Nested `if`

A nested `if` statement is an **if-else** statement with another **if** statement as the **if body** or the **else body**.

Here's an example:

```
if ( num > 0 )        // Outer if
  if ( num < 10 )   // Inner if
    System.out.println( "num is between 0 and 10" ) ;
```

There is an *outer* **if** statement, and an *inner* **if**.

What does Java do when it runs this code?

- Evaluates the condition of the *outer* if. If it evaluates to **false**, don't run the code in the **if body** (which is the inner **if**).
- If the outer **if** condition evaluates to **true**, evaluate the outer **if** condition. If it evaluates to **true**, run *its* **if body** (the **println()** statement).

So basically it evaluates the outer condition, and only when it succeeds does it evaluate the inner condition.

Equivalent to `&&`

Does this sound familiar? It's just **&&** with short-circuiting. We can rewrite the code as:

```
if ( num > 0 && num < 10 )
    System.out.println( "num is between 0 and 10" ) ;
```

Variations of nested if

We can also have an inner **if** statement that is contained in the **if body** of an outer **if** but isn't the **if** body. Here's an example:

```
if ( num > 0 )        // Outer if
{
  System.out.println( "Got here" ) ;
  if ( num < 10 )   // Inner if
    System.out.println( "num is between 0 and 10" ) ;
}
```

In this case, the ***if body*** of the outer **if** contains two statements, one of which is the inner ***if body***.

This can't be rewritten using **&&** because the first **println()** statement only depends on the outer condition.

Nested in `else` body

When an **if** statement is in the ***if body*** then both the outer **if** condition and the inner **if** conditions must evaluate to **true** before the inner ***if body*** can run.

What happens if an **if** statement appears in the **else** body?

```
if ( num > 90 )
{
  System.out.println( "You earned an A" ) ;
}
else
   if ( num > 80 )
   {
      System.out.println( "You earned a B" ) ;
   }
```

The statement "**You earned a B**" only if the outer **if**condition evaluates to **false** and the inner **if** condition evaluates to **true**. In other words, the first **if** condition has to "fail" before it's even possible that the second **println()** prints.

The first condition failing means that **num** is not greater than 90, so it must be less than or equal to 90. The second condition succeeding means that **num** is greater than 80. Combine the two and you get **num** greater than 80, but less than or equal to 90.

We can add yet another **if** statement to the **else** body, to take this one step further.

```
if ( num > 90 )
{
  System.out.println( "You earned an A" ) ;
}
else
   if ( num > 80 )
   {
      System.out.println( "You earned a B" ) ;
   }
   else
      if ( num > 70 )
      {
         System.out.println( "You earned a C" ) ;
      }
```

In this case, you don't see **"You earned a C"** unless the top condition fails (i.e. **num > 90**), and the middle condition fails (i.e. **num > 80**), but the bottom condition succeeds (i.e. **num > 70**).

If you combine all the conditions together need to print **"You earned a C"**, you get something that is less than or equal to 80, and greater than 70.

You should notice that this kind of structure goes through each condition one at a time. When it reaches the first condition that evaluates to **true**, it runs the body, and then skips everything else.

This is called a *prioritized if*. We'll see this more in an upcoming lesson.

Can you triple nest?

Why yes you can. You can quadruply nest, or even more. This is an example of a triply nested **if**.

```
if ( num > 0 )
  if ( num < 10 )
    if ( num % 2 == 0 )
      System.out.println( "num is between 0 and 10 and
even" ) ;
```

As mentioned earlier, you can often use logical operators to avoid some forms of nesting.

Practice exercises

Write a program that takes three integer command-line arguments and prints `equal` if all three are equal, and `not equal` otherwise.

Write a more general and robust version of Quadratic.java that prints the roots of the polynomial $ax^2 + bx + c$, prints an appropriate error message if the discriminant is negative, and behaves appropriately (avoiding division by zero) if *a* is zero.

Determine (if anything) is wrong with each of the following statements?

```
if (a > b) then c = 0;
if a > b { c = 0; }
if (a > b) c = 0;
if (a > b) c = 0 else b = 0;
```

Write a code fragment that prints true if the double variables x and y are both strictly between 0 and 1 and false otherwise.

What does the following code fragment write to the monitor?

```
int sum = 14;
if ( sum < 20 )
 System.out.print("Under ");
else
{
 System.out.print("Over ");
 System.out.println("the limit.");
}
```

What does the following code fragment write to the monitor?

```
int sum = 7;
if ( sum > 20 )
{
 System.out.print("You win ");
}
else
{
 System.out.print("You lose ");
}
System.out.println("the prize
```