Readings
This Week: Ch 6 (Ch 7 in old 2nd ed).
(Reminder: Readings are absolutely vital for learning this stuff!)

Midterms – Save the Dates!
- Midterm #1 is 5:30-6:30pm on February 10 (Tuesday) in Woodward IRC 2
- Midterm #2 is 6-7pm on March 11 (Wednesday) in Woodward IRC 2

If you have an unavoidable conflict for the March 11 exam, you must email me (with your name, student ID, and description of the time conflict) by Wednesday, March 4!

Programming Assignment 2
- Assignment 2 will be up on WebCT tonight!
  - Click on the “Assignments” icon.
  - Due at NOON, March 10 (Tuesday), via electronic hand in.
  - Start early! (I really mean it!)
  - There is some Eclipse setup.

Learning Goals
By the end of the next several lectures you will be able to...
- Write programs that make decisions (“conditionals”, aka if statements) and repeat computations (“iteration”, while loops, for loops)
Learning Goals
By the end of class today you will be able to…
- Write simple \texttt{while} loops and trace their execution, \textit{with more confidence}!
- Write \texttt{for} loops and trace their execution.
- Start nesting loops.

Review: \texttt{while} Statement
\begin{verbatim}
while ( boolean expression )
body
\end{verbatim}
- Simplest form of loop in Java
- Body of loop can be
  - single statement
  - whole block of many statements in curly braces
- Meaning is kind of like English:
  “While you’re still hungry, have some more fries.”
  \begin{verbatim}
  while ( still hungry ) {
    eat fries;
  }
\end{verbatim}

Loan Amortization
- Suppose you have a $100,000 loan at 5\% interest, and you make an annual payment of $6000. How many years will it take you to pay the loan off?

Initially: You owe $100,000
After 1 year:
- Interest accrued: $100,000 * 0.05 = $5000
- Payment: $6000
- You owe $100,000 + $5000 - $6000 = $99,000

After 2 years:
- Interest accrued: $99,000 * 0.05 = $4950
- Payment: $6000
- You owe $100,000 + $4950 - $6000 = $97,950

Are you out of debt yet?

(Oh yeah, I’ve written a Loan class…)
Loan Amortization

- How about printing a table showing the balance after each year for the next 3 years? The next 25 years?

Java Shorthand: for Loops

- Loops like this are really common:

```java
int year = 0;
while (year < term) {
    myLoan.addInterest();
    myLoan.makePayment(annualPayment);
    System.out.println("Balance after …");
    year++;
}
```

Java Shorthand: for Loops

- Four main parts of a loop: 1. Initialize

```java
int year = 0;
while (year < term) {
    myLoan.addInterest();
    myLoan.makePayment(annualPayment);
    System.out.println("Balance after …");
    year++;
}
```

Java Shorthand: for Loops

- Four main parts of a loop: 2. Loop Test

```java
int year = 0;
while (year < term) {
    myLoan.addInterest();
    myLoan.makePayment(annualPayment);
    System.out.println("Balance after …");
    year++;
}
```

Java Shorthand: for Loops

- Four main parts of a loop: 3. Compute stuff in the loop.

```java
int year = 0;
while (year < term) {
    myLoan.addInterest();
    myLoan.makePayment(annualPayment);
    System.out.println("Balance after …");
    year++;
}
```

Java Shorthand: for Loops

- Four main parts of a loop: 4. Update loop variable.

```java
int year = 0;
while (year < term) {
    myLoan.addInterest();
    myLoan.makePayment(annualPayment);
    System.out.println("Balance after …");
    year++;
}
```
Java Shorthand: for Loops

- Java has a for loop statement to make this easy:

```java
int year = 0;
while (year < term) {
    myLoan.addInterest();
    myLoan.makePayment(annualPayment);
    System.out.println("Balance after …");
    year++;
}
```

Java Shorthand: for Loops

- Java has a for loop statement to make this easy:

```java
int year = 0;
for (year < term) {
    myLoan.addInterest();
    myLoan.makePayment(annualPayment);
    System.out.println("Balance after …");
    year++;
}
```

Java Shorthand: for Loops

- Java has a for loop statement to make this easy:

```java
for (int year = 0; year < term) {
    myLoan.addInterest();
    myLoan.makePayment(annualPayment);
    System.out.println("Balance after …");
    year++;
}
```

Java Shorthand: for Loops

- Java has a for loop statement to make this easy:

```java
for (int year = 0; year < term; year++) {
    myLoan.addInterest();
    myLoan.makePayment(annualPayment);
    System.out.println("Balance after …");
    year++;
}
```

Java Shorthand: for Loops

- 1. Initialize

```java
for (int year = 0; year < term; year++) {
    myLoan.addInterest();
    myLoan.makePayment(annualPayment);
    System.out.println("Balance after …");
}
```

Java Shorthand: for Loops

- 2. Loop Test

```java
for (int year = 0; year < term; year++) {
    myLoan.addInterest();
    myLoan.makePayment(annualPayment);
    System.out.println("Balance after …");
}
```
### Java Shorthand: for Loops

