Lecture 10.2

Different Types of Loops, including *for* and *do*

### Counting loops

A loop that relies upon a counting variable to control the end of loop repetition is called a **counting loop**.

**General Form**

```java
initialize_counter
while (counter_satisfies_some_condition) {
    perform_work;
    update_counter_value;
}
```

**Example**

```java
public void makeDotRow( JFrame f ) {
    Oval circle;
    int xDelta, yDelta;
    int count;
    xDelta = w.getWidth() / 10;
    yDelta = w.getHeight() / 10;
    count = 0;
    while (count != 10) {
        dot = new Oval(count*xDelta+2, count*yDelta+2,5,5);
        dot.setBackground(Color.red);
        f.add( dot, 0 );
        count++;
    }
}
```

© 2006 Pearson Addison-Wesley. All rights reserved
Some Exercises

Exercise 1
Write a loop to draw a “ladder with twelve rungs. Each rung is 3 pixels thick and 20 pixels long. Rungs are spaced 14 pixels apart.

Exercise 2
Write a loop to draw ten vertical lines as follows.
• The first line is half the distance between the window sides.
• The second line is half the distance between the window’s left edge and the first line.
• The third line is half the distance between the window’s left edge and the second line.
  . . .
• The tenth line is half the distance between the window’s left edge and the ninth line.

for loop

Syntax

\[
\text{for (initStmt; condition; progressStmt) } \{ \\
\text{loopBody;}\\n\text{\} }
\]

where \text{loopBody} is a sequence of zero or more instructions, \text{initStmt} and \text{progressStmt} are instructions, and \text{condition} is a Boolean expression

Semantics (how to execute a for instruction)
1) execute \text{initStmt}.
2) Evaluate \text{condition}.
3.a) If \text{condition} is true, then execute \text{loopBody} followed by \text{progressStmt} and return to Step 2.
3.b) If \text{condition} is false, then terminate loop.
Comparison of while & for

Both while and for loops are well suited to counting loops

Example

```c
int count;
count = 1;
while (count != 10) {
    // do something
    count++;
}
```

```c
int count;
for (count=1; count != 10; count++) {
    // do something
}
```

A variable declared in initStmt is local to the loop’s body.

```c
for (int count=1; count != 10, count++) {
    // do something
}
```

Sentinel loops

Non-counting loops are sometimes called sentinel loops, because they use some other circumstance (the “sentinel”) to determine when to terminate loop repetition.

Counting Loop: Print the first fifteen numbers that are perfect squares.

Sentinel Loop: Print all perfect squares that are less than 1000.

Which of the following are sentinel loops?

Print the first ten letters of the alphabet.

Print the alphabet up to and including the letter “R”.

Approximate the value of 22/7 to the nearest eight digits.

Calculate the first integer power of 83 that is greater than one billion.

Starting with distance d1 and d2 repeatedly divide them both in half until the resulting lengths are within 1 inch of each other.
The Infinite Loop - Caution #1

An infinite loop (also called a dead loop) is one that will execute “forever”.

Example

```java
public void makeDotRow(JFrame f) {
    Oval dot;
    int xDelta, yDelta;
    int count;
    xDelta = w.getWidth() / 10;
    yDelta = w.getHeight() / 10;
    count = 11;
    while (count != 10) {
        circle = new Oval(count*xDelta+2, count*yDelta+2,5,5);
        circle.setBackground(Color.red);
        f.add(dot, 0);
        count++;
    }
}
```

A More Subtle Example

```java
int k;
k = 20;
while (k > 0) {
    k = Math.round(k / 2.0);
}
```

More About Infinite Loops

Potential Infinite Loop

```java
public int factorial(int n) {
    int result = 1;
    while (n != 0) {
        result = result * n
        n--
    }
    return result;
}
```

A Different Version

```java
public int factorial(int n) {
    int result = 1;
    while (n > 0) {
        result = result * n
        n--
    }
    return result;
}
```

Moral: Infinite loops are not always a bad thing.
Off-by-one Errors - Caution #2

It is common to write a counting loop that executes one time too many or one time too few.

Which of the following repeat the body 100 times?

1. \[
\begin{align*}
\text{int } k &= 1; \\
\text{while ( } k != 100 ) & \{ \\
& \quad \text{// do something} \\
& \quad k++; \\
& \}
\end{align*}
\]

2. \[
\begin{align*}
\text{int } k &= 0; \\
\text{while ( } k < 99 ) & \{ \\
& \quad \text{// do something} \\
& \quad k++; \\
& \}
\end{align*}
\]

3. \[
\begin{align*}
\text{int } k &= 1; \\
\text{while ( } k < 100 ) & \{ \\
& \quad \text{// do something} \\
& \quad k++; \\
& \}
\end{align*}
\]

4. \[
\begin{align*}
\text{int } k &= 100; \\
\text{while ( } k >= 0 ) & \{ \\
& \quad \text{// do something} \\
& \quad k--; \\
& \}
\end{align*}
\]

Loops Don’t Animate - Caution #3

The following is an attempt to cause \texttt{rect} to move. What goes wrong?

\[
\begin{align*}
\text{Rectangle } \texttt{rect}; \\
\texttt{rect} &= \text{new Rectangle}(10, 10, 10, 10); \\
\text{window.add( } \texttt{rect}, 0 ; \\
\text{while ( } \texttt{rect.getX()} < \text{window.getWidth()} ) & \{ \\
& \quad \texttt{rect.setLocation( } \texttt{rect.getX() + 5, rect.getY()} ); \\
& \quad \texttt{rect.repaint();} \\
& \}
\end{align*}
\]
do loop

Syntax

```
    do {
        loopBody;
    } while (condition)
```

where `loopBody` is a sequence of zero or more instructions and `condition` is a Boolean expression.

Semantics (how to execute a do instruction)

1) execute `loopBody`.
2) Evaluate `condition`.
3.a) If `condition` is true, then return to Step 1.
3.b) If `condition` is false, then terminate loop.

Nested Loops

A nested loop is one loop is within the body of another.

```
int a, b, c;
a = 1;
while (a != 101) {
    b = 1;
    while (b != 101) {
        c = 1;
        while (c != 101) {
            System.out.println(a + " + b + " + c);
            c++;
        }
        b++;
    }
    a++;
}
```
Some Exercises

Exercise 1
Write a segment of code to produce a sine values (see below)

<table>
<thead>
<tr>
<th>angle</th>
<th>0°</th>
<th>10°</th>
<th>20°</th>
<th>30°</th>
<th>40°</th>
<th>50°</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 deg</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 deg</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>90 deg</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Exercise 2
Write a segment of code to output roots (square cube, etc.)

<table>
<thead>
<tr>
<th>X</th>
<th>√X</th>
<th>√√X</th>
<th>√√√X</th>
<th>√√√√X</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

More Exercises

Exercise 3
Write a segment of code to produce a table of summations
(Note that $\sum \exp(k)$ denotes $\exp(1)+\exp(2)+\ldots+\exp(n)$)

<table>
<thead>
<tr>
<th>n</th>
<th>$\Sigma k$</th>
<th>$\Sigma k^2$</th>
<th>$\Sigma k^3$</th>
<th>$\Sigma k^4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>40</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Exercise 4
Write a loop to output all possible 4-letter words (using just capital letters).

Exercise 5
Repeat Exercise 4, except that no character may appear more than once in a word.