Logical expressions

There are three logical operators in Java that can be applied to `boolean` expressions:

`&&` logical AND

`||` logical OR

`!` logical NOT (unary operator)

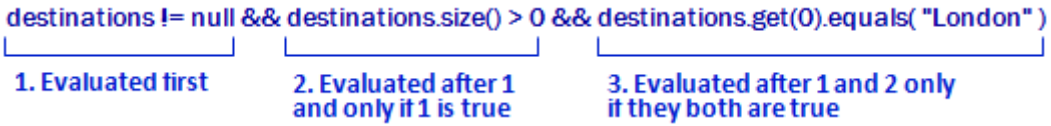
AND has higher priority than OR, so that

```
a || b && c ≡ a || ( b && c )
```

but again, it is always better to put parentheses to explicitly define the order of operations.

The logical operations in Java exhibit so-called *short-circuiting behavior*, which means that the second operand is evaluated only if needed.

This feature is very useful when a part of an expression cannot be evaluated (will cause error) if another part is not true. For example, let `destinations` be a collection of places an agent in the model must visit. To test if the first place to visit is London, you can write:



Short-circuiting behavior of logical operations

Here we first test if the list of destinations exists at all (does not equal `null`), then, if it exists, we test if it has at least one element (its size is greater than 0), and if yes, we compare that element with the string `"London"`.