Lecture 2
Review

Given an int number, e.g.:
int number = 10;

Write code that, if the number is a multiple of 5, it prints HiFive, and if the number is divisible by 2, it prints HiEven.
## Logical Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>not</td>
<td>logical negation</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>and</td>
<td>logical conjunction</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Truth Table for **not** Operator: !

<table>
<thead>
<tr>
<th>p</th>
<th>!p</th>
<th>Example (assume age = 24, weight = 140)</th>
</tr>
</thead>
<tbody>
<tr>
<td>true</td>
<td>false</td>
<td>!(age &gt; 18) is false, because (age &gt; 18) is true.</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>!(weight == 150) is true, because (weight == 150) is false.</td>
</tr>
</tbody>
</table>
## Truth Table for `and` Operator: `&&`

<table>
<thead>
<tr>
<th>$p_1$</th>
<th>$p_2$</th>
<th>$p_1 &amp;&amp; p_2$</th>
<th>Example (assume age = 24, weight = 140)</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
<td>(age &lt;= 18) &amp;&amp; (weight &lt; 140) is false, because both conditions are both false.</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
<td>(age &lt;= 18) &amp;&amp; (weight == 140) is false, because one condition is false</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
<td>(age &gt; 18) &amp;&amp; (weight &gt; 140) is false, because (weight &gt; 140) is false.</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
<td>(age &gt; 18) &amp;&amp; (weight &gt;= 140) is true, because both (age &gt; 18) and (weight &gt;= 140) are true.</td>
</tr>
</tbody>
</table>
### Truth Table for **or** Operator:  ||

| $p_1$ | $p_2$ | $p_1 \text{ || } p_2$ | Example (assume age = 24, weight = 140) |
|-------|-------|-----------------------|----------------------------------------|
| false | false | false                 | (age > 34) || (weight >= 150) is false, because both evaluate to false |
| false | true  | true                  | (age > 34) || (weight <= 140) is true, because (age > 34) is false, but (weight <= 140) is true. |
| true  | false | true                  | (age > 14) || (weight >= 150) is false, because (age > 14) is true. |
| true  | true  | true                  | (age > 14) || (weight <=150) is true, because both conditions evaluate to true |
Practice: Determining Leap Year?

This program first prompts the user to enter a year as an `int` value and checks if it is a leap year.

A year is a leap year if it is divisible by 4 but not by 100, or it is divisible by 400.

You try: How would you write this conditional?
Problem: Determining Leap Year?

This program first prompts the user to enter a year as an int value and checks if it is a leap year.

A year is a leap year if it is divisible by 4 but not by 100, or it is divisible by 400.

\[(\text{year} \mod 4 == 0 \land \text{year} \mod 100 \neq 0) \lor (\text{year} \mod 400 == 0)\]
**Problem: Computing Taxes**

The US federal personal income tax is calculated based on the filing status and taxable income. There are four filing statuses: single filers, married filing jointly, married filing separately, and head of household. The tax rates for 2009 are shown below.

<table>
<thead>
<tr>
<th>Marginal Tax Rate</th>
<th>Single</th>
<th>Married Filing Jointly or Qualifying Widow(er)</th>
<th>Married Filing Separately</th>
<th>Head of Household</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>$0 – $8,350</td>
<td>$0 – $16,700</td>
<td>$0 – $8,350</td>
<td>$0 – $11,950</td>
</tr>
<tr>
<td>15%</td>
<td>$8,351 – $33,950</td>
<td>$16,701 – $67,900</td>
<td>$8,351 – $33,950</td>
<td>$11,951 – $45,500</td>
</tr>
<tr>
<td>35%</td>
<td>$372,951+</td>
<td>$372,951+</td>
<td>$186,476+</td>
<td>$372,951+</td>
</tr>
</tbody>
</table>
Problem: Computing Taxes, cont.

```java
if (status == 0) {
    // Compute tax for single filers
}
else if (status == 1) {
    // Compute tax for married file jointly
    // or qualifying widow(er)
}
else if (status == 2) {
    // Compute tax for married file separately
}
else if (status == 3) {
    // Compute tax for head of household
}
else {
    // Display wrong status
}
```

(Liang, Introduction to Java Programming, Tenth Edition, (c) 2015 Pearson Education, Inc. All rights reserved.)
switch Statements

switch (status) {
    case 0:  compute taxes for single filers;
        break;
    case 1:  compute taxes for married file jointly;
        break;
    case 2:  compute taxes for married file separately;
        break;
    case 3:  compute taxes for head of household;
        break;
    default: System.out.println("Errors: invalid status");
        System.exit(1);
}
switch Statement Flow Chart

- status is 0
  - Compute tax for single filers
  - break

- status is 1
  - Compute tax for married jointly or qualifying widow(er)
  - break

- status is 2
  - Compute tax for married filing separately
  - break

- status is 3
  - Compute tax for head of household
  - break

- default
  - Default actions
Trace switch statement

Suppose day is 2:

```java
switch (day) {
    case 1:
    case 2:
    case 3:
    case 4:
    case 5: System.out.println("Weekday"); break;
    case 0:
    case 6: System.out.println("Weekend");
}
```
Trace switch statement

```java
switch (day) {
    case 1:
    case 2:
    case 3:
    case 4:
    case 5: System.out.println("Weekday"); break;
    case 0:
    case 6: System.out.println("Weekend");
}
```
Trace switch statement

```java
switch (day) {
    case 1:
    case 2:
    case 3:
    case 4:
    case 5: System.out.println("Weekday"); break;
    case 0:
    case 6: System.out.println("Weekend");
}
```

Fall through case 3
Trace switch statement

switch (day) {
    case 1: 
    case 2: 
    case 3: 
    case 4: 
    case 5: System.out.println("Weekday"); break;
    case 0: 
    case 6: System.out.println("Weekend"); 
}

Fall through case 4
Trace switch statement

```java
switch (day) {
    case 1:
    case 2:
    case 3:
    case 4:
    case 5: System.out.println("Weekday"); break;
    case 0:
    case 6: System.out.println("Weekend");
}
```

**Fall through case 5**
Trace switch statement

switch (day) {
    case 1:
    case 2:
    case 3:
    case 4:
    case 5: System.out.println("Weekday"); break;
    case 0:
    case 6: System.out.println("Weekend");
}
Trace switch statement

```java
switch (day) {
    case 1:
    case 2:
    case 3:
    case 4:
    case 5: System.out.println("Weekday"); break;
    case 0:
    case 6: System.out.println("Weekend");
}
```
Your Turn!

- Write a program that will allow a user to choose the size of drink that they want and will output the price of the drink using a switch statement:

  Small: $1.00
  Medium: $1.50
  Large: $2.00
  Big gulp: $3.00