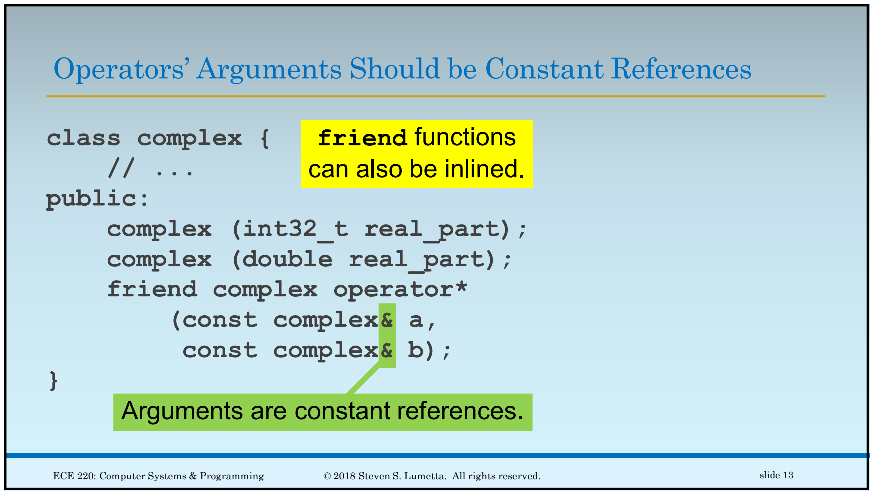


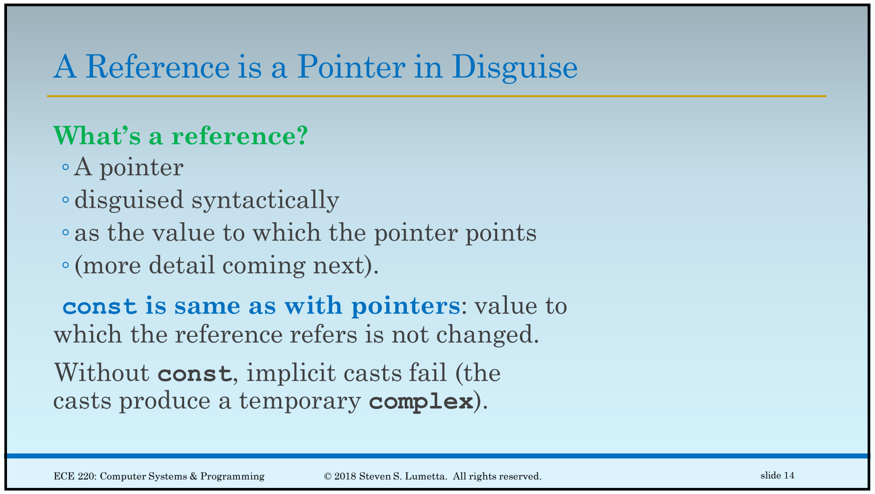
Operators' Arguments Should be Constant References

```
class complex {
    // ...
public:
    complex (int32_t real_part);
    complex (double real_part);
    friend complex operator*
        (const complex& a,
         const complex& b);
}
```

friend functions can also be inlined.

Arguments are constant references.

13



A Reference is a Pointer in Disguise

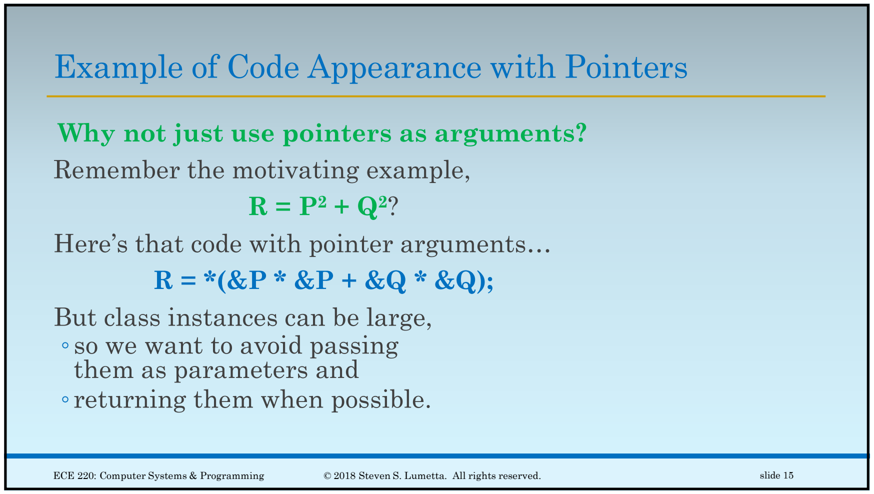
What's a reference?

- A pointer
- disguised syntactically
- as the value to which the pointer points
- (more detail coming next).

const is same as with pointers: value to which the reference refers is not changed.

Without **const**, implicit casts fail (the casts produce a temporary **complex**).

14



Example of Code Appearance with Pointers

Why not just use pointers as arguments?

Remember the motivating example,

$$R = P^2 + Q^2?$$

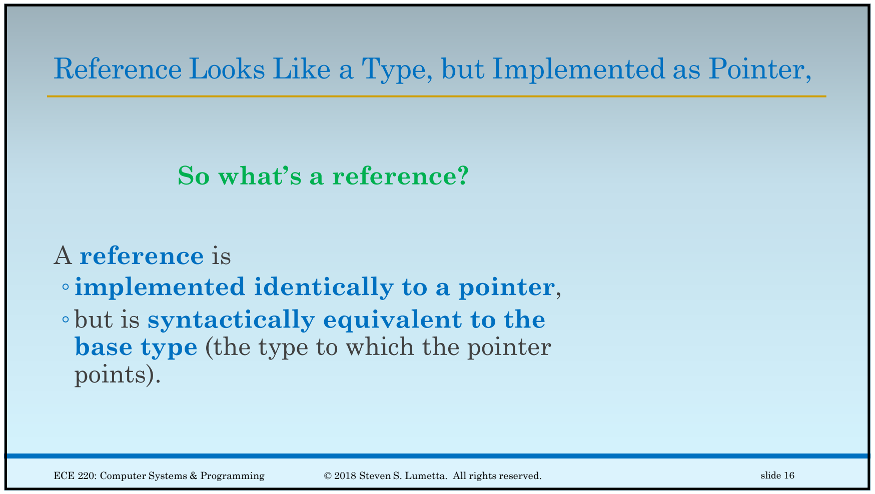
Here's that code with pointer arguments…

R = *(&P * &P + &Q * &Q);

But class instances can be large,

- so we want to avoid passing them as parameters and
- returning them when possible.

15



Reference Looks Like a Type, but Implemented as Pointer,

So what's a reference?

A **reference** is

- **implemented identically to a pointer**,
- but is **syntactically equivalent to the base type** (the type to which the pointer points).

16