# Sept 7 Slides

 $\Box A U \varnothing =$  $\Box A U U =$  $\Box A U A =$ 

 $\Box A U \varnothing = A$  $\Box A U U =$  $\Box A U A =$ 

Identity law

 $\Box A U \varnothing = A$  $\Box A U U = U$  $\Box A U A =$ 

Identity law Domination law

 $\Box A U \varnothing = A$  $\Box A U U = U$  $\Box A U A = A$ 

Identity law Domination law Idempotent law

 $\Box A U \varnothing = A$  $\Box A U U = U$  $\Box A U A = A$  $\Box A U B = B U A$ 

Identity law Domination law Idempotent law Commutative law

 $\square A \cap U =$ 

#### □ A ∩ Ø =

□ A ∩ A =

 $\Box A \cap U = A$ 

Identity law

□ A ∩ Ø =

 $\square A \cap A =$ 

 $\Box A \cap U = A$ 

 $\Box A \cap \emptyset = \emptyset$ 

Identity law

**Domination law** 

 $\square A \cap A =$ 

 $\Box A \cap U = A$ 

 $\Box A \cap \emptyset = \emptyset$ 

 $\Box A \cap A = A$ 

Identity law

**Domination law** 

Idempotent law

- $\Box \mathsf{A} \cap U = \mathsf{A}$
- $\Box A \cap \emptyset = \emptyset$
- $\Box A \cap A = A$
- $\Box A \cap B = B \cap A$

Identity law

**Domination law** 

Idempotent law

Commutative law

## **Propositions** A proposition is a statement that is either true or false

**Examples of Proposition** 

(Eggs are blue) = *p* 

(I am a human) = q

(2 + 3 = 5) = r

Examples of things that aren't Proposition

What are you doing Friday?

What is 3 + 3?

Sit down!

# A proposition is a statement that is either true or false

When dealing with propositions, we abstract away difficulties of defining, and we can just give them letters (define variables), like *p* 

A proposition, *p*, is a statement that is either true or false. "True" or "False" is considered the "truth value" of *p*.

https://www.cs.virginia.edu/luther/2102/F2020/symbols.html

Concept	Java/C	Python	This class	Bitwise	Other
true	true	True	op or $1$	-1	T, tautology
false	false	False	$\perp$ or $0$	0	F, contradiction

A proposition is a statement that is either true or false

A proposition is a statement that is either true or false

- V is "or"
- $\Lambda$  is "and"
- ¬ is "not"

- $\bullet$  V is "or"
- ∧ is "and"
- ¬ is "not"

$$= \{ x \in U \mid x \in S \land x \not\in T \}$$

We can modify, combine and relate propositions with *connectives:* 

- V is "or"
- $\Lambda$  is "and"
- ¬ is "not"

# $S \setminus T = \{ x \in U \mid x \in S \land x \notin T \}$

- $\bullet$  V is "or"
- ∧ is "and"
- ¬ is "not"

$$= \{ x \in U \mid x \in S \lor x \in T \}$$

We can modify, combine and relate propositions with *connectives:* 

- $\bullet$  V is "or"
- $\Lambda$  is "and"
- ¬ is "not"

# $S \cup T = \{x \in U \mid x \in S \lor x \in T\}$

We can modify, combine and relate propositions with *connectives:* 

- $\bullet$  V is "or"
- $\Lambda$  is "and"
- ¬ is "not"

# $= \{ x \in U \mid x \in S \land x \in T \}$

We can modify, combine and relate propositions with *connectives:* 

- $\bullet$  V is "or"
- $\Lambda$  is "and"
- ¬ is "not"

# $S \cap T = \{ x \in U \mid x \in S \land x \in T \}$

We can modify, combine and relate propositions with *connectives:* 

- V is "or"
- $\Lambda$  is "and"
- ¬ is "not"

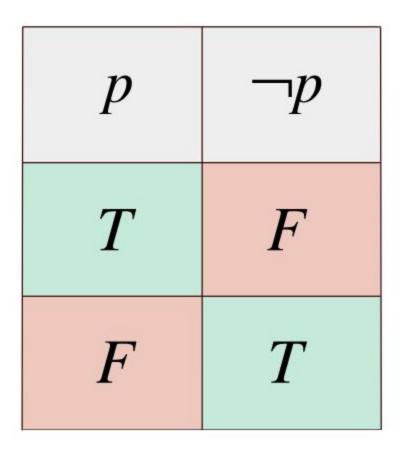
Set theory is a branch of mathematical logic. So it makes sense to use logical language and symbols to describe sets.

#### "Not" operator

#### How to define:

#### Make a truth table

#### "Not" operator



#### "And" operator

#### "And" operator

p	q	$p \wedge q$
Т	Т	Т
Т	F	F
F	Т	F
F	F	F

#### "Or" operator

#### "Or" operator

р	q	$p \lor q$
Т	Т	Т
Т	F	Т
F	Т	Т
F	F	F

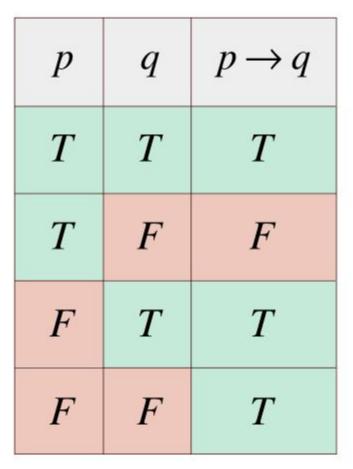
#### "Implies" operator

## If p, then q

The conditional  $p \rightarrow q$  can be expressed by different sentences, some of them are listed below:

- p implies q
- p is a sufficient condition for q
- q is a necessary condition for p
- q follows from p
- p only if q

#### "Implies" operator



Supose A  $\subset$  B and B  $\subseteq$  C are true, and B  $\subseteq$  D and D  $\subseteq$  B are both false.

For each of the following, decide if it

- must, could, or can't be empty - how it must relate ( $\subseteq$ ,  $\subset$ ,  $\supseteq$ ,  $\supset$ , =) to the four named sets (if any)

0

a. A n B

- b. A u B
- c. B n C
- d. B u C
- e. B n D
- f. B u D

g. C n D

h. C u D i. A n C j. A u C k. A n D l. A u D