Set 7: Boolean Operators

Skill 7.1: Evaluate true and false statements

Skill 7.1 Concepts

So far, we have looked at three fundamental variable types… int, double, and String. In this lesson we will look at another very important type of variable, "Boolean". This type of variable has only two possible values, true or false.
Consider the following example. Suppose that we know that x = 3 and y = 97. Below are statements about x and y, and whether or not they are true or false individually, AND whether the entire statement is true or false.

Statement	True or False
`(( x < 10 ) AND ( y = 97 )`	First part is true, second part is true, entire statement is true
`(( x < 10 ) AND ( y = -3 )`	First part is true, second part is false, entire statement is false
`(( x < 10 ) AND ( y ≠-3 )`	First part is true, second part is true, entire statement is true
`(( x < 10 ) OR ( y = 97 ))`	Either part is true, entire statement is true
`(( x < 10 ) OR ( y = -3 ))`	Either part is true, entire statement is true

Skill 7.1 Exercise 1

Skill 7.2: Evaluate true and false statements using proper java syntax

Skill 7.2 Concepts

The above examples illustrate how true and false statements are evaluated, however the syntax is not correct. In order for java to correctly read the syntax the "AND" and "OR" statements must be replaced with the correct symbols as shown below,

Statement	Java syntax explanation
`(( x < 10 ) && ( y == 97 )`	*And* is replaced with &&, = is replaced with "=="
`(( x < 10 ) && ( y == -3 )`	*And* is replaced with &&, = is replaced with "=="
`(( x < 10 ) && ( y != -3 )`	*And* is replaced with &&, ≠ is replaced with "!="
`(( x < 10 ) || ( y == 97 ))`	*OR* is replaced with ||, = is replaced with "=="
`(( x < 10 ) || ( y == -3 ))`	*OR* is replaced with ||, = is replaced with "=="

Skill 7.2 Exercise 1

Skill 7.3: Use truth tables to evaluate whether a statement is true or false

Skill 7.3 Concepts

Truth tables are helpful for showing how && and || work for various combinations. These are illustrated below,

a	b	(a && b)	a	b	(a || b)
false	false	false	false	false	false
false	true	false	false	true	true
true	false	false	true	false	true
true	true	true	true	true	true

Skill 7.3 Exercise 1

Skill 7.4: Apply the *not* operator

Skill 7.4 Concepts

The not (or negation) operator is used to indicate opposite. For example, what is the opposite of true?... False of course. The "not" operator is indicated with the ! sign. Below are some examples.

Code	Output
`System.out.println(!true);`	False
`System.out.println(!false);`	True
`System.out.println(!(3<5));`	False
`System.out.println(!(1==0));`	True

Skill 7.4 Exercise 1

Skill 7.5: Create Boolean variables

Skill 7.5 Concepts

Boolean variables are denoted with the word “boolean” and can be assigned a value that evaluates to true or false. Below are some examples,

Code	Output
`int p = 2;`	False
`int j = 3;`	True
`int x = 1;`	False
`int c = 0;`	True
`boolean a = false;`	
`boolean b = true;`	
`boolean z = (( p < j ) && ( x != c));`	
`System.out.println(a);`	
`System.out.println(b);`	
`System.out.println(z);`	

Skill 7.5 Exercise 1

Skill 7.6: Apply operator precedence

Skill 7.6 Concepts

Just like math operations follow an order of precedence, so too do operations like AND (&&) and OR (||). Consider a problem like,

```
System.out.println( ( true && false ) || ((true && false) || false) );
```

In this example we can tell what parts we should do first because of the grouping by parenthesis. However, what if we had a different problem like the one below?

```
System.out.println( false && true || true );
```

Which part do we do first? As it turns out we would first do && and then ||. Because false and true are not the same thing, false && true evaluates to false. Next we consider false or true, which is true. The order of precedence for the operators we are studying are as follows,

! == != && ||

The order of precedence for operators is illustrated in the following examples,

Code	Output	Explanation
`System.out.println( false || true && false );`	**False**	Do the true && false part first to get a result of false. Then do false || false to get a final result of false
`System.out.println( true || false && false );`	**True**	Do the false && false part first to get a result of false. Then do true || false to get a final result of true

Skill 7.6 Exercise 1

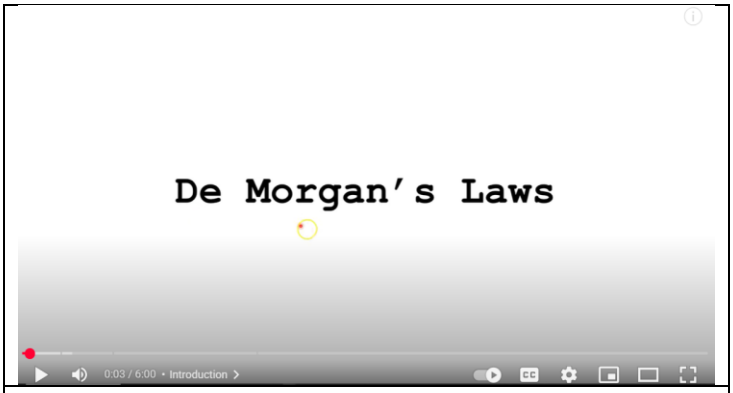
Skill 7.7: Apply De Morgan's Law

Skill 7.6 Concepts

DeMorgan's Theorems are basically two sets of rules or laws developed from the Boolean expressions for AND, OR and NOT using two input variables, A and B. These two rules or theorems allow the input variables to be negated and converted from one form of a Boolean function into an opposite form.

DeMorgan's first theorem states that two (or more) variables NOR'ed together is the same as the two variables inverted (Complement) and AND´ed, while the second theorem states that two (or more) variables NAND'ed together is the same as the two terms inverted (Complement) and OR´ed. That is replace all the OR operators with AND operators, or all the AND operators with an OR operators.

The video below explains this more clearly.



https://www.youtube.com/watch?v=AGyjo2DLxjM&t=2s

Skill 7.7 Exercises 1 thru 3