JavaDoc
Conditionals / Boolean Expressions
Lecture 13

Announcements
- 1st Midterm Exam: TODAY, 6:30 pm
  - Locations:
    - Henning 200, 201, 202
    - Woodward 4, 5
  - Room allocation by last name
    - List of names vs rooms is online (WebCT and section web page)
  - Material: Chapters 1-4

Reading Assignments
- Reading for this week:
  - Edition 2: Ch. 6.1-6.4
  - Edition 3: Ch. 5.1-5.4

Objectives
- Learn how to create documentation with JavaDoc
- Get a basic understanding of the Java if statement
- Understand Java conditional expressions
- Learn the Java relational and Boolean operators

Recap: Scope – Variable Types
- Scope gives rise to different variable types
  - Local variables – variables defined within the body of a method
    - Exist while method is being executed
    - Can be accessed from within method body
    - Values are assigned inside method body
  - Formal parameters of method
    - Exist while method is being executed
    - Can be accessed from within method body
    - Values assigned externally by calling method

Scope – Variable Types
- Variable types (cont):
  - Fields – variables belonging to an object
    - Exist as long as the object they belong to
    - Can be accessed from any method in object’s class
      - (or by everybody if declared public)
    - Values assigned by constructors and mutator methods
      - (or by everybody if declared public)
Recap: Another Example Class

- BankAccount
  - Want to be able to
    - Create account with name, initial balance
    - Query the name of the account holder
    - Query the balance
    - Deposit a certain amount
    - Withdraw a certain amount
    - Transfer a certain amount from one account to another

Cleanup Pass

- Would we hand in our code as it stands?
  - good use of whitespace?
  - well commented?
    - every class, method, parameter, return value
  - clear, descriptive variable naming conventions?
  - constants vs. variables or magic numbers?
  - fields initialized?
  - good structure?
  - follows specification?
- Ideal: do as you go
  - commenting first is a great idea!
- Acceptable: clean up before declaring victory

Commenting Code

- Conventions
  - explain what classes and methods do
  - plus anywhere that you’ve done something non-obvious
    - usually better to say why than what
      - not useful
        - int wishes = 3; // set wishes to 3
      - useful
        - int wishes = 3; // follow fairy tale convention

javadoc Comments

- Specific format for method and class comments
  - running javadoc program will automatically generate HTML documentation
- Rules
  - /** to start, first sentence used for method summary
  - @param tag for parameter name and explanation
  - @return tag for return value explanation
  - other tags: @author, @version, etc.
  - */ to end
- Running
  - % javadoc Die.java
  - % javadoc *.java

javadoc Method Comment Example

```java
/**
 * Sets the die shape, thus the range of values it can roll.
 * @param numSides the number of sides of the die
 */
public void setSides(int numSides) {
    sides = numSides;
}

/**
 * Gets the number of sides of the die.
 * @return the number of sides of the die
 */
public int getSides() {
    return sides;
}
```

javadoc Class Comment Example

```java
/**
 * Die: simulate rolling a die
 * @author: CPSC 111, Section 206, Spring 05-06
 * @version: Jan 31, 2006
 * This is the final Die code. We started on Jan 24,
 * tested and improved in on Jan 26, and did a final
 * cleanup pass on Jan 31.
 */
```
Change of Gears:
Conditional Statements

- So far, execution goes line-by-line in order:
  ```
  int x = 3;
  int y = 5;
  int z = x + y;
  System.out.println("The answer is " + z);
  ```
- One of the powerful ideas that distinguishes a computer is the ability to make a decision:
- Decide to do something or not, depending on what's been computed.

Java's if statement

- Syntax:
  ```
  if ( condition ) statement;
  ```
- Examples:
  ```
  if (tax < 0) tax = 0;
  
  if (age < 19)
      System.out.println("Sorry, you may not buy alcohol.");
  ```

Example

```java
import java.util.Scanner;
public class IfTest {
    public static void main(String[] args) {
        System.out.println("How old are you?");
        Scanner scan = new Scanner(System.in);
        int age = scan.nextInt();
        if (age < 19)
            System.out.println("You are too young to buy alcohol.");
        System.out.println("Bye bye");
    }
}
```

Kurt’s Example

```java
import java.util.Scanner;
public class Feelgood {
    public static void main (String[] args) {
        int age;
        Scanner scan = new Scanner (System.in);
        System.out.println ("Enter your age: ");
        age = scan.nextInt();
        if (age < 20)
            System.out.println("Really, you look like you "+ "are " + (age + 5) + ".") ;
        System.out.println ("You don't look a day over " + (age - 10) + "!");
    }
}
```

