All C expressions have a type:

* Constants
* Variables
* Function return values
* Every sub-expression of a larger expression has a type

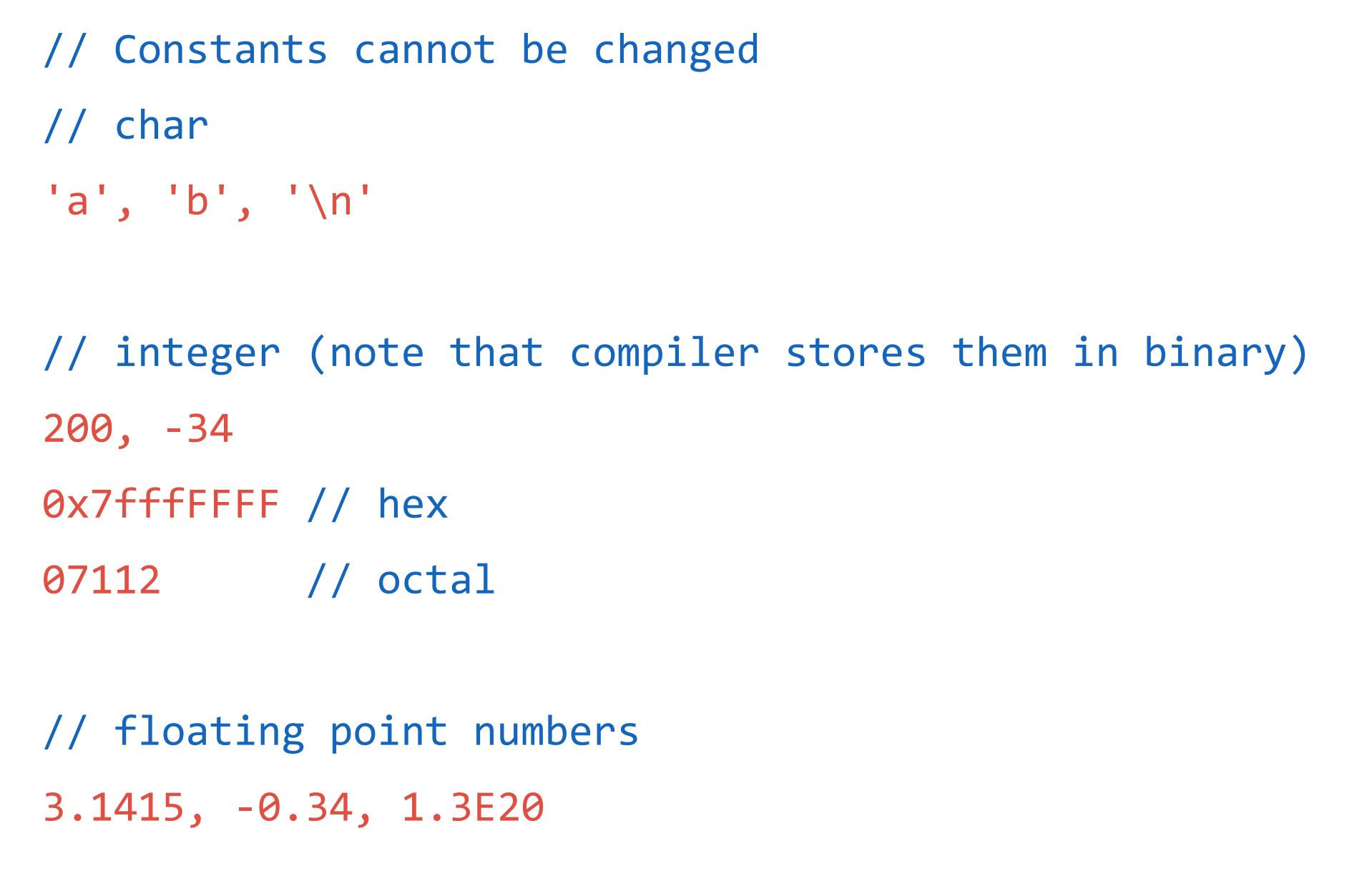
Adding a semicolon to an expression makes it a statement

Data Types (basic)

Int

Char (single bit, stores a character)

Float (decimals)



