More with Boolean

Sep 20

if statements, scope
“We should forget about small efficiencies, say about 97% of the time: premature optimization is the root of all evil.”

— Donald Knuth
Combining booleans
and and or and not

Boolean "Truth Table"

<table>
<thead>
<tr>
<th></th>
<th>and</th>
<th>or</th>
<th>not</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRUE</td>
<td>t</td>
<td>t</td>
<td>f</td>
</tr>
<tr>
<td>FALSE</td>
<td>f</td>
<td>f</td>
<td>t</td>
</tr>
</tbody>
</table>
The good integer problem

An integer is "good" if it is evenly divisible by 3 or evenly divisible by 5. However it is not good if it is evenly divisible by both 3 and 5. But, the integer is always good if it is evenly divisible by 7. Just print "true" or "false"

Algorithm:
\[ x \leq \text{integer to be tested for goodness} \]
Conditionals

doing different things on different inputs

• Consider abbreviated AnalyzeNumber for last class
• It kind of sucks
• Better:
  • "5 is positive" when greater than 0
  • "-5 is negative" when less than 0

```java
public class AnalyzeNumber5 {
    public static void main(String[] args) {
        int num = Integer.parseInt(args[0]);
        boolean isPositive = num > 0;
        System.out.println(num + " is positive " + isPositive);
    }
}
```
if

• if (boolean) { do something }
• else { do something else }

```java
public class AnalyzeNumber5b {
    public static void main(String[] args) {
        int num = Integer.parseInt(args[0]);
        boolean isPositive = num > 0;
        if (isPositive) {
            System.out.println(num + " is positive");
        } else {
            System.out.println(num + " is negative");
        }
    }
}
```
Max of 3 numbers

• Given 3 numbers (from the command line) print the largest
Chalkboards

• Redo the Max of 3 program but make it handle case when numbers may be equal
{} and "Scope"

• In Java, can declare a new variable at any time
  • its lifetime -- scope -- is from declaration until the next matching { }

• Prior to "if" all variables ended at end of program
  • Names can sometimes be re-used inside {} (usually not)
    • BUT this is confusing to people (not computers)
      • so do not
public class Scoper {
    public static void main(String[] args) {
        int a = 4;
        int b = 1;
        if (b > 0) {
            int c = 8;
            a = c;
        }
        System.out.println(a);
        int x = 5;
        if (x > 0) {
            int y = 7;
            System.out.println(x + y);
        }
        System.out.println(y);
        int m = -2;
        int n = 4;
        if (n > 0) {
            int m = 2;
            m = 4;
        }
        System.out.println(m + n);
    }
}
if

Variations

• Lots of legal ways to write if statements
• VSC does not like some
• Others are a bad idea
• Still others are incorrect

```java
public class if3 {
    public static void main(String[] args) {
        int val = Integer.parseInt(args[0]);
        boolean isPositive = val > 0;
        if (val > 0) {
            System.out.println(val + " is positive");
        }
        if (val > 0) { System.out.println(val + " is positive"); }
        if (val>0) System.out.println(val + " is positive");
        if isPositive {
            System.out.println(val + " is positive");
        }
    }
}
```