**3. Loop Body**

```java
for (int year = 0; year < term; year++) {
    myLoan.addInterest();
    myLoan.makePayment(annualPayment);
    System.out.println("Balance after ");
}
```

**4. Update Loop Variable**

```java
for (int year = 0; year < term; year++) {
    myLoan.addInterest();
    myLoan.makePayment(annualPayment);
    System.out.println("Balance after ");
}
```

---

### for Versus while Statement

- Anything that can be done with one type of loop can be done with another
- **for** statement convenient when
  - loop should be executed specific number of times
  - number can be determined before loop starts
- **while** statement convenient when
  - don't know yet how many times to execute loop body
  - but can check if it's time to end loop as you go

---

### Nested Loops

- **Very simple for loop**
  ```java
  public class SimpleLoop {
      public static void main (String[] args) {
          for (int i = 1; i <= 3; i++) {
              System.out.println(i);
          }
      }
  }
  ```
  - What does it do? Prints

- **What does it do?**

---
Nested Loops

- Very simple for loop

```java
class SimpleLoop {
    public static void main(String[] args) {
        for (int i = 1; i <= 3; i++) {
            System.out.println(i);
        }
    }
}
```

What if for every number below, want multiplication table of value times 2, x3, etc?

```
1 2 3
2 4 6
3 6 9
```

For every number printed by loop above
- want another loop to print numbers in a row, instead

How do we do that?

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

```java
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();
        }
    }
}
```
Nested Loops

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

```java
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();
        }
    }
}
```

Nested Loops

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

```java
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();
        }
    }
}
```
Nested Loops

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

```
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();
            }
        }
    }
}
```

1  1  2  3

Nested Loops

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

```
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();
            }
        }
    }
}
```

1  1  2  3

Nested Loops

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

```
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();
            }
        }
    }
}
```

1  1  2  3

Nested Loops

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

```
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();
            }
        }
    }
}
```

1  1  2  3

Nested Loops

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

```
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();
            }
        }
    }
}
```

1  1  2  3

Nested Loops

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

```
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();
            }
        }
    }
}
```

1  1  2  3

Nested Loops

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

```
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();
            }
        }
    }
}
```

1  1  2  3

Nested Loops

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

```
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();
            }
        }
    }
}
```

1  1  2  3

Nested Loops

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

```
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();
            }
        }
    }
}
```

1  1  2  3

Nested Loops

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

```
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();
            }
        }
    }
}
```

1  1  2  3

Nested Loops

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

```
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();
            }
        }
    }
}
```

1  1  2  3

Nested Loops

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

```
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();
            }
        }
    }
}
```

1  1  2  3

Nested Loops

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

```
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();
            }
        }
    }
}
```

1  1  2  3

Nested Loops

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

```
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();
            }
        }
    }
}
```

1  1  2  3

Nested Loops

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

```
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();
            }
        }
    }
}
```

1  1  2  3

Nested Loops

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

```
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();
            }
        }
    }
}
```

1  1  2  3

Nested Loops

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

```
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();
            }
        }
    }
}
```

1  1  2  3
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();
        }
    }
}
```

1  2  3
1  2  3
1  2  3
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();
        }
    }
}
```

1  2  3
2  3  6
3  6  9

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();
        }
    }
}
```

1  2  3
2  3  6
3  6  9
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();
            }
        }
    }
}
```

```
1 2
2 4 6
```
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();
        }
    }
}
```

1 2 3
1
2 3
2
3
3

```
```
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();
        }
    }
}
```

1 2 3
2 4 6
3 6 _

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();
        }
    }
}
```

1 2 3
2 4 6
3 6 9_

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();
        }
    }
}
```

1 2 3
2 4 6
3 6 9_

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();
        }
    }
}
```

1 2 3
2 4 6
3 6 9_

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();
        }
    }
}
```

1 2 3
2 4 6
3 6 9_

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();
        }
    }
}
```

1 2 3
2 4 6
3 6 9_
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();
        }
    }
}
```

Exit!

Questions?

Midterm #1

- Solutions are posted.
- Protocol for re-mark requests:
  - re-mark requests must be in writing (paper attached to exam) and submitted to instructor
  - re-marking is for mistakes, not for "I should have gotten more points for my wrong answer."
  - entire exam re-marked

Midterm #1

- Raw Score Stats:
  - Average: 57.1, Low: 11, High: 100 (Out of 100)
- Scaling Formula: 
  \[ \text{Scaled Score} = (\text{raw}/100)^{0.7} \times 100 \]

<table>
<thead>
<tr>
<th>Raw Score</th>
<th>Scaled Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>27</td>
</tr>
<tr>
<td>30</td>
<td>43</td>
</tr>
<tr>
<td>45</td>
<td>57</td>
</tr>
<tr>
<td>60</td>
<td>70</td>
</tr>
<tr>
<td>75</td>
<td>82</td>
</tr>
<tr>
<td>90</td>
<td>93</td>
</tr>
</tbody>
</table>