Lesson 47: Nested `for()` Loops (W14D1)

Balboa High School

Michael Ferraro

November 16, 2015

Do Now

1. Download class `PrimeFinder` from here, importing it into new Eclipse project `Lesson47`.
2. `PrimeFinder` provides the `isPrime()` method. Modify `main()` to use `isPrime()` to print a list of all primes from 2 to 101.
3. Generate a list of the first 50 prime numbers, starting from 2.

Aim

Students will learn how to structure and use nested `for()` loops.

First: Nested `if()` Statements

- You've already used these — `if()` within an `if()`

First: Nested `if()` Statements

- You've already used these — `if()` within an `if()`
- Example: `processRoll()` method from `CrapsGame.java`

```
if( point == 0 ) { //first roll
    if( total == 7 || total == 11 ) {
        //we win -- set point to zero, return 1
    } ...
}
```

First: Nested `if()` Statements

- You've already used these — `if()` within an `if()`
- Example: `processRoll()` method from `CrapsGame.java`

```
if( point == 0 ) { //first roll
    if( total == 7 || total == 11 ) {
        //we win -- set point to zero, return 1
    } ...
}
```

- Incidentally, `&&` may also be used:

```
if( point == 0 && ( total == 7 || total == 11 ) ) {
    //we win -- set point to zero, return 1
} ...
```

Nested `for()` Loops

- As you'd expect, these are `for()` loops within `for()` loops.
- **Predict** the output of this example:

```
public class NestedFor {

    public static void main(String[] args) {

        for ( int i = 1 ; i < 5 ; i++ ) {
            for ( int j = 1 ; j < 4 ; j++ ) {
                System.out.println(i + ", " + j);
            }
        }

    }
}
```

Nested for() Loops

Now create this class in project Lesson47 and see how it works.

```
public class NestedFor {

    public static void main(String[] args) {

        for ( int i = 1 ; i < 5 ; i++ ) {
            for ( int j = 1 ; j < 4 ; j++ ) {
                System.out.println(i + ", " + j);
            }
        }

    }
}
```

Processing

- Processing is a Java-like language that makes developing graphics quick and easy
- There are multiple web-based Processing developent tools
- We'll use OpenProcessing.org...

Processing: A Quick Demo

In a new "sketch":

```
size(400,600);                  //sets window size

ellipse(200,100,50,80);         //ellipse centered
                                //@ (200,100),
                                //50px wide, 80px tall
```

Processing: A Quick Demo

In a new "sketch":

```
size(400,600);

background(255);  //white
fill(255,0,0);    //red=255, green=0, blue=0

ellipse(200,100,50,80);
```

Nested `for()` Loops in Processing

1. See the Flash videos.
2. Open this prepared sketch.
3. Copy and paste code from that sketch into a new sketch.
4. Read the comments in that file carefully!
5. Modify the nested `for()` loops to make your Processing sketch work like the 3 videos.

Revisiting `PrintTriangle`

Think back to the `PrintTriangle` exercise.

```
*
**
***
****
*****
```

Complete this table:

row #	# stars
1	1
...	...
5	...

Revisiting `PrintTriangle`

- Create a class called `PrintTriangleFor` in `Lesson47`
- Those students looking for a challenge: Using nested `for()` loops, write a method that takes an integer argument, `n`, and constructs a triangle with a base of length `n`.
- Everyone else: Let's work together to solve that problem...

Revisiting PrintTriangle

Here's a starting point for PrintTriangleFor:

```
public class PrintTriangleFor {

    public static void main(String[] args) {
        //printTriangle(5);

    }

    public static void printTriangle(int n) {
        //nested for() loops here...
    }

}
```

Revisiting `PrintTriangle`

Pseudocode for `printTriangle()`:

```
for each row until row 'n' reached,
    print out 'row' stars
```

→ for row #1, print out 1 star

→ for row #2, print out 2 stars

→ for row #3, print out 3 stars

→ ...

→ for row `n`, print out `n` stars

Revisiting `PrintTriangle`

Pseudocode for `printTriangle()`:

```
for each row until row 'n' reached,
    print out 'row' stars
```

→ for row #1, print star #1

→ for row #2, print star #1, star #2

→ for row #3, print star #1, star #2, star #3

→ ...

→ for row `n`, print star #1, star #2, star #3, ..., star `n`

Revisiting `PrintTriangle`

- See if those hints are enough to get you started. I'll be around to help you if you get stuck!
- Once finished, move on to the next (HW) slide.

HW

- Finish §5 of PS #7 (see problems here)
- §6 introduction questions re: nested `for()` loops
- §6.1, required reading (Litvin §7.6)