Opening Discussion

- Minute essay comments
  - Board vs. computer was split.
  - Calling functions.
  - Why format pay returns a string.
  - False is a keyword.
- Everything moved back.
Comparisons

- The condition needs to be a Boolean expression.
- The most common basic forms of these are comparisons.
- Use == and != for equality and non-equality.
- The ordering comparison operators are <, >, <=, and >=.
Let's write the code for the skating problem example.

Guarding division is another example.

What are some other simple examples of places where conditional execution would be helpful?
Motivation

- I want to have a function that tells me if two squares intersect.
- The function will be given the x and y location of the center of each square as well as the length of the side of each square.
- It should return a Boolean telling if they intersect.
We talked about comparisons of values in the last class.

We can also combine Boolean expressions together using Boolean logic.

There are four Boolean operators:

- `&&` for and
- `||` for inclusive or
- `^` for exclusive or
- `!` for not
The && and || operators are short circuit operators.

This means that if the value is known after evaluating the first operand, the second operand won't be evaluated.

This can prevent errors.

Let's look at an example of this with division by zero.
Nesting ifs

- What you put in an if can be any expression or statement.
- As a result, you can put an if inside of another if.
- As we will see, Scala doesn't care what you nest inside of things. You write the logic that makes sense to you and says what you want to say.
There is a second conditional expression in Scala called match.

\[ expr \text{ match } \{
\begin{align*}
\text{case } & \text{pattern } \Rightarrow \text{expr} \\
\text{case } & \text{pattern } \Rightarrow \text{expr} \\
& \ldots
\end{align*}
\} \]

There are lots of options for the pattern, but the simplest one is literal values.
When might you want to use conditional execution? (Hint: any time you would use if in a sentence probably works.)