Notes (Day 8)

- Introduce Java Decision Code: **Ifs**
- Demos
  - Bounce-a-ball
  - Arrow-Paddle Game

First: IW2

```
// Suppose that x and y are int variables
int lg = Math.max(x, y);
int sm = Math.min(x, y);

// If x is larger, then after the code above
// runs, lg is equal to x and sm is equal to y

// If y is larger, then the opposite occurs

// Should come in handy for problem #5
```

Java Decisions

Concept: running code only on certain conditions

Example: login - hide private information until password entered

Format

```
if (userPass == actualPass) {
  // allow login, show info
}
```

Comparison Operators

```
if (x == y) // equal to
if (x != y) // not equal to
if (x > y)  // greater
if (x < y)  // less
if (x >= y) // greater/equal
if (x <= y) // less/equal
```

Boolean Values & Logic Review

In Integer value is a whole number ... -3, -2, -1, 0, 1, 2, 3, ...

You compute on integers with `+ - * / %`

A Boolean value is one that is either `true` or `false`

You compute on booleans with...

- `&&` (and)
- `||` (or)

Example

Suppose `b` and `c` are Boolean values.

Suppose `b` has the value `true`.

Suppose `c` has the value `false`.

Then

(`b && c`) evaluates to `false`

(`b || c`) evaluates to `true`

In General

`(b && c)`

evaluates to `true` only when both `b` and `c` are separately `true`

evaluates to `false` when one or both are `false`

`(b || c)`

evaluates to `true` when either `b` or `c` (or both) are `true`

evaluates to `false` when both `b` and `c` are separately `false`

Caution

In everyday English, we frequently use the words *and* and *or* to mean different things than what we mean in Java

Specifically, in English we often use *or* to mean "one of two options, but not both options" (an *exclusive* or).

In Java, we use *or* to mean "any one of many options, possibly more than one, possibly all options" (an *inclusive* or).

Who Cares?

In Java, when you decide to run code when any one of many conditions are true, you use `||` for computation.

When you decide to run code only when all of many conditions are true, you use `&&` for computation.

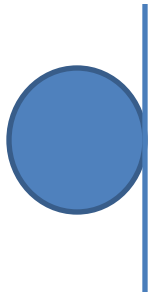
Textbook Guidance

Chapter 5, p. 179-221

Demo

Bounce-a-Ball

Change direction when the ball is on **any** wall (||)



Arrow-Paddle Game

Score a point when the lead point on the arrow is between **all** the sides of the paddle (&&)