Announcements

• TA Office Hours
  – Sanjit Jhala
    • Tuesday 4 – 5, BE 352
  – Karl Brandt
    • Wednesday 9:30 – 10:30, BE 352

• SlugLUG meeting
  – Tonight: 8:00 PM by the whiteboards
  – Topic: Basic Intro to Unix commands

Chapter 3

• Statements and Control Flow
  – Up to now, all of our programs have executed from top to bottom without making any decisions
  – These programs do exactly the same thing every time they are run.
  – What if we want a program to do different things for different input?

Conditional Behavior

• We want our program to behave differently based on the value of its input.
• We see conditionals in real life:
  – If it is raining, wear your raincoat
  – If the waves are good, go surfing
  – If you are thirsty, drink some water
  – "If my girlfriend calls, tell her I'm not here."
Conditional in Java

• In Java, conditional behavior is expressed using the if statement
  – The if statement tells the computer to do some action only if some specified condition is satisfied
  – if (inputValue < 0)
    System.out.println("Bad input");

But First…

• Before we can go into more detail, we need to learn about some other kinds of statements:
  – Declaration statements
  – Expression statements
  – Block statements

Declaration Statements

• We've already seen declaration statements in our programs
  – double circumference;
  – String word1 = "Hello, ";
  – int price, change, dimes, pennies;
• A declaration statement has a type, followed by
  – a single variable name, or
  – a comma-separated list of variable names.
• Declaration statements are terminated with a ;
• The variables may be initialized in the declaration
Declaration Statements

- Error on page 56 of book (third line):
  - It says, "Declaration statements start with a type and are followed by a comma separated by a list of variables."  WRONG
  - It should say, "Declaration statements start with a type and are followed by a comma-separated list of variables."  CORRECT

Expression Statements

- Expression statements
  - An expression followed by a ;
  - We've seen two types so far:
    - Assignment expression
      - area = width * height;
    - Method call expression
      - System.out.println("Hello");
  - Not all expressions are valid in expression statements (more on this later)

Block Statements

- Block Statement: one or more statements inside braces, e.g.,
  
  ```
  int a = 4;          // Statement
  System.out.println(a); // Statement
  ```
  // Statement

- A Block Statement is a Statement
- This example contains 3 statements.
Block Statements

- Block Statements can contain Block Statements
- Variables declared within a block disappear when the block has finished executing

```java
{x = 1;
int y = 2;
System.out.print(x + y);
}
x += 3;  // y is not known here
```

Empty Statement

- Notice that the block statement does not end with a `;
- `;` by itself is an empty statement
  - An empty statement does nothing
  ```java
  x = x + 1;
  
  
  ```
- There are three statements in this example

Empty Statement

- Why would you want an empty statement?
- We’ll see when we discuss loops
Boolean Expression

- A boolean expression is any expression that evaluates to true or false
- Boolean Expressions include
  - true
  - false
  - Comparison
  - Result of logical operation

Relational Operators

<table>
<thead>
<tr>
<th>Name</th>
<th>Symbol</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Equal</td>
<td>==</td>
<td>a == b</td>
</tr>
<tr>
<td>Not equal</td>
<td>!=</td>
<td>a != b</td>
</tr>
<tr>
<td>Less than</td>
<td>&lt;</td>
<td>a &lt; b</td>
</tr>
<tr>
<td>Greater than</td>
<td>&gt;</td>
<td>a &gt; b</td>
</tr>
<tr>
<td>Less than or equal to</td>
<td>&lt;=</td>
<td>a &lt;= b</td>
</tr>
<tr>
<td>Greater than or equal to</td>
<td>&gt;=</td>
<td>a &gt;= b</td>
</tr>
</tbody>
</table>

Comparisons

- Comparisons (using relational operators) evaluate to true or false
  ```java
  int a = 5;
  int b = 7;
  boolean flag;
  flag = (a < b);
  System.out.println(flag);
  ```
Logical Operators

- Logical operators work with boolean operands
  - && AND a && b
  - || OR a || b
  - ! NOT !a

Logical Operations: Examples

- Example 1
  ```java
  int x, y;
  boolean b;
  x = Console.in.getInt();
  y = Console.in.getInt();
  b = (x == y);
  System.out.println(b);
  ```

- Example 2
  ```java
  boolean b = ((age >= 18) && (age < 65));
  System.out.println("Full Fare Adult: " + b);
  b = ((age < 18) || (age >= 65));
  System.out.println("Eligible for reduced fare: " + b);
  ```