Change to Syllabus

- I will be out-of-town on Friday April 10
- There was a midterm scheduled for Monday April 13

Changes:
- Class is cancelled on Friday April 10
- Midterm is moved to Tuesday April 14
- Review on Monday April 13
- Syllabus has been updated on my website

Review: Tiling with a for Loop

Again, we produce our row of tiles, using for
- Code may execute 0 or more times

for ( int col = 0; col < numSquares; col++ )
{
    int x = col * size;
    Rectangle rect = new Rectangle(x);
    if ( col % 2 == 0 )
    {
        rect.setBackground( Color.red );
    }
    window.add( rect );
}

Loop declaration – this time, the initialization of loop variable, condition, and progress all happen on the same line, at the start

Main work – code to be repeatedly executed

Nesting Loops

- We can produce our rows and columns of tiles, using nested for loops

for ( int row = 0; row < 8; row++ )
{
    for ( int col = 0; col < 8; col++ )
    {
        Rectangle rect = new Rectangle( col * recSize, row * recSize, recSize, recSize );
        // in even-numbered rows (0, 2, 4, ...)
        if ( row % 2 == 0 )
        {
            if ( col % 2 == 0 )
            {
                rect.setBackground( Color.red );
            }
            else // in odd-numbered rows (1, 3, 5, ...)
            {
                if ( col % 2 == 1 )
                {
                    rect.setBackground( Color.red );
                }
                window.add( rect );
            }
        }
    }
}

Nested Loop declaration – we loop over a second variable inside the original, going row by row, then column by column for each row

Main work – Here, we use both variables, in order to generate the full board
Understanding Nested Loops

- We can place **any type** of loop inside of another
- When we do so, we work from the **outside** to the **inside**

```java
for (int i = 0; i < 4; i++)
    for (int j = 0; j < 2; j++)
        System.out.println(i + " " + j);
```

<table>
<thead>
<tr>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
</tr>
<tr>
<td>0 1</td>
</tr>
<tr>
<td>1 0</td>
</tr>
<tr>
<td>1 1</td>
</tr>
<tr>
<td>2 0</td>
</tr>
<tr>
<td>2 1</td>
</tr>
<tr>
<td>3 0</td>
</tr>
<tr>
<td>3 1</td>
</tr>
</tbody>
</table>

- **outer loop** for `i`: runs 4 times
- **inner loop** for `j`: runs 2 times
- **8 total: 4 i-loops * 2 j-loops**

---

Exercise: Prime Numbers

- Recall that an integer is **prime** if it is divisible only by 1 and itself, otherwise it is **composite**
- Write code that reads an integer from the user and determines if it is prime.
  - You may use a **while** loop or a **for** loop
  - Develop an algorithm before you start writing Java code

---

This Week & Next:

- **Meetings this week:**
  - Monday/Tuesday/Friday: regular classroom
  - Wednesday: in the CS Lab (16 Wing)
- **Program 03:** due 11:59 PM, Tuesday 10 March
- **Reading Assignment 4:** due 12:00 PM, Monday 09 March
- **Office Hours:** Wing 212
  - Monday/Wednesday/Friday: 11:00 AM–12:00 PM
  - Tuesday: 3:15 PM–4:15 PM
- **Lab and Tutor Hours:** On my website