

## Boolean Logic

### A. AND

- a. Both conditions need to be true.
- b. Truth Table:

A	B	A && B
true	true	true
true	false	false
false	true	false
false	false	false

- c. Symbol in Java is &&
- d. Example: You can only vote in an American presidential election if you are over 18 and an American Citizen. If you are a 10 year old American, you can't vote. Also, if you are a 30 year old from Tanzania, you also can't vote. You can vote only if both conditions are met.

```
boolean mayVote=false;
if (age>=18 && citizenship.equals("American"))
{
    mayVote=true;
}
```

- e. Note: you cannot write  $10 < x < 100$  as you would in math. You must write  $(x > 10 \ \&\& \ x < 100)$

### B. OR

- a. One of the conditions must be true.
- b. Truth table:

A	B	A    B
true	true	true
true	false	true
false	true	true
false	false	false

- c. Symbol in Java is ||
- d. Example: To graduate from college you must take general education classes outside your major. For each required class (example: US history), you must have taken a US history class in college, or passed the US history AP test.

```
boolean apCredit, collegeClass; //set earlier
boolean fulfilledRequirementA = false;
if (apCredit || collegeClass)
{
    fulfilledRequirementA=true;
}
```

### C. Lazy Evaluation

- a. For both AND and OR, the conditions are checked from left to right and java stops checking once it has determined the truth value of the statement.
  - i. If you have apCredit above, once java finds that is true, it knows the OR is also true and stops.
  - ii. If you are less than 18, the AND knows that the entire AND will be false so it stops checking.

D. NOT

- a. Negate the input A.
- b. Truth table:

A	!A
true	false
false	true

- c. Symbol in Java is !
- d. Example we have seen before: inequality checks

```
int number; //already set
if(number % 5 != 0) {
    boolean multipleOfFive = false;
}
```

- e. Use the not sign on larger expressions. If the score on the test is less than 60 percent then they fail:

```
if ( !(grade > 60) ) {
    System.out.println("You failed this class.");
}
```

E. When using boolean Variables write if(booleanVar) rather than if (booleanVar==true)

**Logic Rules**

Commutative Law	A    B = B    A A && B = B && A
Rules of Logical Negation	A    true = true A && false = false  A    false = A A && true = A
Complements	A    !A = true A && !A = false  A    A = A A && A = A
Absorption	A    (A && B) = A A && (A    B) = A
Associative Law	A    (B    C) = (A    B)    C A && (B && C) = (A && B) && C
Distributive Law	A && (B    C) = (A && B) && (A && C) A    (B && C) = (A    B) && (A    C)
De Morgan's Law	!(A && B) = !A    !B !(A    B) = !A && !B

Proof of De Morgan's Law with a truth table

A	B	A    B	!(A    B)	!A	!B	!A && !B
false	false	false	true	true	true	true
false	true	true	false	true	false	false
true	false	true	false	false	true	false
true	true	true	false	false	false	false

