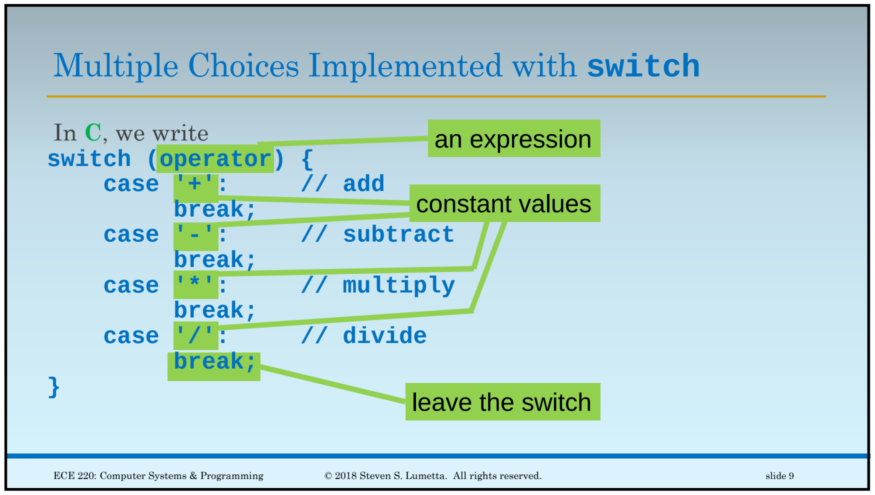


Multiple Choices Implemented with **switch**

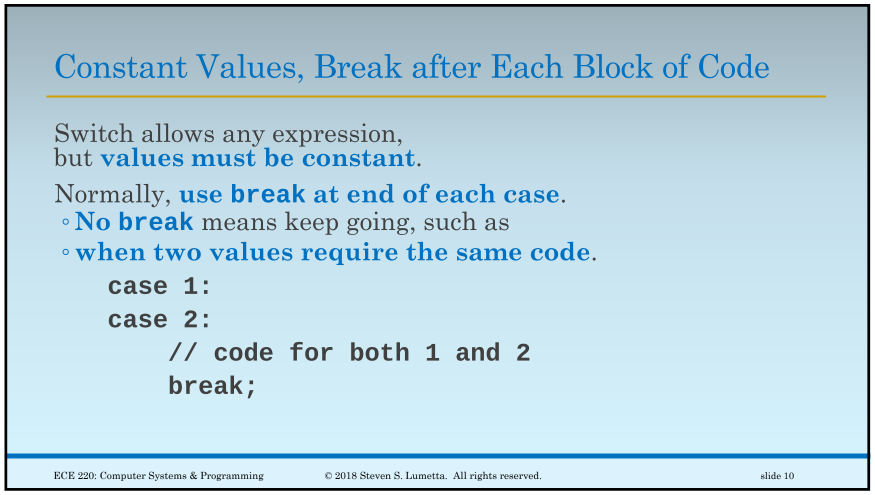
In **C**, we write

```
switch (operator) {
    case '+':      // add
         break;
    case '-':      // subtract
         break;
    case '*':      // multiply
         break;
    case '/':      // divide
         break;
}
```

an expression

constant values

leave the switch

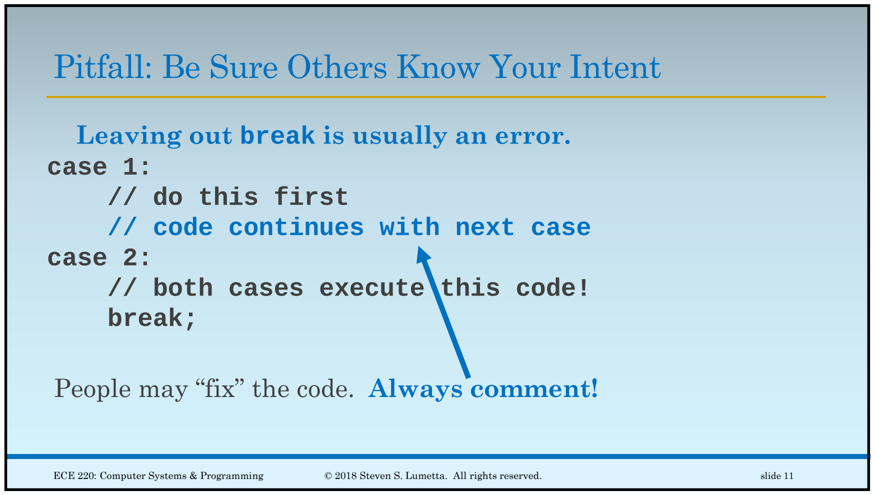


Constant Values, Break after Each Block of Code

Switch allows any expression,
but **values must be constant**.

Normally, **use `break` at end of each case**.
- **No `break`** means keep going, such as
- **when two values require the same code**.

```
    case 1:
    case 2:
        // code for both 1 and 2
        break;
```

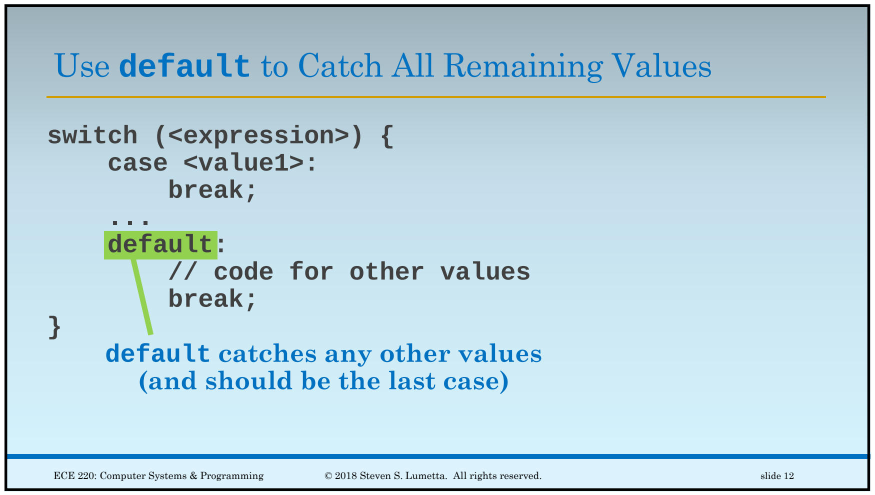


Pitfall: Be Sure Others Know Your Intent

Leaving out `break` is usually an error.

```
case 1:
    // do this first
    // code continues with next case
case 2:
    // both cases execute this code!
    break;
```

People may “fix” the code. **Always comment!**



Use **default** to Catch All Remaining Values

```
switch (<expression>) {
    case <value1>:
        break;
    ...
    default:
        // code for other values
        break;
}
```

**`default` catches any other values
(and should be the last case)**