Class Design, Conditionals, Loops: Putting it all Together

Lecture 18

Borrowing from slides by Alan Hu, Kurt Eiselt, Paul Carter, and Tamara Munzner

News
- Assignment 2 is out
  - Due next Friday, Oct 30
- Midterm 2 coming up the week after
  - Wednesday, Nov 4, 6:30-7:30
  - Note: the rooms will be different
- Material: conditionals, loops, arrays
  - Arrays are next up (Friday or Monday)

Reading Assignments
- Reading for this week: looping/iterations
  - Edition 3: Ch. 6.1-6.5
  - Edition 2: Ch. 7.1-7.5

Objectives for Today
- Improve understanding of nested loops
- Practice developing more complicated classes with loops and conditionals

Recap: Comparing Objects
- Versace gold lame evening gown with iridescent sequined bodice
- Versace gold lame evening gown with iridescent sequined bodice
- these values tested for equality with test of claimCheck1 == claimCheck2
- two different pointers (references), so false

Recap: Comparing Objects
- Versace gold lame evening gown with iridescent sequined bodice
- Versace gold lame evening gown with iridescent sequined bodice
- these values tested for equality with test of claimCheck1 == claimCheck2
- same pointers (references), so true
Recap: Comparing Objects: Moral

- Do NOT use == to compare objects for equality! (Unless you really mean to check for the same instance)
- Since Java doesn’t know what “same” means for objects, when you create a class, you must create methods to do this comparison when designing a class

Recap: Comparing Strings

- Strings in Java are weird. They are objects, but often act sort of like primitive types...
- Tamara’s old example:

```java
String name1 = "Bubba";
String name2 = "Bubba";
System.out.println(name1 == name2);  // prints false
```

- Surprise! Java internally tries to merge String objects together. So, sometimes == works on strings like on objects, and sometimes it works like on primitive types.
- Moral: Don’t use == on the String type!

Recap: Comparing Floating Point Numbers

- Is 0.3 the same thing as 1.0/10.0 + 1.0/10.0 + 1.0/10.0 ???
- No, floating point numbers are finite-precision approximations
- Better test for equality:
  - if( Math.fabs( f1 – f2 )< TOLERANCE ) …
  - Where TOLERANCE is something like 1e-10
  - Depends on application

Recap: Short-Circuiting Evaluation

- Raise your hand when you’ve computed the value of:

```
(3*2-6)*((4234+9834)*23432)/3.141592653589
```

- Java does something similar, but only for logical operators

Recap: Short-Circuiting Logical Operators

- This only makes a difference if the expression has side-effects, i.e. if it changes the values of variables
  - E.g: if( (b> a) && (C++ == 10) ) …
  - The value of c only gets incremented if (b>a) is true…
- This makes programs really hard to read!
  - Refrain from using side-effects in logical expressions…

Recap: Nested Loops

- Put a loop inside a loop
  - trace to see how it works

```java
public class NestedLoop {
    public static void main (String[] args) {
        for (int i = 1; i <= 3; i++) {
            for (int j = 1; j <= 3; j++) {
                System.out.print((i * j) + "  ");
            }
            System.out.println();
        }
    }
}
```
Recap: Modified Example…
- How would we only output the bottom left triangle of this matrix?

1  2  4
2  6  9

- How about other sub-triangles?
  Let's do a trace…

Another Example: Monte Hall Problem
- From old TV game show “Let’s Make a Deal”…
- There are three doors, labeled 1, 2, and 3.
- Behind one door is a new car.
- Behind the other two are goats.
- You pick a door.
- Monte Hall opens another door and shows you a goat. He asks you if you want to keep your current pick, or switch to the other unopened door.
- You win what’s behind the door you pick.
- Should you switch doors?

To Switch Or Not To Switch
- Does it make a difference?

- Try playing the game a million times, either switching each time, or not.

- Better yet, have the computer simulate playing a million times!

Monte Hall Game Tester
- What will this program look like?

Monte Hall Game Experiment
- What will this program look like?
- main method will loop 1,000,000 times
  - Each iteration, create a new game instance
  - Play the game.
    - Switch or not, depending on which test.
  - If we won, count it.
- Print the result.

Monte Hall Game Class Design
- What will this program look like?
- main method will loop 1,000,000 times
  - Each iteration, create a new game instance
  - Play the game.
    - Switch or not, depending on which test.
  - If we won, count it.
- Print the result.
Generalized Monte Hall

- What happens if we have 4 doors? 5 doors? n doors?
- Should we switch?