CSI31
Introduction to Computer Programming I

Dr. Sharon Persinger

November 14, 2018

Topics

- Operations with Boolean expressions
- Boolean values of all data types
- Short-circuit evaluation

Boolean expressions

- A Boolean expression is an expression that evaluates to either True or to False.
- Common expressions use comparison operators: ==, !=, < , <=, > >=
- Examples

Boolean operators

- and:

p	q	p and q
True	True	True
True	False	False
False	True	False
False	False	False

- or

p	q	p or q
True	True	True
True	False	True
False	True	True
False	False	False

- not

p	not p
True	False
False	True

- Python has these operators.

Build up complex Boolean expressions

- p and not q or s
- q or r and s
- a and b or not a and c

- What is the order of operations?
  - not, then and then or
- Use parentheses to prevent confusion

Examples

- An expression that is True when the two Point objects p1 and p2 are equal.
- An expression that is True when x is equal to 0 and y is not equal to 0.
- An expression that is True when x is greater than 10 or y is greater than 10.
- An expression that is True when x is greater than 10 or y is greater than 10, but not both.
- An expression that is True when the two Point objects p1 and p2 are not equal.

Boolean Algebra Identities

- a and False == False
- a and True == a
- a or True == True
- a or False == a
- not (not a) == a
- a or (b and c) ==

(a or b) and (a or c)

- a and (b or c) ==

(a and b) or (a and c)

- DeMorgan’s Laws
- not(a or b) ==

(not a) and (not b)

- not(a and b) =

(not a) or (not b)

Use deMorgan's Law

- To rewrite the expression that is True when the two Point objects p1 and p2 are different.

Some details about Boolean operators

- First, any data type can be used as a Boolean expression.
  - bool(1)? bool(7.0)? bool(0)?
  - bool(‘x’)? bool(‘’)? bool(‘abc’)?
  - bool([])? bool([3, 4, 5])? bool([0])?
  - How is the value calculated?

How Boolean operators are evaluated

operator	Operational definition
x and y	If x is False, return x. Otherwise, return y.
x or y	If x is True, return x. Otherwise, return y
not x	If x is False, return True. Otherwise, return False.

Examples?

and, or are both short-circuit operators. A value is returned as soon as it is known.

An infinite loop and why

- While response[0] == ‘y’ or response[0] == ‘Y’:
- To control an interactive loop
- What if instead we used
- While response[0] == ‘y’ or ‘Y’:   ?

- Why does this happen?

Some interesting examples

```
ans = input("What flavor do you want[vanilla]?"
if ans != '':
    flavor = ans
else :
    flavor = 'vanilla'
```

```
ans = input("What flavor do you want[vanilla]?"
if ans :
    flavor = ans
else :
    flavor = 'vanilla'
```

Some interesting examples

```
ans = input(“What flavor do you want[vanilla]?”
flavor = ans or ‘vanilla’
```

```
flavor = input(“What flavor do you want[vanilla]?” or ‘vanilla’
```