Variables must be declared and initialized before they can be used

Similar to Java

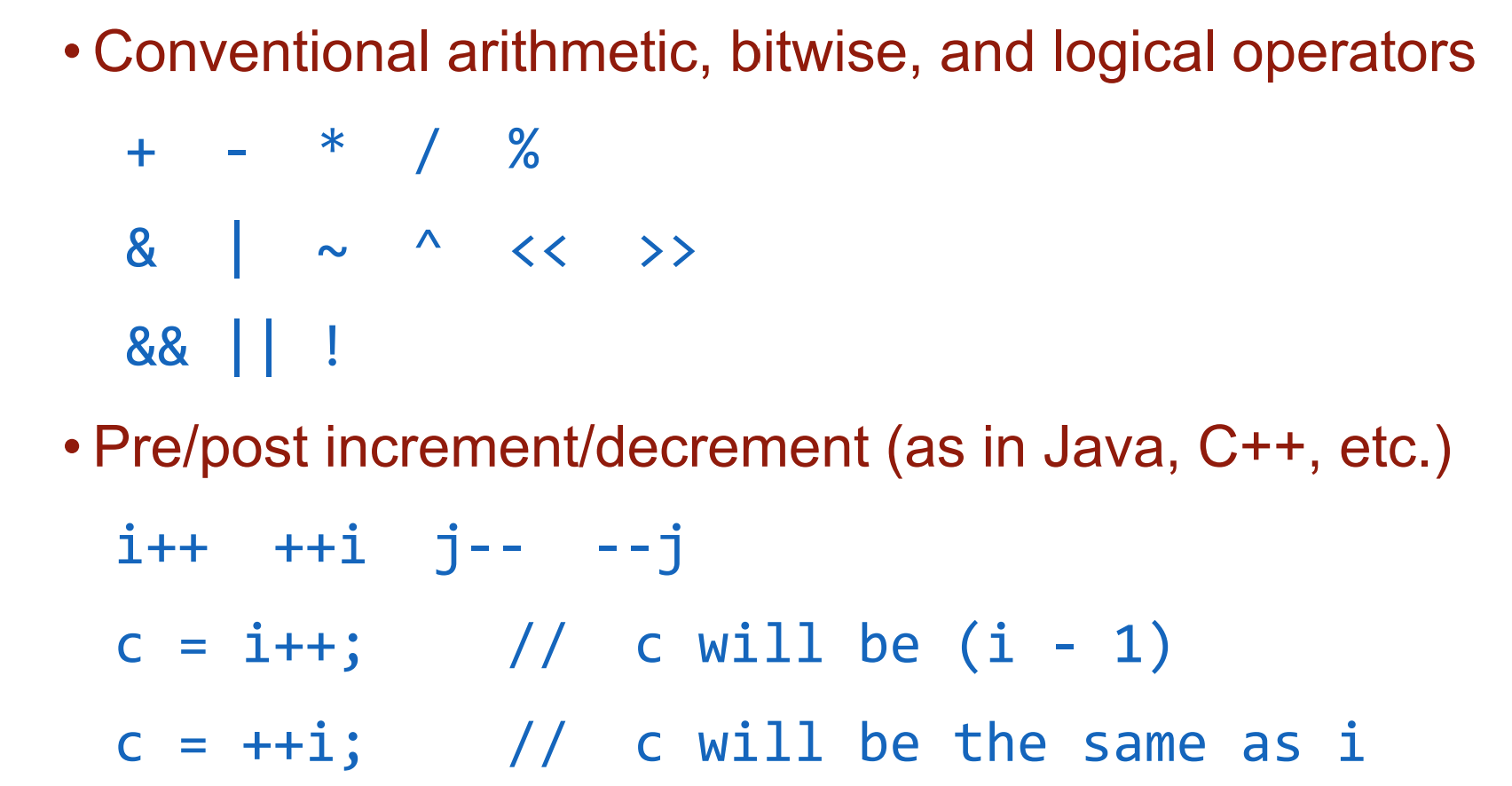
Cant start with digits

Multiple variables can be declared together

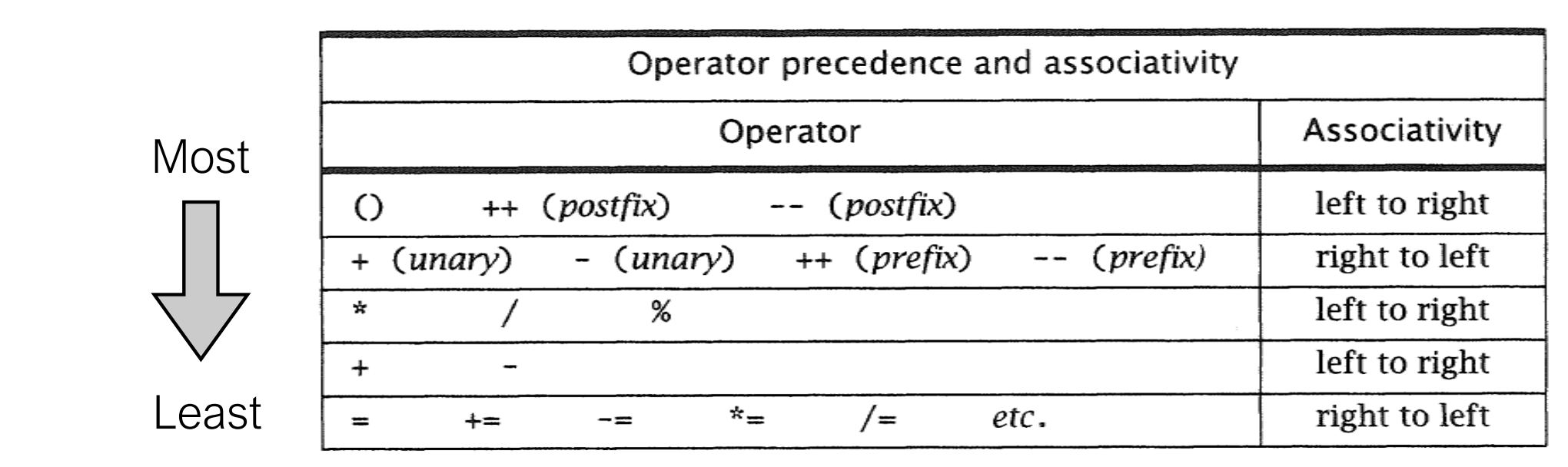
Char c;

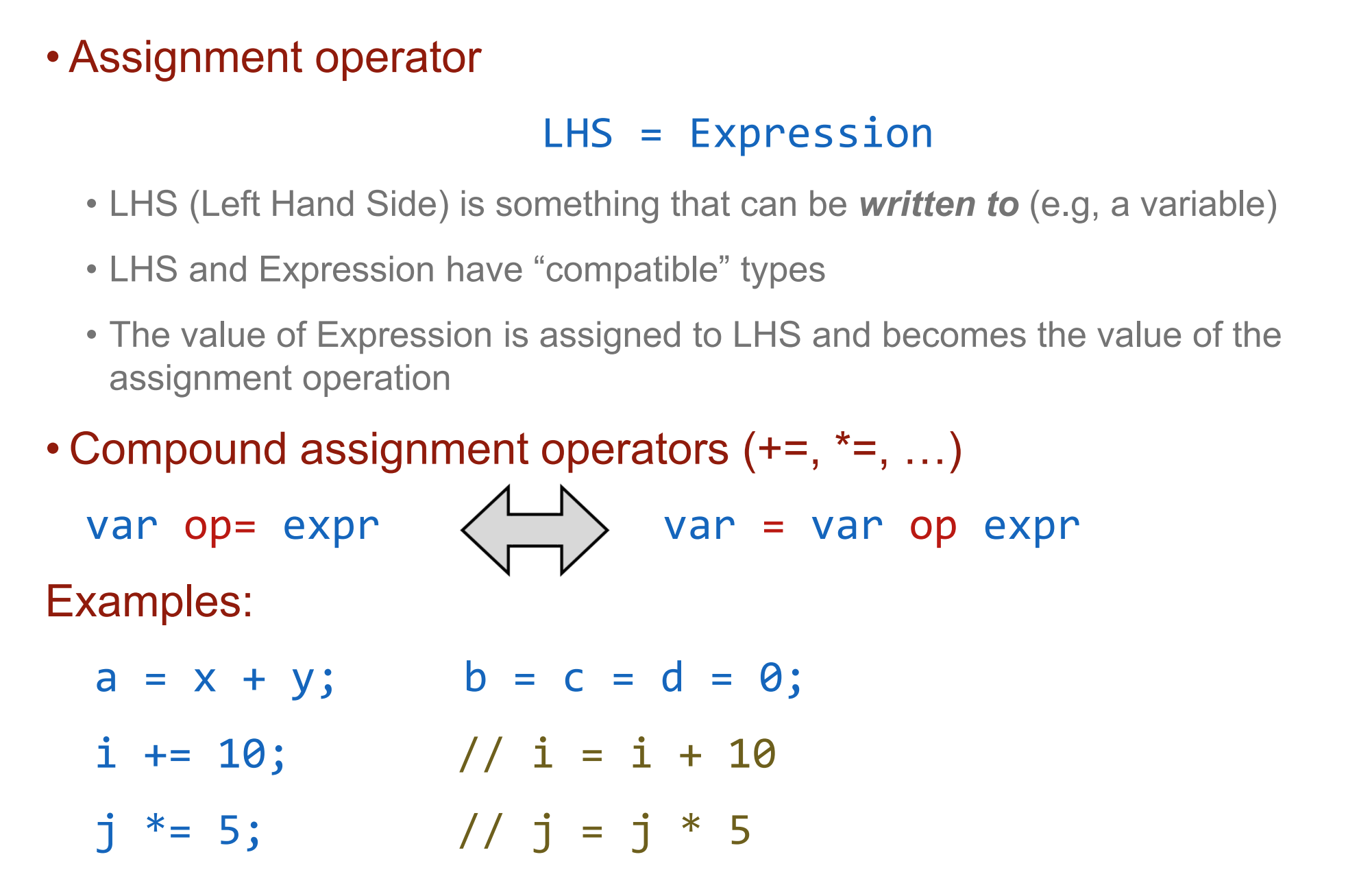
int i, j, k = 1; //Declares i, j, and k, but also initializes k

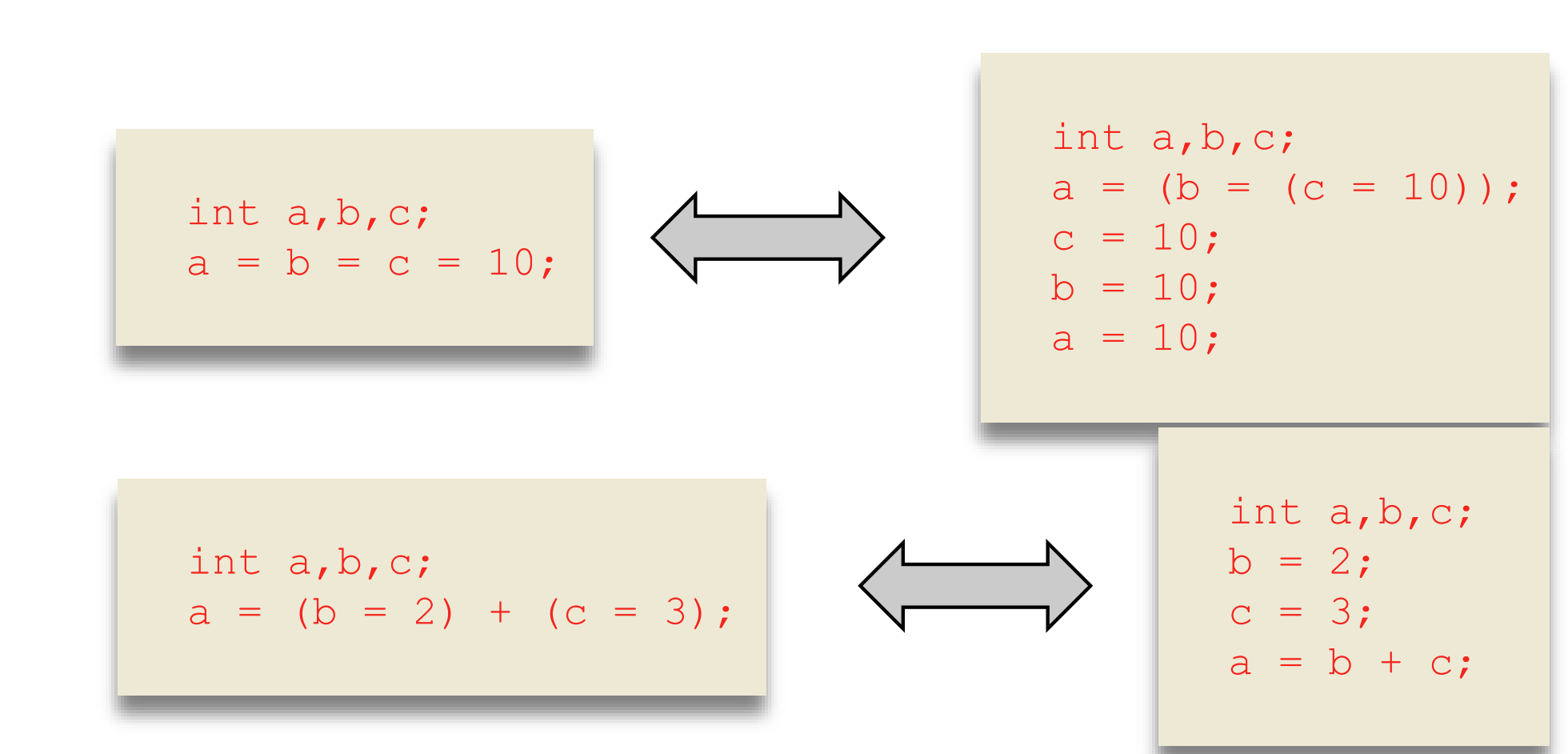
float f;

