

slide 3

### Our Class Focuses on Four Types of Operator in C

The **C** language supports many operators.

In our class, we consider four types:

- arithmetic operators
- **bitwise** Boolean operators
- relational / comparison operators
- the **assignment** operator

ECE 120: Introduction to Computing

We also introduce logical operators, but leave their full meaning for ECE220.

© 2016 Steven S. Lumetta. All rights reserved.

Five Arithmetic Operators on Numeric Types Arithmetic operators in C include • addition: +• subtraction: \* • multiplication: • division: • modulus: % (integers only) The C library includes many other functions, such as exponentiation, logarithms, square roots, and so forth. We leave these for ECE220. ECE 120: Introduction to Computing © 2016 Steven S. Lumetta. All rights reserved. slide 4

Declare: int A = 120; int B = 42; Then A + B evaluates to 162 A - B evaluates to 78 A * B evaluates to 5040 A % B evaluates to 36 A / B evaluates to 2 What's going on with division?		• <b>unsigned int</b> • <b>A</b> is a large numb Integer division • Trying to <b>divide</b> (floating-point pr	by 0 ends the program oduces infinity or NaN). evaluates to an integer,	
ECE 120: Introduction to Computing © 2016 Steven S. Lumetta. All rights reserved.	slide 5	ECE 120: Introduction to Computing	© 2016 Steven S. Lumetta. All rights reserved.	slide 6

Integer div	vision is round	ded to an i	nteger.		
Rounding	depends on t	the proce	ssor.		
Most mode	rn processors	s round to	owards 0	, so	
	11 / 3	evaluates	to 3		
	-11 / 3	evaluates	to <b>-3</b>		
Modulus A	% B is defin	ned such tl	nat		
(A /	B) * B + (	(A % B)	is equal t	A c	
So (-11 %	3) evaluate	es to <b>-2</b> .			
M	dulus is not	t alwavs i	oositive.		

Bitwise operat	tors in	C include	
• AND:	&		
• OR:	1		
• NOT:	~		
• XOR:	^		
• left shift:	<<		
• right shift:	>>		
In some langu	ages, <b>'</b>	means exponentation,	
n some langu out not in the			

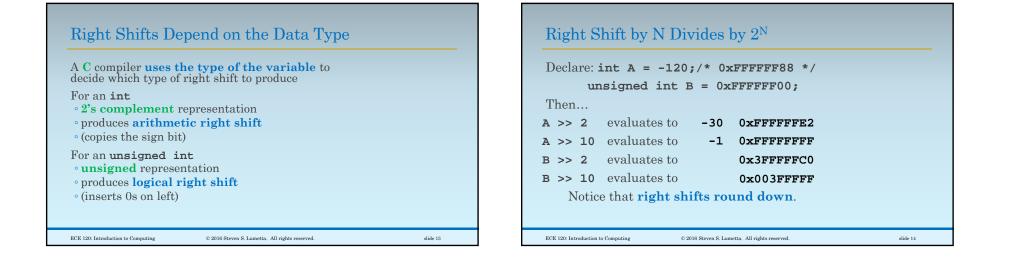
Bitwise Operators Treat Numbers as Bits	Bitwise Operators Treat Numbers as Bits
Declare: int $A = 120$ ; int $B = 42$ ;	Declare: int $A = 120$ ; int $B = 42$ ;
/* A = 0x00000078, B = 0x0000002A	/* A = 0x00000078, B = 0x0000002A
using C's notation for hexadecimal. */	using C's notation for hexadecimal. */
Then	Then
<b>A &amp; B</b> evaluates to <b>40 0x0000028</b>	<b>A &amp; B</b> evaluates to <b>40 0x0000028</b>
0000 0000 0000 0000 0000 0111 1000 Apply AND to	A   B evaluates to 122 0x000007A
AND 0000 0000 0000 0000 0000 0000 0000 0	~A evaluates to -121 0xFFFFF87
0000 0000 0000 0000 0000 0000 0010 1000 pairs of bits.	<b>A</b> ^ <b>B</b> evaluates to <b>82 0x0000052</b>
ECE 120: Introduction to Computing © 2016-2017 Steven S. Lumetta. All rights reserved. alide 9	ECE 120: Introduction to Computing © 2016-2017 Steven S. Lumetta. All rights reserved. slide 10

slide 11

# Left Shift by N Multiplies by $2^N$

Shifting left by N bits adds N 0s on right. • It's like multiplying by 2 <sup>N</sup> .
• N bits lost on left! (Shifts can overflow.)
Declare: int A = 120;/* 0x00000078 */
unsigned int B = 0xFFFFFF00;
Then
A << 2 evaluates to 480 0x000001E0
B << 4 evaluates to ( <b!) 0xfffff000<="" td=""></b!)>
ECE 120: Introduction to Computing © 2016 Steven S. Lumetta. All rights reserved.

