EECS 1022 P Jan 18 Lecture

Yves Lesperance

Do This Week

- Quiz #1 on Wed at 10:30, see announcement on Moodle site.
- Lab Audio Generation: prepare, do pre-lab quiz, go to your lab section, and do report
- Read and digest lecture material

Relational Operators

< <= > >=   Numeric operands

== !=   Operands of any type

All relational operands are "odd" in that they violate closure: no matter what the operand type is, the result type is always **boolean**.

Operator Precedence

Precedence	Operator	Operands	Syntax	true if
-7 →	<	numeric	x < y	x is less than y
	<=	numeric	x <= y	x is less than or equal to y
	>	numeric	x > y	x is greater than y
	>=	numeric	x >= y	x is greater than or equal to y
	instanceof			x instanceof C is true if object reference x points at an instance of class C or a subclass of C.
-8 →	==	any type	x == y	x is equal to y
	!=	any type	x != y	x is not equal to y

Boolean Expressions

- **Relational Expression**
  `(k < 0)`

- **Boolean Variable**
  `boolean b = k < 0`

- **Boolean Expression**

  `(k < 0 || b && m > h)`

  Uses boolean operators: &&, ||, and !

Operator Precedence

```
. ++ -- !
cast
* / %
+ -
<  <=  >  >=
== !=
&&
||
= op=
```

2
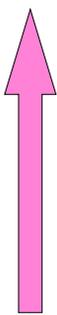
Examples

- Express the condition $x \in [a, b)$

- Express the condition $x \notin [a, b)$

Examples

- Express the condition $x \in [a, b)$

  `(x >= a && x < b)`

- Express the condition $x \notin [a, b)$

  `(!(x >= a && x < b))`

  Can use deMorgan's Law to convert negated conjunctions to disjunctions.

  `(x < a || x >= b)`

Operations on `char` data

- Can use integer operators with `char` data, e.g.

```
char let1 = 'D';
char let2 = (char) (let1 + 1);
```

- Here `let1` is promoted to `int` when 1 is added and must be cast back to `char`.
- What is the value of

```
(let1 – 'A' + 1)?
```

- `char` has no operators so both arguments of – are promoted to `int`.