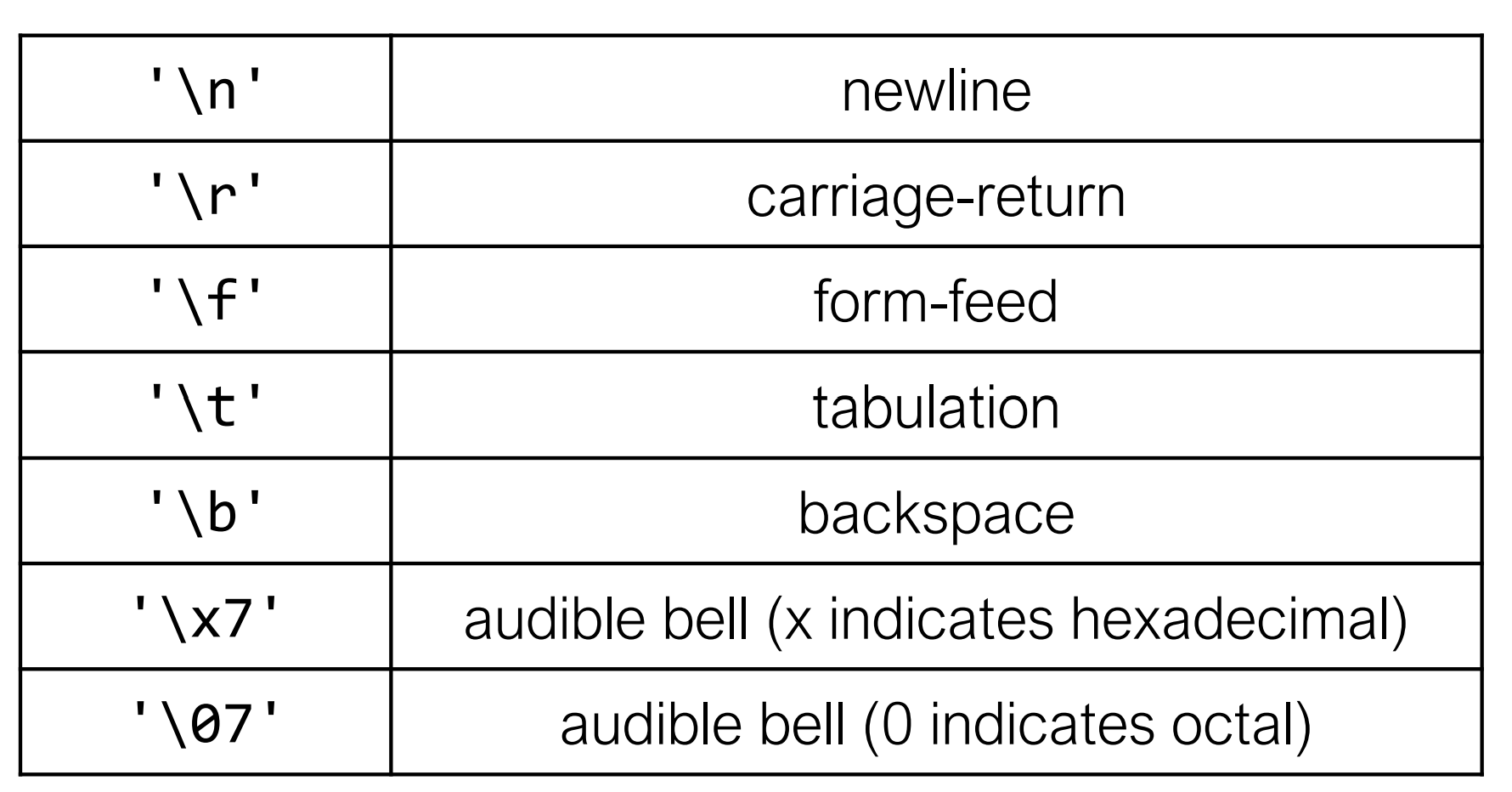


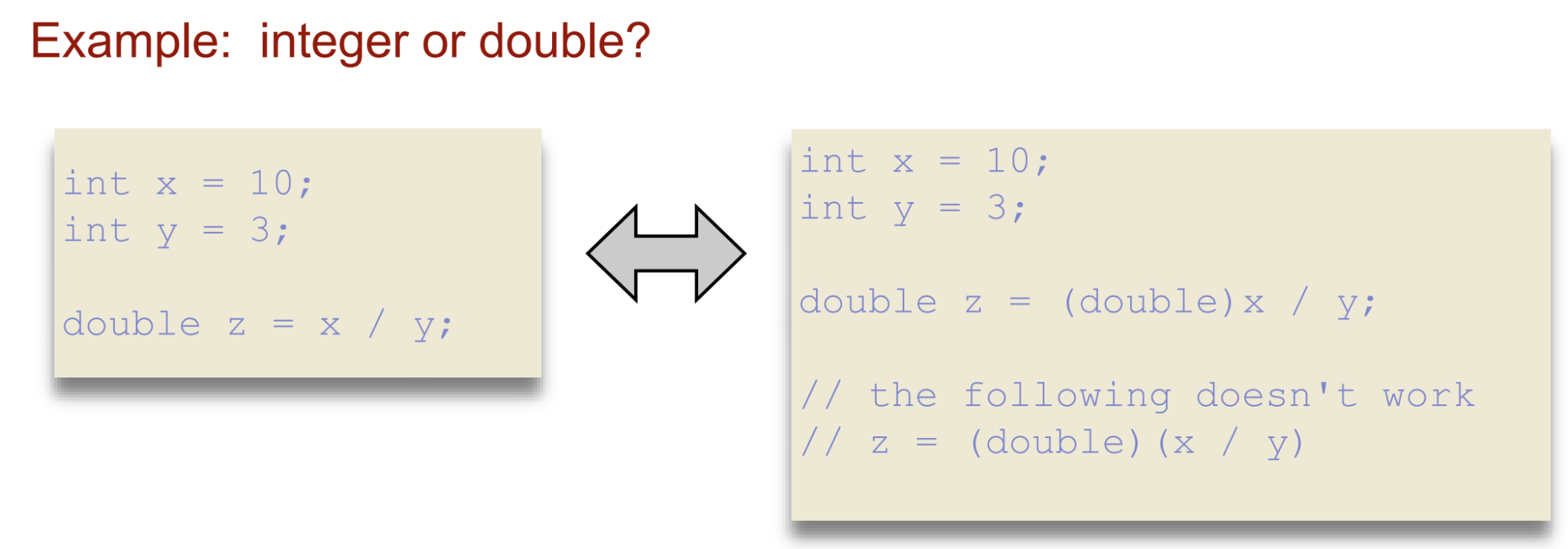
Use parentheses to circumvent below table

