Lesson 3

Intro to Logic

Today's Lesson Plan

1. Review boolean concept
1. Relational operators (<=)
1. Logical operators(&&)
1. If statements
1. For loops

Boolean Review

Boolean variables are variables that hold true/false information.

For example,

```
boolean hasCandy = true;

boolean notHappy = false;
```

Relational Operators

Relational operators are symbols that compare the value of two numbers.

== (equal to)	!= (not equal to)
>= (greater than or equal to)	<= (less than or equal to)
> (greater than)	< (less than)

It is not recommended to use these operators on String type variables.

Relational Operators

Examples:

`5 == 3` is false

`1 != 4` is true

`90 >= 90` is true

`4 <= 8` is true

`60 > 80` is false

`100 < 50` is false

Logical Operators

Logical operators compare boolean values. Some examples are `&&` (and), `||` (or), and `!` (not).

`&&` - **And**: both sides need to be true for the whole expression to be true, otherwise it is false.

`||` - **Or**: one side needs to be true for the expression to be true, it is only false if both sides are false.

`!` - **Not**: true becomes false, false becomes true

Logical Operators cont.

&&	**true**	**false**
true	true	false
false	false	false

||	**true**	**false**
true	true	true
false	true	false

```
boolean hasCandy = true;
boolean notHappy = false;
```

`hasCandy && notHappy` is false

`hasCandy || notHappy` is true

If and else statements

If and else statements can execute code if a certain condition is true.

```
if(isRaining == false){

    System.out.println(“What a beautiful day!”);

}else{

    System.out.println(“Grab an umbrella!”);

}
```

If statements - Try it out

Write the following code in the program:

```
if(4 > 5) {
    System.out.println(“4 is greater than 5!”);
} else {
    System.out.println(“4 is not greater than 5!”);
}
```

You can change the condition or outputs of any part of the code.

For Loops

For loops repeatedly run statements a certain number of times

Initialize variable, condition, increment

```
for(int i = 0; i < 10; i++){  starting point = 0

                              Ending point = 9

    System.out.print(i);

}
```

The condition is checked every time before running the loop. If the condition is true, the loop will continue.

Backward For Loops

This is the same thing as the normal for loop, except it moves in the opposite direction.

Initialize variable, condition, increment

```
for(int i = 10; i > 0; i--){

    System.out.print(i + " ");

}
```

This code will print "10 9 8 7 6 5 4 3 2 1 "

For Loops - Try it out

Write the following code in the program to test out for loops:

```
for(int i = 0; i < 10; i++) {
    System.out.println(i);
}
```

You can change the condition or outputs of any part of the code. Play with the for loop a little to understand it more.

Questions?

Contact us here:

Email:
codingpower101@gmail.com
Website:
codingpower.org