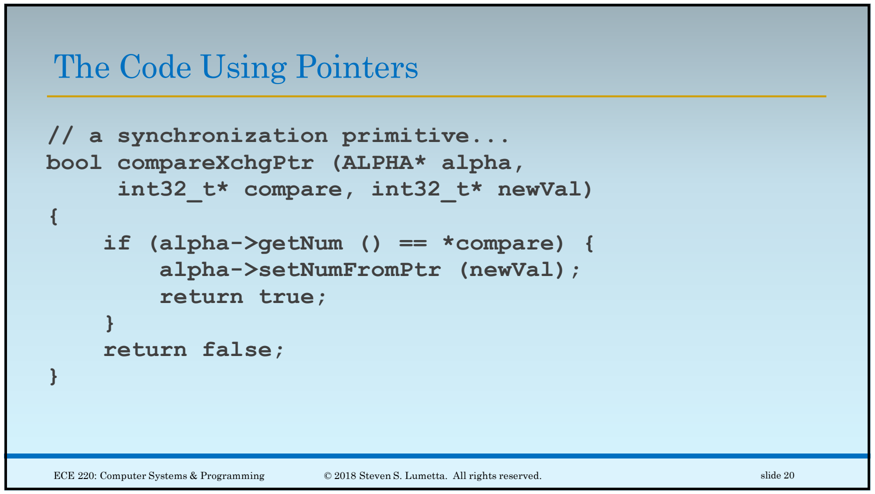
Use a Class with Get/Set Routines

```
class ALPHA {
  int32_t num;
public:
  ALPHA (int32_t val) : num (val) { }
  int32_t getNum () const {
    return num;
  }
  void setNumFromPtr (int32_t* where) {
    num = *where;
  }
};
```

Let's start with a simple class.

A little odd, but useful for our purpose.

19



The Code Using Pointers

```
// a synchronization primitive...
bool compareXchgPtr (ALPHA* alpha,
     int32_t* compare, int32_t* newVal)
{
    if (alpha->getNum () == *compare) {
        alpha->setNumFromPtr (newVal);
        return true;
    }
    return false;
}
```

20