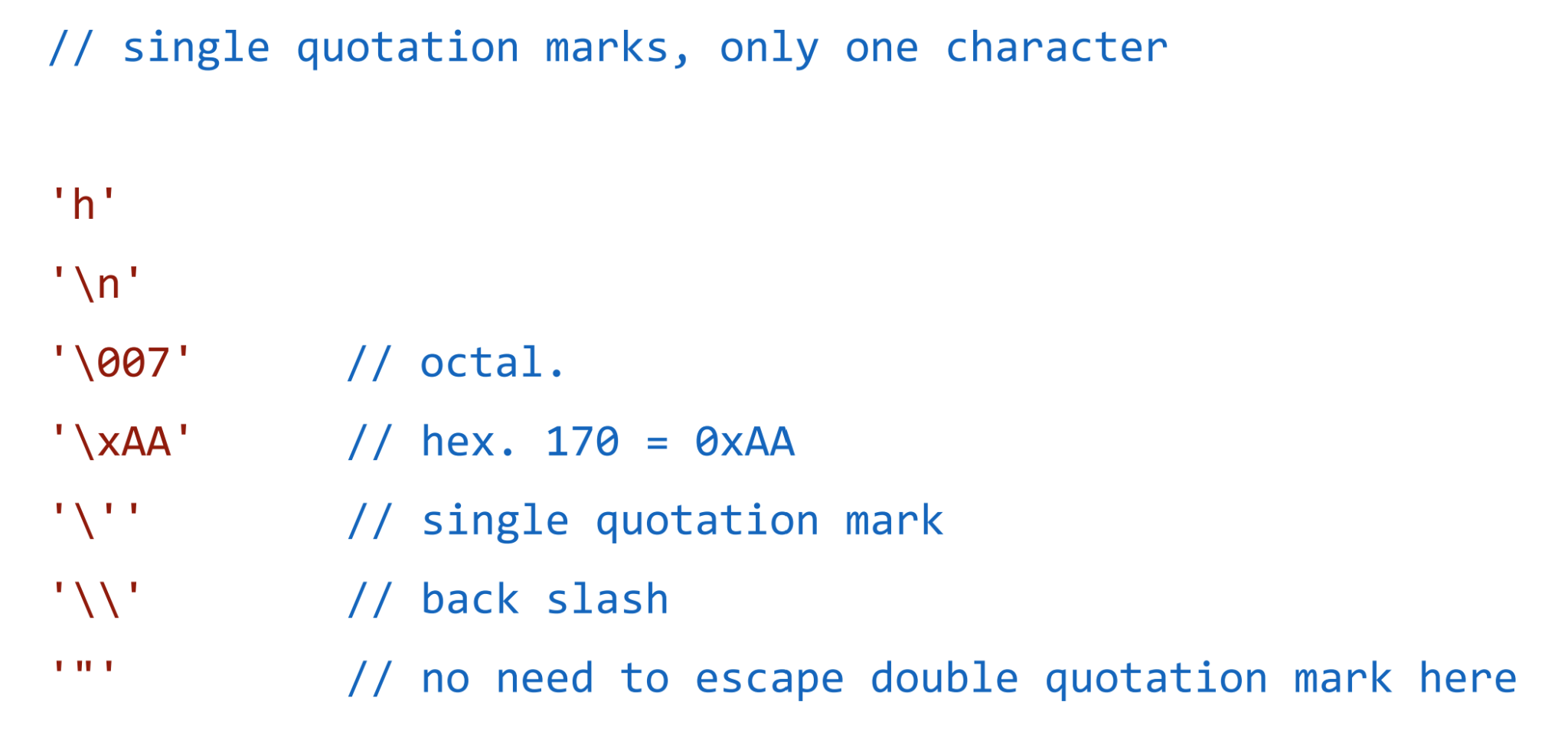


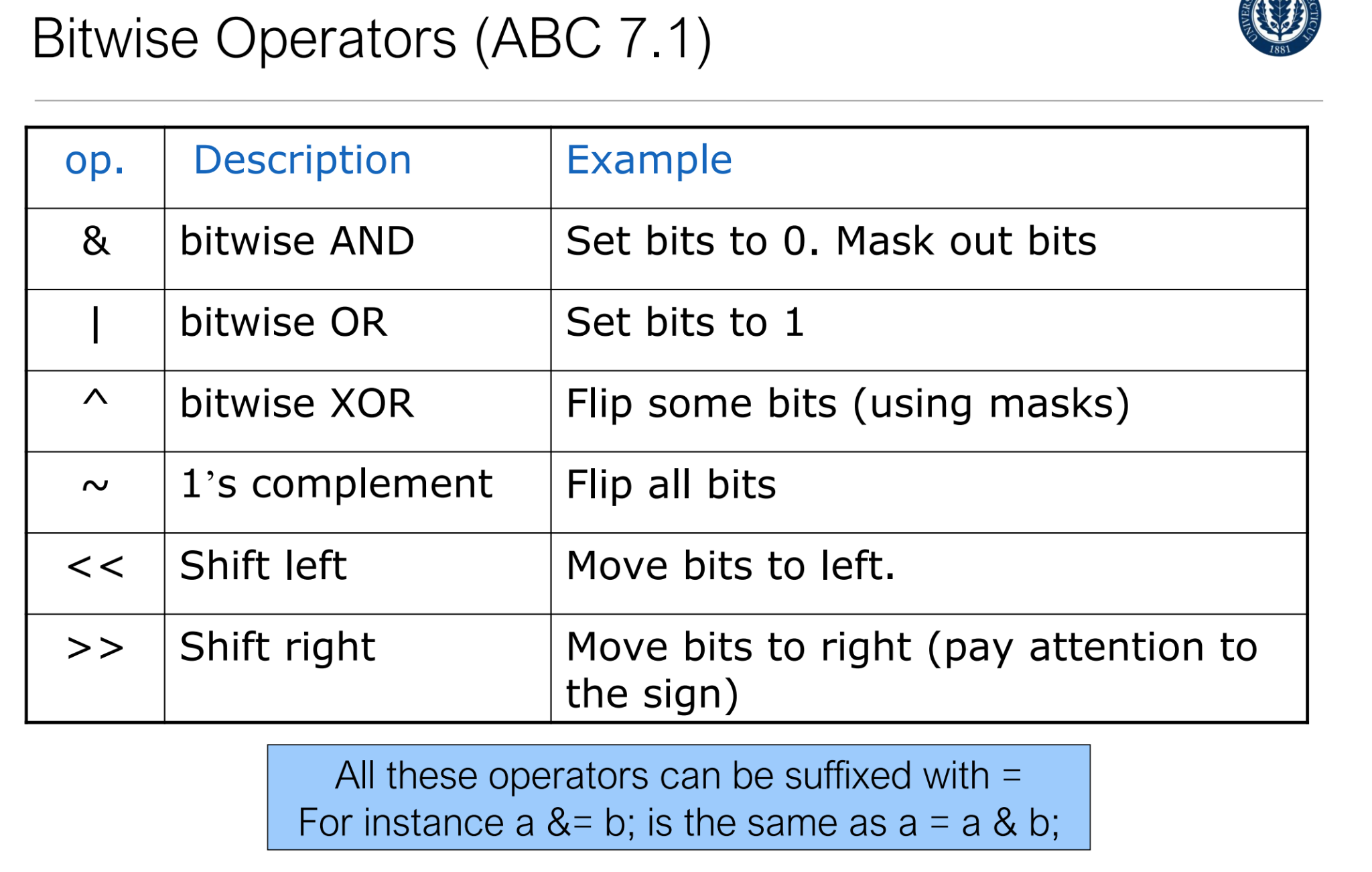


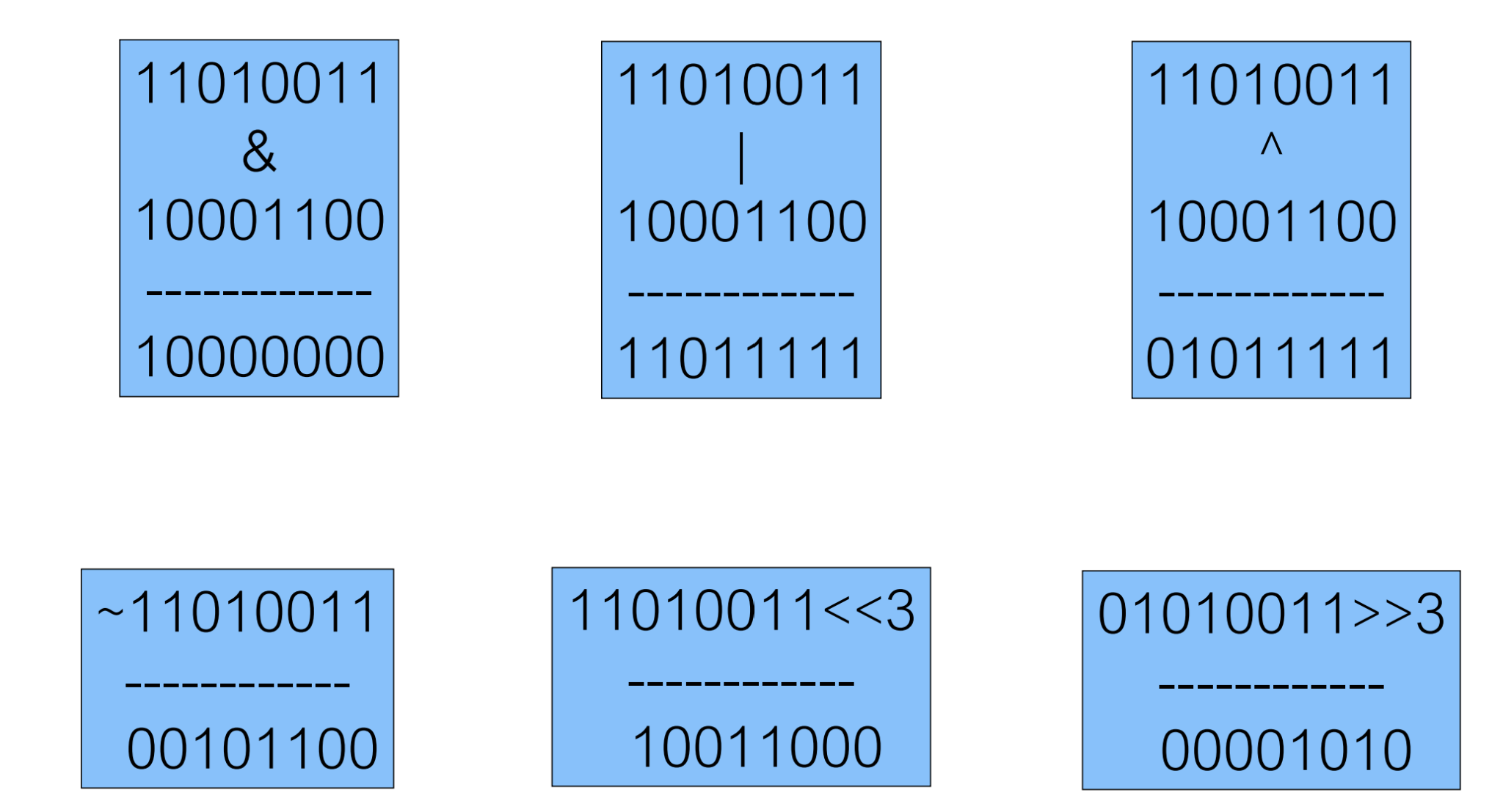


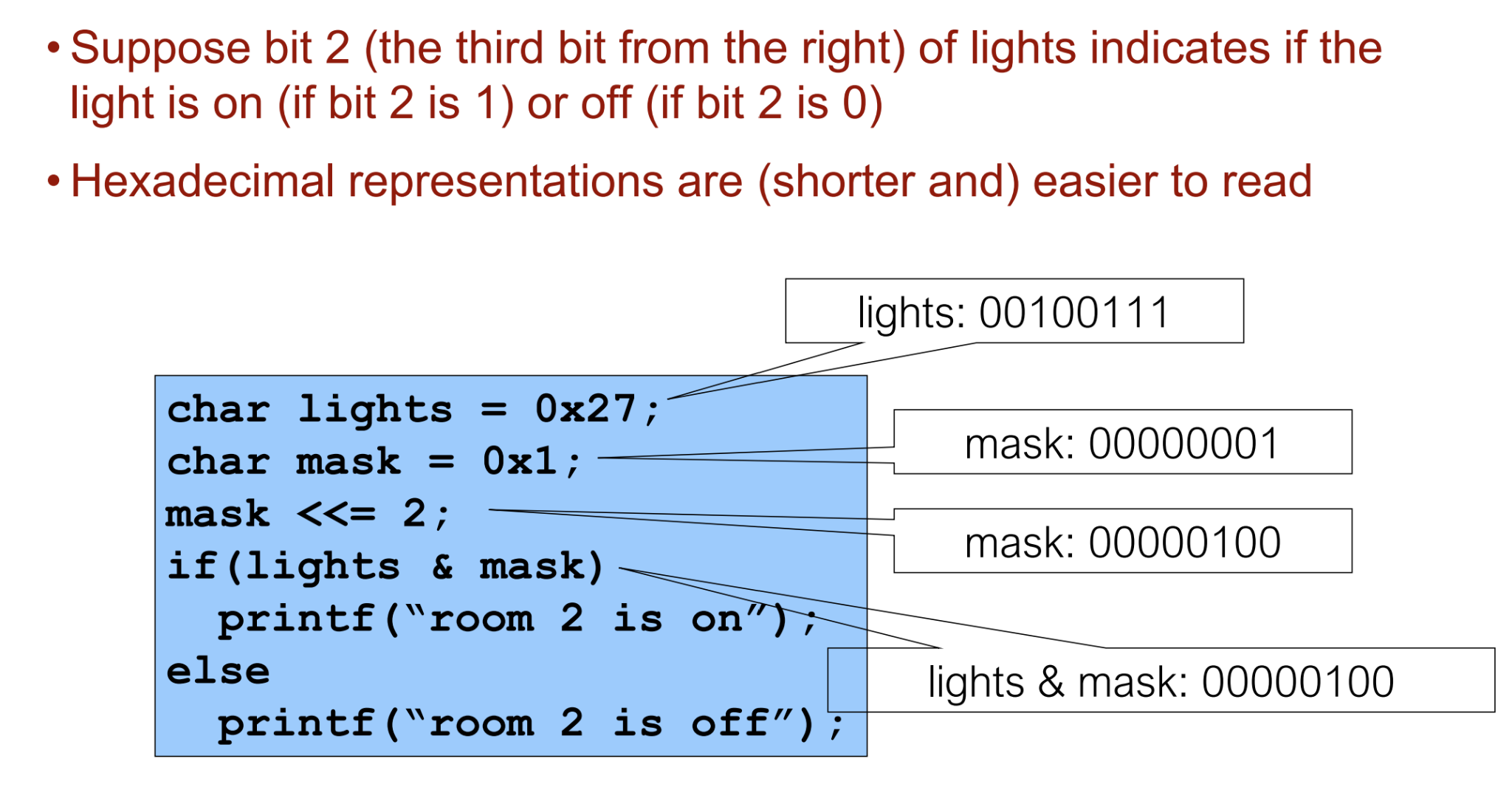














| 1/23: Professor focused on the data types and general changes the C has. Slept through the parts on