Repetition Statements

CSC 1051 – Data Structures and Algorithms I

Dr. Mary-Angela Papalaskari
Department of Computing Sciences
Villanova University

Course website:
www.csc.villanova.edu/~map/1051/

Some slides in this presentation are adapted from the slides accompanying Java Software Solutions by Lewis & Loftus
Repetition structures in Java

**while** loop:

```java
int count = 0;
while (count < 5)
{
    System.out.println (count);
    count++;
}
```

- Other repetition structures (Chapter 6 in text)
  - the **do** loop
  - the **for** loop
The **while** and **do** loops are similar.

```java
int count = 0;
while (count < 5)
{
   System.out.println (count);
   count++;
}
```

```java
int count = 0;
do
{
   System.out.println (count);
   count++;
} while (count < 5);
```
Similar – but not the same:

- **while Loop**
  - condition evaluated
  - true
  - false
  - statement

- **do Loop**
  - condition evaluated
  - true
  - false
  - statement

```java
int count = 0;
while (count < 5)
{
    System.out.println (count);
count++;
}

int count = 0;
do
{
    System.out.println (count);
count++;
} while (count < 5);
```

- The body of a do loop executes at least once
The **do** Statement in Java

- A **do** statement has the following syntax:

```java
do
{
    statement-list;
}
while (condition); //end do
```

- The **statement-list** is executed once initially, and then the **condition** is evaluated.

- The **statement-list** is executed repeatedly until the condition becomes **false**.
Example

This code inputs a number from the user:

```java
System.out.print("Enter a numeric grade (0 to 100):"hift); 
grade = scan.nextInt();
```

• Use a do loop to do **input verification**
Try this:

- Write a do loop to print the even numbers from 2 to 100.
Try this:

- Write a do loop to keep repeating a computation while a user has more data. (lab 4, problem 2B)
for: a loop with built in “counter”

- initialization
- condition evaluated
  - true
    - statement
    - increment
  - false
- initialization
**for**: a loop with built in “counter”

```
int count = 0;
while (count < 5) {
    System.out.println (count);
    count++;
}
```
for: a loop with built in “counter”

```
for (int count = 0; count < 5; count++)
    System.out.println (count);
```

Example

```java
int count = 0;
while (count < 5)
{
    System.out.println (count);
    count++;
}
```
The **for** Statement

- A **for** loop is functionally equivalent to the following **while** loop structure:

```plaintext
for ( initialization ; condition ; increment )
    statement;
```

```plaintext
initialization;
while ( condition )
{
    statement;
    increment;
}
```
The for Statement

- A *for statement* has the following syntax:

```plaintext
for ( initialization ; condition ; increment )
    statement;
```

The *initialization* is executed once before the loop begins.

The *statement* is executed until the *condition* becomes false.

The *increment* portion is executed at the end of each iteration.
The for Statement

- A *for statement* has the following syntax:

```
for (int count = 0; count < 5; count++)
    System.out.println (count);
```

The *initialization* is executed once before the loop begins.

The *statement* is executed until the *condition* becomes false.

The *increment* portion is executed at the end of each iteration.
The for Statement

• The increment section can perform any calculation:

```java
for (int num=100; num > 0; num -= 5)
    System.out.println (num);
```

• A for loop is well suited for executing statements a specific number of times that can be calculated or determined in advance
Try this:

- Write a for loop to print the even numbers from 2 to 100.
Character data reminder

Character data is stored as numeric codes. In Java, a char is stored as a 16-bit number – same as a short integer. We can convert back and forth between integers and characters using casts. Note that going between char and short is a narrowing conversion in both directions.

<table>
<thead>
<tr>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>byte</td>
<td>char</td>
</tr>
<tr>
<td>short</td>
<td>byte or char</td>
</tr>
<tr>
<td>char</td>
<td>byte or short</td>
</tr>
<tr>
<td>int</td>
<td>byte, short, or char</td>
</tr>
<tr>
<td>long</td>
<td>byte, short, char, or int</td>
</tr>
<tr>
<td>float</td>
<td>byte, short, char, int, or long</td>
</tr>
<tr>
<td>double</td>
<td>byte, short, char, int, long, or float</td>
</tr>
</tbody>
</table>

(Figure 2.6 page 84 in textbook – Java narrowing conversions)
Try this:

• Write a for loop to print out the ASCII codes corresponding to the characters ‘A’ through ‘Z’.

  *Hint:* Use a cast to convert from char to int
Try this:

- Write a **for** loop to print out the characters corresponding to the ASCII codes 0-255.

  **Hint:** Use a cast to convert from **int** to **char**
Try this:

Write a `for` loop to print $n$ stars, where $n$ is an integer parameter
Stars.java

public class Stars
{
    // Prints a triangle shape using asterisk (star) characters.
    public static void main (String[] args)
    {
        final int MAX_ROWS = 10;

        for (int row = 1; row <= MAX_ROWS; row++)
        {
            for (int star = 1; star <= row; star++)
            {
                System.out.print ("*");
                System.out.println();
            }
        }
    }
}
public class Stars {
    public static void main (String[] args) {
        final int MAX_ROWS = 10;

        for (int row = 1; row <= MAX_ROWS; row++) {
            for (int star = 1; star <= row; star++)
                System.out.print ("*");

            System.out.println();
        }
    }
}
The for Statement

• Each expression in the header of a for loop is optional

• If the initialization is left out, no initialization is performed

• If the condition is left out, it is always considered to be true, and therefore creates an infinite loop

• If the increment is left out, no increment operation is performed
Homework

• Review Sections 6.3, 6.4, and 6.5
  • **Always** do all self-review exercises when you review
• Exercises EX6.1 - 6.17 (important to get lots of practice with writing loops!)
• Look at some additional examples of programs from the text:
  • [ReverseNumber.java](#)
  • [Multiples.java](#)
  • [EvenOdd.java](#)

Some slides in this presentation are adapted from the slides accompanying Java Software Solutions by Lewis & Loftus

CSC 1051 M.A. Papalaskari, Villanova University