LOGICAL EXPRESSIONS

Writing the "Test" part of control statements

Delroy A. Brinkerhoff

TWO KINDS OF OPERATORS

RELATIONAL

- == Equal to
- != Not equal to
- < Less than
- <= Less than or equal to
- > Greater than
- >= Greater than or equal to

LOGICAL

- && Logical AND
- || Logical OR
- ! Logical Not

UNDERSTANDING LOGICAL OPERATORS

LOGICAL AND

E1	E2	E1 && E2
F	F	F
F	T	F
T	F	F
T	T	T

LOGICAL OR

E1	E2	E1 || E2
F	F	F
F	T	T
T	F	T
T	T	T

LOGICAL NOT

E	! E
F	T
T	F

SHORT CIRCUIT EVALUATION

LOGICAL AND

- `if (E1 && E2)`
- If E1 is false, E1 && E2 is false regardless of the value of E2
- E2 is NOT evaluated
- `if (n != 0 && 100 / n > min)`

LOGICAL OR

- `if (E1 || E2)`
- If E1 is true, E1 || E2 is true regardless of the value of E2
- E2 is NOT evaluated
-
```
if (s == nullptr || s->length() == 0)
    return;
```

SHORT CIRCUIT EVALUATION (2)

C++

```
if (s == nullptr || s->length() == 0)
        return;
```

- To translate to Java, change
  - `nullptr` to `null`
  - `->` to `.`

JAVA

```
if (s == null || s.length() == 0)
        return;
```

- If s is null, then s == null is true
- `if (true || s.length() == 0)`

BOOLEAN TYPE: TRUE AND FALSE

- In C++ 0 is false and non-0 is true
- That means that numeric types and expressions can be used in control statements
- The bool data type, and true and false are a syntactic candy coating for ints (false = 0 and true = 1)

```
if (n % 2)        // n > 0
    cout << "n is odd\n";
else
    cout << "n is even\n";
```

A COMMON ERROR

INCORRECT

```
if (counter = 10)
    . . .
```

CORRECT

```
if (counter == 10)
    . . .
```

A COMMON ERROR

INCORRECT

```
if (counter = 10)
    . . .
```

CORRECT

```
if (counter == 10)
    . . .


boolean running;
if (running = false)
```