information location and size. Setting Bits etc. General changes in the C language are nothing difficult. |
| --- |

1/25

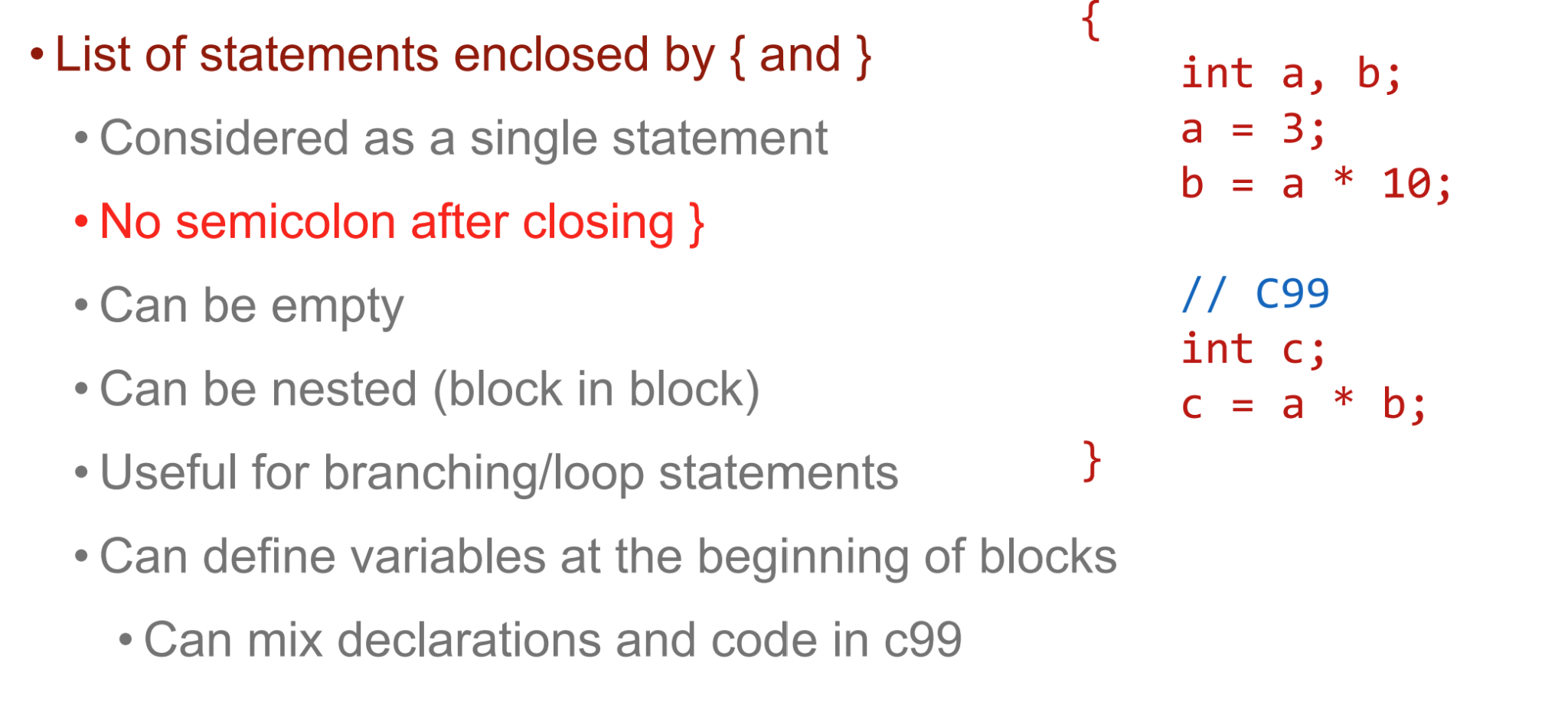
Flow of Control

States are **Normally** executed sequentially

For selective or repeated execution we have all the usual suspects:

