

CSE 20 4-28-20

11

chapter 5 ... continue

comparison operators ✓

logical operators

math names

and
or
not

conjunction
disjunction (incl. or)
negation

let A, B be bool expressions

Truth tables

A	B	$A \text{ and } B$
T	T	T
T	F	F
F	T	F
F	F	F

(binary op)

A	B	$A \text{ or } B$
T	T	T
T	F	T
F	T	T
F	F	F

(binary op)

A	not A
F	T
T	F

← unary op.

Identities:

$$\begin{cases} A \text{ and } B == B \text{ and } A \\ A \text{ or } B == B \text{ or } A \end{cases}$$

$$\begin{cases} A \text{ and } (B \text{ or } C) == (A \text{ and } B) \text{ or } (A \text{ and } C) \\ A \text{ or } (B \text{ and } C) == (A \text{ or } B) \text{ and } (A \text{ or } C) \end{cases}$$

$$\begin{cases} \text{not } (A \text{ and } B) == (\text{not } A) \text{ or } (\text{not } B) \\ \text{not } (A \text{ or } B) == (\text{not } A) \text{ and } (\text{not } B) \end{cases}$$

or and

De Morgan's Laws

Proof of 1st deMorgan

4

$$\text{not}(A \text{ and } B) \\ ==$$

A	B	A and B	not A	not B	((not A) or (not B))
T	T	T	F	F	F
T	F	F	F	T	T
F	T	F	T	F	T
F	F	F	T	T	T

more identities:

$$\begin{cases} A \text{ and } \text{False} == \text{False} \\ A \text{ or } \text{True} == \text{True} \end{cases}$$

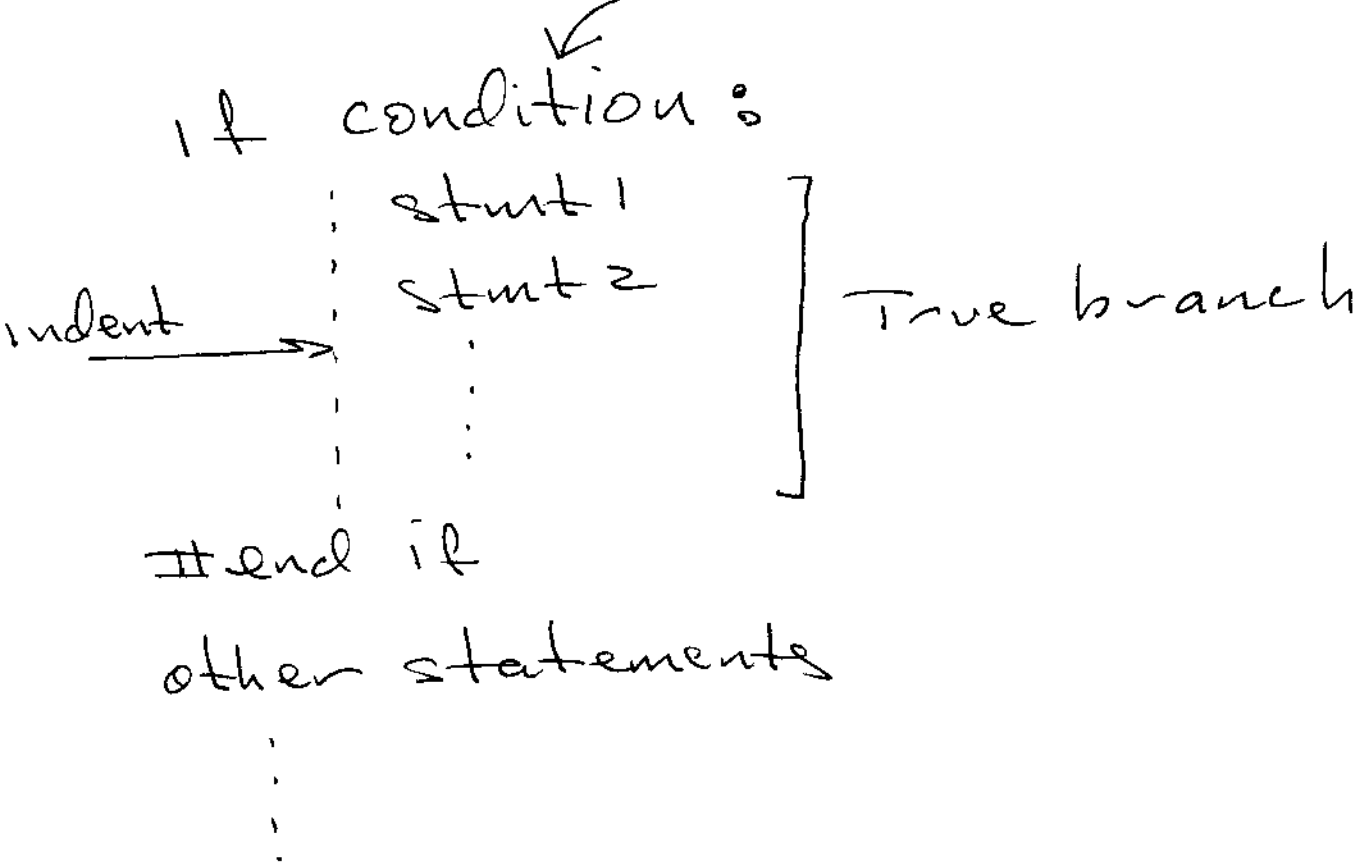
$$\begin{cases} A \text{ or } \text{False} == A \\ A \text{ and } \text{True} == A \end{cases}$$

$$\text{not}(\text{not } A) == A$$

conditional operations (branching)

allow us to change flow of execution during a program.

- if statement.



◦ if with else clause

```

      ↖----- bool expression
if cond :
    statements ] true branch
else :
    statements ] false branch
#end if-else
other statements
    
```

- chained conditionals
(if-elif-else)

```

if cond1:
    statements
elif cond2:
    statements
elif cond3:
    statements
else:
    statements
# end if-elif-else
statements
    
```

} optional

Ex, Sort3.py

Ex, BetterSort.py

to swap x with y :

$$\begin{cases} \text{temp} = x \\ x = y \\ y = \text{temp} \end{cases}$$

<u>x</u>	<u>y</u>	<u>temp</u>
7	6	7
6	7	