Kurt’s Example

```java
import java.util.Scanner;
public class Feelgood {
    public static void main (String[] args) {
        int age;
        Scanner scan = new Scanner (System.in);
        System.out.println ("Enter your age: ");
        age = scan.nextInt();
        if (age < 20)
            System.out.println("Really, you look like you "+ "are " + (age + 5) + ".");
        System.out.println ("You don't look a day over " + (age - 10) + "!");
        else
            System.out.println("You don't look a day over " + (age - 10) + "!");
    }
}
```
Java's if statement with else

- Syntax:
  ```java
  if ( condition ) statement;
  else statement;
  ```

- Example:
  ```java
  if (age < 19)
      System.out.println("You may not buy alcohol.");
  else
      System.out.println("Buy beer!");
      System.out.println("This lecture sponsored by Molson");
  ```

Conditional Statements

- So far, execution goes line-by-line in order:
  ```java
  int x = 3;
  int y = 5;
  int z = x + y;
  System.out.println("The answer is " + z);
  ```

- One of the powerful ideas that distinguishes a computer is the ability to make a decision:
  - Decide to do something or not, depending on what's been computed.

Java's if statement

- Syntax:
  ```java
  if ( condition ) statement;
  ```

- Examples:
  ```java
  if (tax < 0) tax = 0;
  ```

Example

```java
import java.util.Scanner;
public class IfTest {
    public static void main(String[] args) {
        System.out.println("How old are you?");
        Scanner scan = new Scanner(System.in);
        int age = scan.nextInt();
        if (age < 19)
            System.out.println("You are too young to buy alcohol.");
        System.out.println("Bye bye");
    }
}
```
Java's if statement with else

- **Syntax:**
  ```java
  if ( condition ) statement;
  else statement;
  ```
- **Example:**
  ```java
  if (age < 19)
  System.out.println("You may not buy alcohol.");
  else
  System.out.println("Buy beer!");
  System.out.println("This lecture sponsored by Molson");
  ```

Conditional Statement

- **Boolean expression:** test that returns true or false
- **Conditional statement:** choose which statement will be executed next based on boolean expression

Boolean Expressions

- **Boolean expression:** test which returns either true or false when evaluated
  - aka conditional
- **Consists of operands and operators, like arithmetic expression**
  - but operators only return true or false when applied to operands
- **Two different kinds of operators**
  - relational
  - logical

Relational Operators

- **Compares two values (operands)**
- Operators (See Appendix F.)
  - `==` equal
    - returns true if they are equal, false otherwise
    - note: do not confuse this with `=`
  - `!=` not equal
    - returns true if they are not equal, false otherwise
  - `<` less than
  - `<=` less than or equal to
  - `>` greater than
  - `>=` greater than or equal to

Equality Example

```java
int a = 3;
int b = 5;
int c = 10;
if (a == b)
  System.out.println("these two values are equal");
if ((b - a) == a)
  System.out.println("that was a silly example");
if (a != b)
  System.out.println("nope!");
```
Logical Operators

- Way to combine results from relational operators into single test
- AND, OR, and NOT
  - in terms from math or philosophy class
- Operators
  - && logical AND
  - || logical OR
  - ! logical NOT

Logical AND

- Logical AND of values a and b evaluates to
  - true if both a and b are true
  - false otherwise
  - “If you are smart and beautiful, then I want to meet you”

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>a &amp;&amp; b</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>false</td>
<td>true</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
<td>false</td>
</tr>
<tr>
<td>true</td>
<td>true</td>
<td>true</td>
</tr>
</tbody>
</table>

Logical OR

- Logical OR of values a and b evaluates to
  - true if either a or b are true
  - true if both are true
  - false otherwise
  - “If you are mean or annoying, please go away.”

| a   | b   | a || b |
|-----|-----|-------|
| false | false | false |
| false | true  | true  |
| true  | false | true  |
| true  | true  | true  |

Logical NOT

- Logical NOT of value a evaluates to
  - true if a is false
  - false if a is true

<table>
<thead>
<tr>
<th>a</th>
<th>! a</th>
</tr>
</thead>
<tbody>
<tr>
<td>false</td>
<td>true</td>
</tr>
<tr>
<td>true</td>
<td>false</td>
</tr>
</tbody>
</table>

Logical Operator Examples

```java
int a = 3;
int b = 6;
int c = 10;

if ((b > a) && (c == 10))
    System.out.println("this should print");

if (!(b > a))
    System.out.println("this should not print");

if !(b > a)
    System.out.println("what happened?");
```
Logical Operator Examples

- is !(b > a) the same as
  - (a > b) NO!
  - (a >= b) YES!
  - (b < a) NO! (Same as (a > b))