Blocks, If and if-else, while, for, switch, break, continue

Blocks compound statements



Bools are either 0 or 1 (of int type)

Comparison (known)

Logical operators && || ! (and or not)

Branching: if and if-else statement

“Exp” is typically a comparison or logical expression but can be any expression (float/double, point, …)

The statements can be compound statements (blocks)

Or other if statements

if (i < j) min = i; else min = j; //if else combined into one line

// if else with blocks

if (i < j) {

k = i;

printf("i is selected.\n");

} // no ; here

else{

k = j;

printf("j is selected.\n");

}

Ternary operator

Takes three expression as operands

exp1 ? exp2 : exp3

exp1 is evaluated first •

If exp1 is non-zero (true),

exp2 is evaluated and its value is used as the value of the ternary expression

If exp1 is zero (false),

exp3 is evaluated and its value is used as the value of the ternary expression

Example: min = i < j ? i : j;

I if true

J if false

While Loop

Computing sum of 0.99

Int i = 0, sum 0;

While (i < 100) {

Sum = sum + i;

I++;

}

// Same as

While (i < 100) sum += i++;

Do-while loop

Checks condition after executing loop body

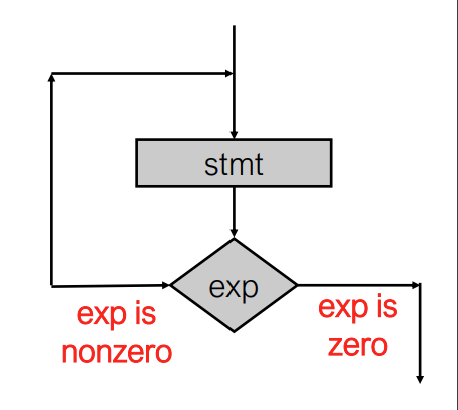
Statement happens at least once

int i = 0, sum = 0;

do {

sum = sum + i; i++;

} while (i < 100);



For loop

Sometimes called “counting” loop

More like swiss-army knife!

Three expressions:

Initialization, condition, increment

Equivalent to

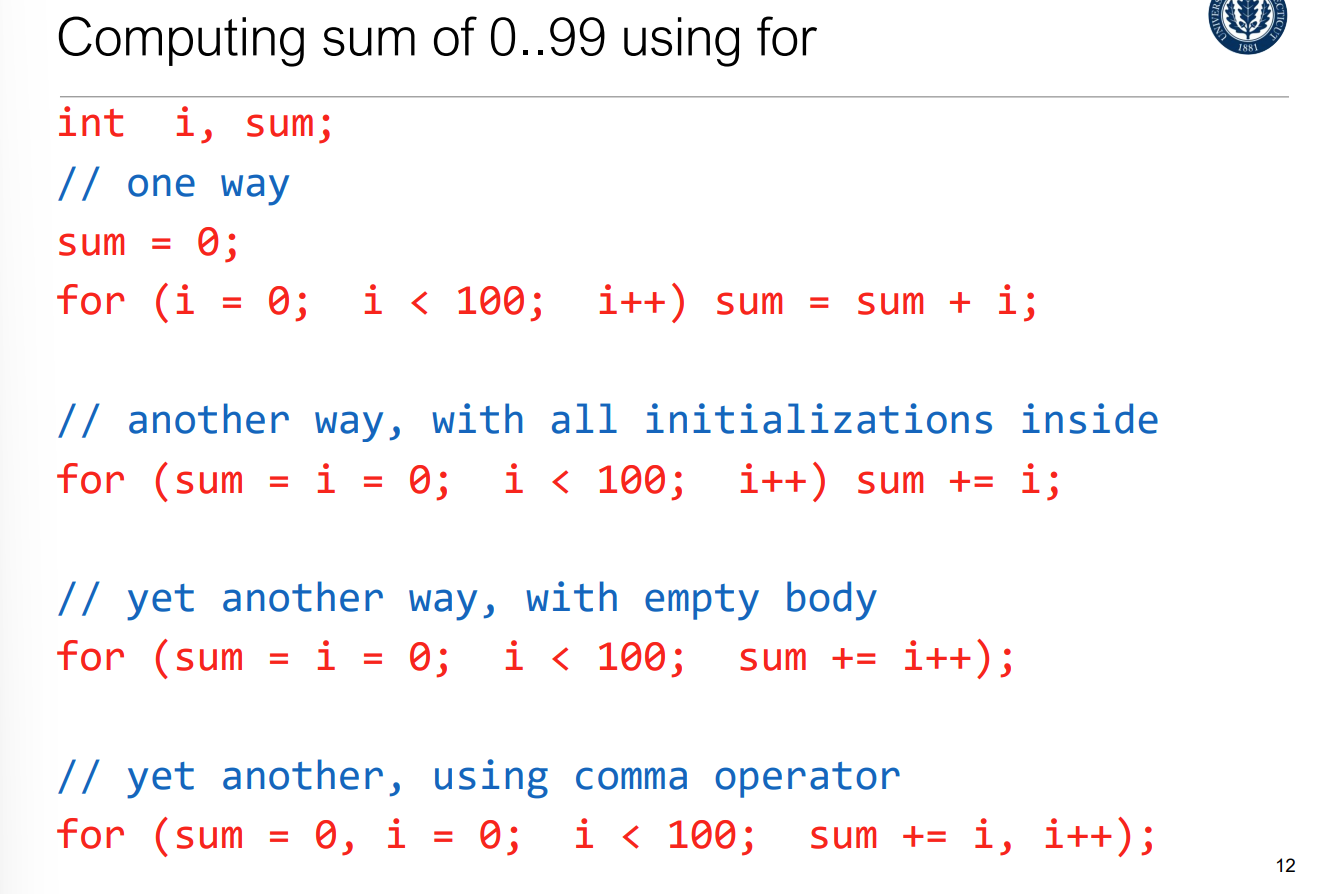
Exp1;

