Truth and Consequences

The type “boolean”

Logic as Algebra

George Boole, 1815-1864
via Claude Shannon

Basic idea:
Treat “true” and “false” as values
Treat “and”, “or”, “not” as operations
true and x = x
true or x = true

The “boolean” primitive type

Like int, float, etc.
but just two values: true, false
» Note to C and C++ programmers: “true” and “false” are not synonyms for 1 and 0 in Java

boolean b;
b = true;

boolean values

Typically from comparisons:
int i=7; int j = 8;
boolean b = ( i == j ); // false

Combine with “and” (&&), “or” (||), “not” (!)
boolean c = (i < j);
b = b && (i > j) || c;
Algebraic properties of “boolean”

“or” is like addition (commutative, associative)
(x || y || z) == x || (y || z) == (x || y) || z

“and” is like multiplication (commutative, associative)

Multiplication distributes over addition:
(x && (y || z)) == (x && y) || (x && z)

Using boolean values

We use boolean (true, false) values to make decisions

```java
if (cost > limit) {
    System.out.println("Sorry, out of money.");
}
```

Common Comparisons

- `==` “equal” or same as (the same value)
  (more subtle than it looks: later we will distinguish between “the same thing” and “equal value”)

- `>` “greater than” (7 > 8, “foo” > “bar”)

- `<` “less than”

- `>=, <=` for ≤, ≥

- `!=` for ≠

Boolean Operators

<table>
<thead>
<tr>
<th>Java</th>
<th>logic, math</th>
<th>electronics</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp;&amp;</td>
<td>and, ∧, ⋅</td>
<td>AND</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>!</td>
<td>not, ¬</td>
<td>NOT</td>
</tr>
</tbody>
</table>
if ( ... ) { ... } else { ... }

int limit = 1000;
int balance = 5000;
int cost = 1001;
if (cost > limit || cost > balance) {
    System.out.println("Sorry, out of money.");
} else {
    balance = balance - cost;
}

else if ... (alternative form)

if (cost > limit) {
    System.out.println("Limit exceeded");
} else if (cost > balance) {
    System.out.println("Sorry, out of money");
} else {
    balance = balance - cost;
}

but watch out for ...

if (cost > limit)
    System.out.println("Charge limit exceeded");
else if (cost > balance)
    System.out.println("Limit exceeded");
else
    System.out.println("Ok");
    balance = balance - cost;
Summary: True or False

boolean type represents logical values
  just two values, true and false
comparisons create boolean values
  i < j, “Ben” < “Jerry”
create expressions with &&, ||, !
  like arithmetic with numbers (almost)
use in “if” and “if/else” statements
  control what the program does

Design and Build: Max

Given two numbers, x and y, print the larger of the two values

Given three numbers x, y, z, print the largest of the three values

Design and Build: Time Overlap

Tools like Doodle and WhenIsGood check for overlap between available times

Given two time ranges hh:mm to hh:mm, print the overlapping range hh:mm to hh:mm or print “No overlapping time”