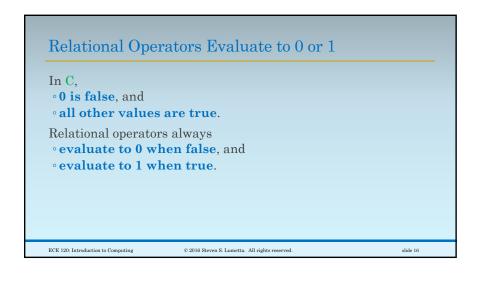
Right Shift by N Divides by 2 <sup>N</sup>	
A question for you: What bits appear on the left when shifting right?	
Declare: int A = 120;/* 0x0000078 */	
A >> 2 evaluates to <b>30 0x000001E</b>	
What about <b>0xFFFFF00 &gt;&gt; 4</b> ?	
Is <b>0xFFFFFF00</b> equal to	
<b>-256</b> $(/16 = -16$ , so insert 1s)? or equal to	
4,294,967,040 (/16 = 268,435,440, insert 0s)?	
ECE 120: Introduction to Computing © 2016 Steven S. Lumetta. All rights reserved.	slide 12

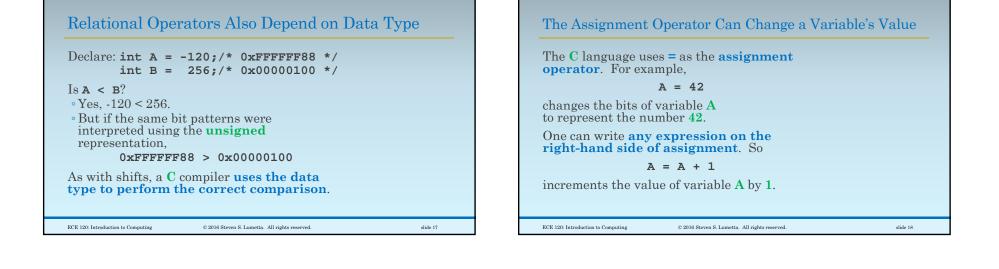


slide 15

#### Six Relational Operators

Relational operators	in <b>C</b> i	nclude
• less than:	<	
• less or equal to:	<=	
• equal:	==	(TWO equal signs)
• not equal:	!=	
• greater or equal to:	>=	
• greater than:	>	
C operators cannot in they be reordered (so		
ECE 120: Introduction to Computing	© 2016	Steven S. Lumetta. All rights reserved.





#### Only Assign Values to Variables

A **C** compiler can not solve equations.

For example,

ECE 120: Introduction to Computing

A + B = 42

results in a compilation error (the compiler cannot produce instructions for you).

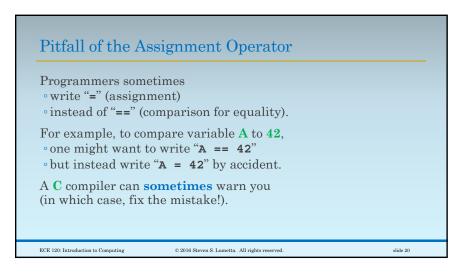
The left-hand side of an assignment must be a variable.\*

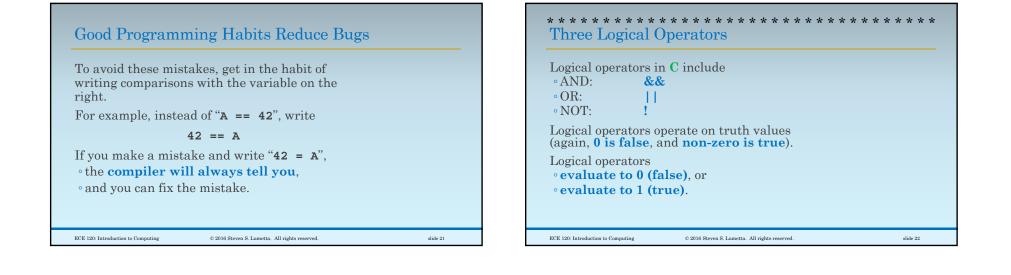
\* For ECE120. ECE220 teaches other ways to use the assignment operator.

© 2016 Steven S. Lumetta. All rights reserved.

. All rights reserved.

slide 19





Declare: int A = 120;	int $B = 42;$
Then	
(0 > A    100 < A)	evaluates to 1
(120 == A && 3 == B)	evaluates to <b>0</b>
!(A == B)	evaluates to 1
!(0 < A && 0 < B)	evaluates to <b>0</b>
(B + 78 == A)	evaluates to 1
(So no bitwise calculatio	ons, just true/false.)

A task for you:		
Evaluate the	C expression: 1 + 2 * 3	
Did you get 7?		
Why not 9? (1 +	2) * 3	
Multiplication co • in elementary s • and in <b>C</b> !	mes before addition school	
The order of oper precedence.	rations is called operator	

## Never Look Up Precedence Rules!

Another task for you: Evaluate the C expression: 10 / 2 / 3 Did you get 1.67? Is it a friend's birthday? Perhaps it causes a divide-by-0 error? Or maybe it's ... 1? (10 / 2) / 3, as int If the order is not obvious, • Do NOT look it up. • Add parentheses!