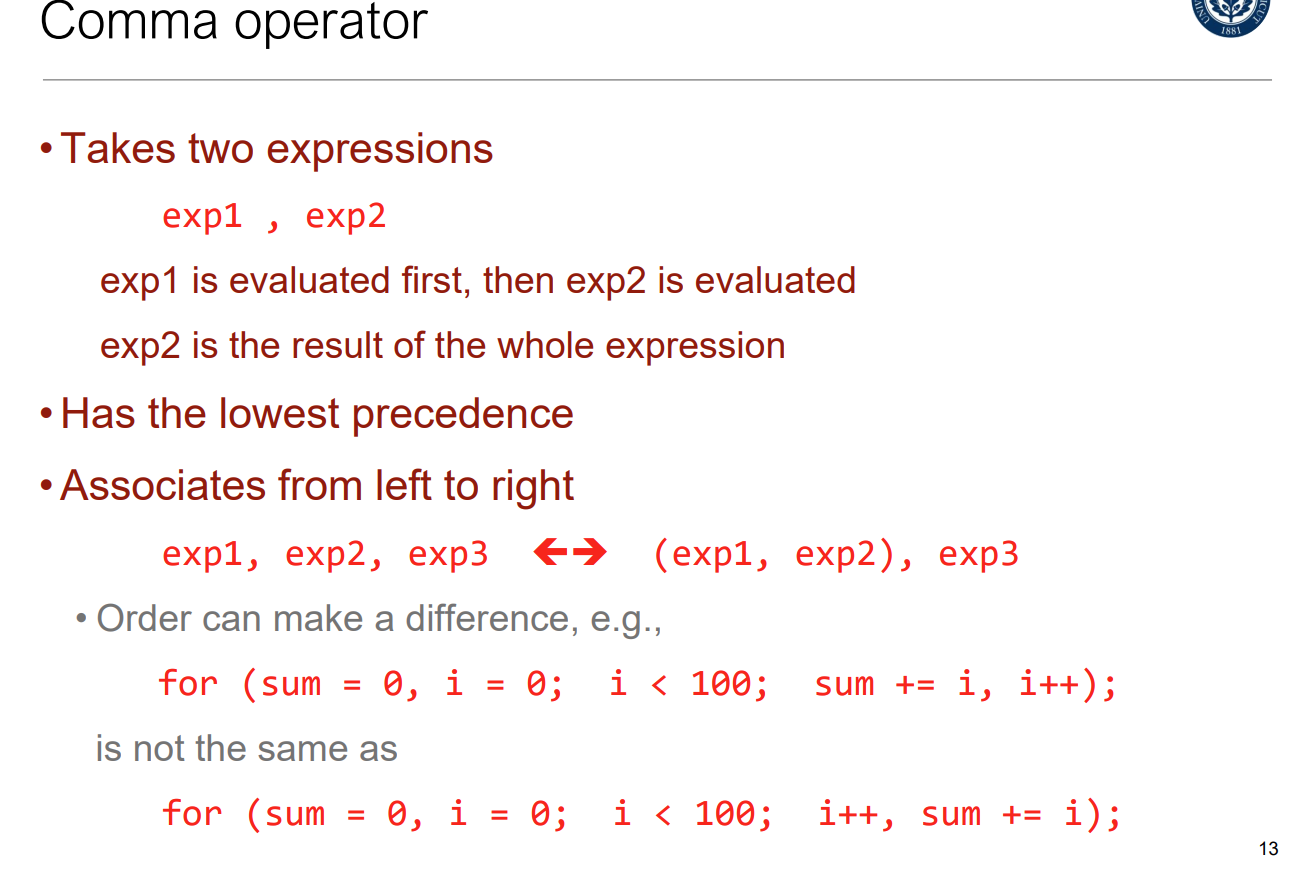
While (exp2) {

<stmt>

exp3;

}





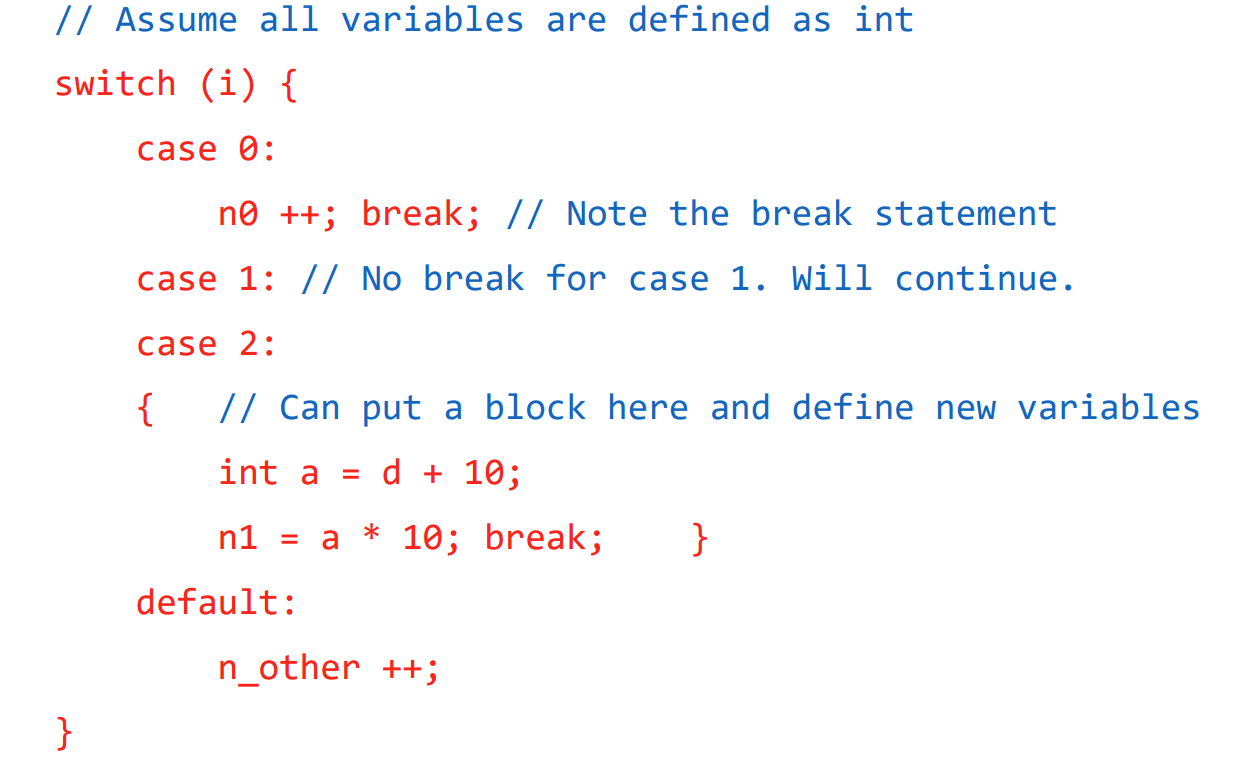
Else if else if (same as in python)

Switch

Also called “selection” statement

Works like a train station

Assume i is either 0,1, or 2

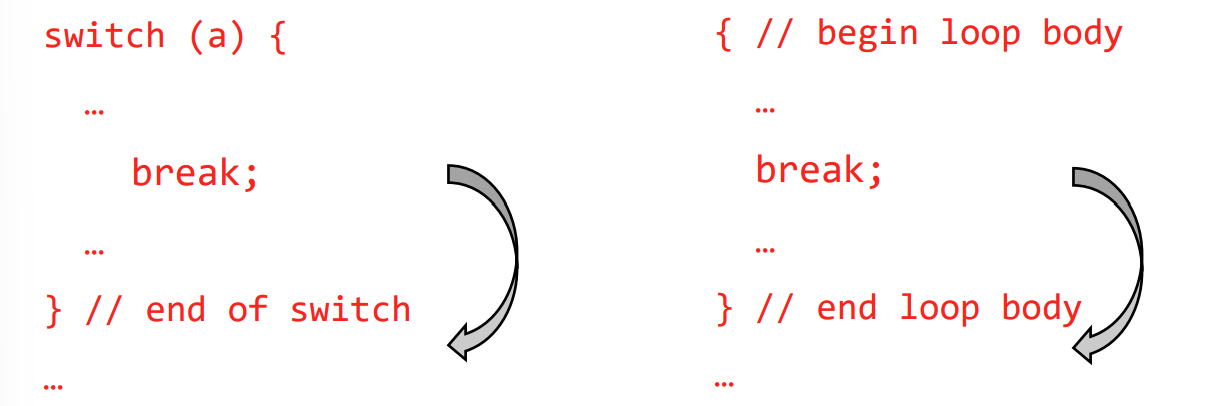


Break Statement

Most commonly used in switch statements

Prevents fall through to next case

Also works in loops (for, while, do-while)

