

Programming Languages and Tools: Programming with C++ CS:3210:0003

Lecture/Lab #4

Operators

- Applied to **operands**
- *Fixedness*: Infix vs prefix vs postfix
 $x + y$ $+ x y$ $x y +$
- *Arity*: how many operands? Unary, binary, ternary, etc.
- *Type*: type of operands, return type

Operators

Operator	Description	Fixedness	Arity	Input Type	Return Type
=	Assignment	Infix	Binary	Any Type T	T
+, -, *, /, %	Arithmetic	Infix	Unary, Binary	Ints, Floats	Ints, Floats

Operators

Operator	Description	Fixedness	Arity	Input Type	Return Type
=	Assignment	Infix	Binary	Any Type T	T
+, -, *, /, %	Arithmetic	Infix	Unary, Binary	Ints, Floats	Ints, Floats
++, --	Increment, Decrement	Prefix/Postfix	Unary	Int	Int

Increment/Decrement

- ++ to increase value of operand by 1
- -- to decrease value of operand by 1
- Prefix: increment *before* operation
- Postfix: increment *after* operation

Relational Operators

Operator	Description	Fixedness	Arity	Input Type	Return Type
=	Assignment	Infix	Binary	Any Type T	T
+, -, *, /, %	Arithmetic	Infix	Unary, Binary	ints, floats	ints, floats
++, --	Increment, Decrement	Prefix/Postfix	Unary	int	int
==, !=, <, >, <=, >=	Relational	Infix	Binary	ints, floats	bool

Logical Operators

Operator	Description	Fixedness	Arity	Input Type	Return Type
=	Assignment	Infix	Binary	Any Type T	T
+, -, *, /, %	Arithmetic	Infix	Unary, Binary	ints, floats	ints, floats
++, --	Increment, Decrement	Prefix/Postfix	Unary	int	int
==, !=, <, >, <=, >=	Relational	Infix	Binary	ints, floats	bool
!	NOT	Prefix	Unary	bool	bool
&&, , ^	AND, OR, XOR	Infix	Binary	bools	bool

Logical Operators

Operator	Description
NOT x	true if x is false
x AND y	true if x and y are true
x OR y	true if at least one of x and y are true
x XOR y	true if exactly one of x and y are true

- Truth tables enumerate result of operator for all possible values of operands

Truth Tables

Activity

Add functions `ortable()` and `xortable()` to print truth tables of `||` and `^` and call them from `main`

Branching

- Use if ... else to execute statements conditionally
 - if (conditional-expression)
 - do something when exp evaluates to true;
 - else
 - do something when exp evaluates to false;
- Ability to modify program *flow*
- Can have more than 2 branches by nesting if else

Activity

Create 09ifelse3.cpp that inputs 2 integers i and j, and

- prints "Positive numbers\n" if *both* of them are positive
- prints "Negative numbers\n" if *both* of them are negative
- prints "Mixed\n" otherwise