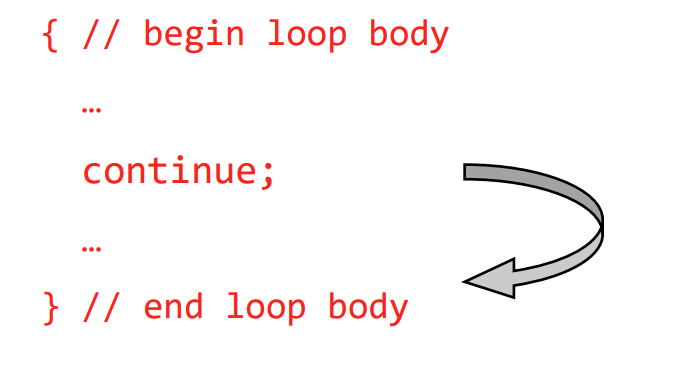


Continue state

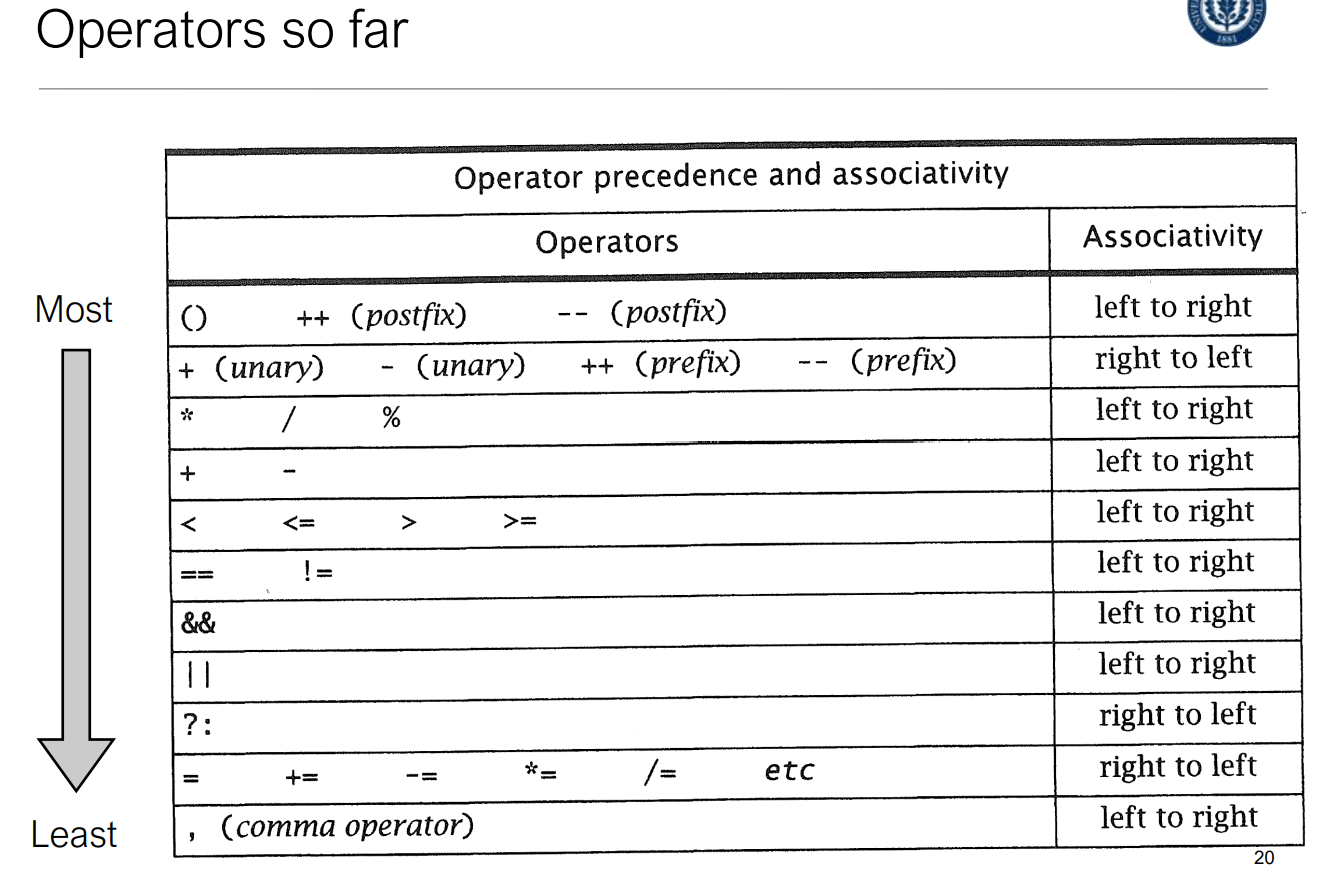
Similar to break however used for loops only,

Breaks the current loop and immediately starts the next loop;

Works also in nested if/else

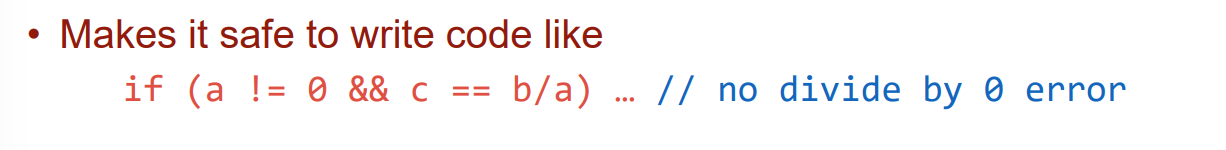


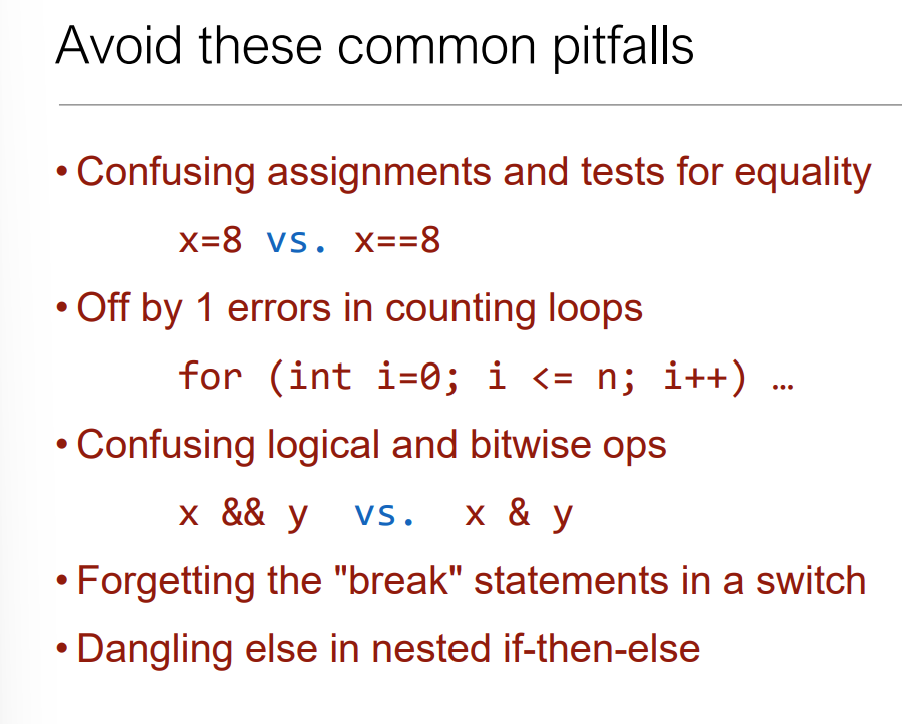
Imagine arrow going the other way

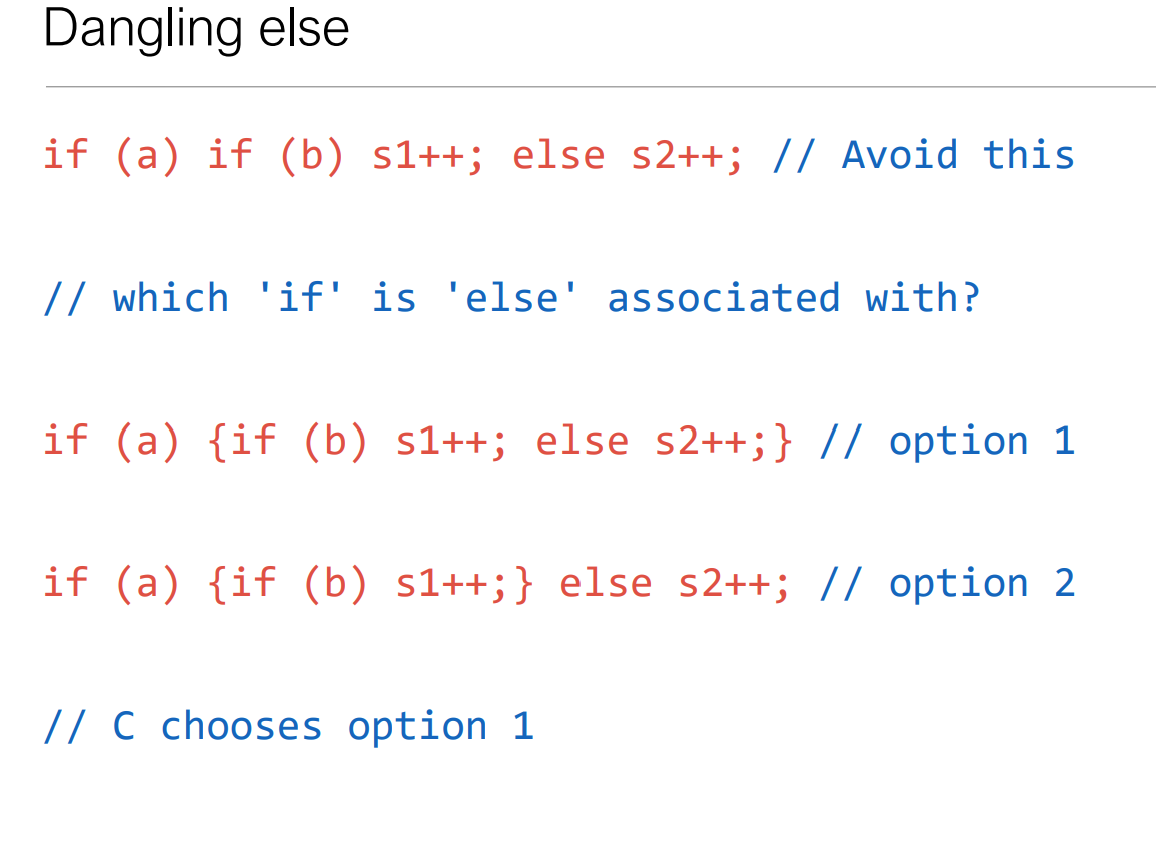


With && and ||

if the first && is false or first || is true, it skips the second statement





=-