Loops revisited: do & for loops

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 **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**.
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:

```
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*
Try this:

- Write a do loop to print the even numbers from 2 to 100.
For some things the **do** loop is more appropriate:

```java
System.out.println("input a number >5");
int num = scan.nextInt();

while (num <= 5) {
    System.out.println("type a number >5");
    num = scan.nextInt();
}
```

```
do
{  
    System.out.println("type a number >5");
    num = scan.nextInt();
} while (num <= 5)
```
For some things the do loop is more appropriate:

```java
boolean more = true;
while (more) {
    System.out.print("Enter Quality Points ");
    qp = scan.nextInt();

    System.out.print ("Enter Credits ");
    credits = scan.nextInt();

    gpa = (double) qp /credits;
    System.out.println("GPA = " + gpa);
    System.out.print ("Again? 1=yes, 0=no ");
    more = (1 == scan.nextInt());
}
System.out.println("Thank you. Goodbye. ");
```

do {
    System.out.print("Enter Quality Points ");
    qp = scan.nextInt();

    System.out.print ("Enter Credits ");
    credits = scan.nextInt();

    gpa = (double) qp /credits;
    System.out.println("GPA = " + gpa);
    System.out.print ("Again? 1=yes, 0=no ");
} while (1 == scan.nextInt())
System.out.println("Thank you. Goodbye. ");
```
Another example: ReverseNumber.java

```java
import java.util.Scanner;

public class ReverseNumber {
    public static void main (String[] args) {
        int number, lastDigit, reverse = 0;
        Scanner scan = new Scanner (System.in);
        System.out.print("Enter a positive integer: ");
        number = scan.nextInt();
        do {
            lastDigit = number % 10;
            reverse = (reverse * 10) + lastDigit;
            number = number / 10;
        } while (number > 0);
        System.out.println("That number reversed is "+ reverse);
    }
}
```

Sample Run

Enter a positive integer: 2896
That number reversed is 6982
Another example: **EvenOdd.java**

```java
import javax.swing.JOptionPane;

public class EvenOdd {
    // Determines if the value input by the user is even or odd.
    // Uses multiple dialog boxes for user interaction.
    public static void main (String[] args) {
        String numStr, result;
        int num, again;
        do {
            numStr = JOptionPane.showInputDialog("Enter an integer: ");
            num = Integer.parseInt(numStr);
            result = "That number is " + ((num%2 == 0) ? "even" : "odd");
            JOptionPane.showMessageDialog(null, result);
            again = JOptionPane.showConfirmDialog(null, "Do Another?");
        } while (again == JOptionPane.YES_OPTION);
    }
}
```

The **JOptionPane** class provides methods that for creating dialog boxes.
import javax.swing.JOptionPane;

public class EvenOdd {
    public static void main (String[] args) {
        String numStr, result;
        int num, again;
        do {
            numStr = JOptionPane.showInputDialog(“Enter an integer: “);
            num = Integer.parseInt(numStr);
            result = “That number is “ + ((num%2 == 0) ? “even” : “odd”);
            JOptionPane.showMessageDialog(null, result);
            again = JOptionPane.showConfirmDialog(null, “Do Another?”);
        } while (again == JOptionPane.YES_OPTION);
    }
}
for: a loop with built in “counter”

- initialization
- condition evaluated
  - true: statement
  - false: increment

**for**: a loop with built in “counter”

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

---

**Example**

- **Initialization**: `int count = 0;`
- **Condition**: `while (count < 5)`
- **Statement**: `System.out.println(count);`
- **Increment**: `count++;`
**for**: a loop with built in "counter"

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

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

Example
The **for** Statement

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

```c
initialization;
while ( condition )
{
    statement;
    increment;
}
```

```c
for ( initialization ; condition ; increment )
    statement;
```
The for Statement

- A *for statement* has the following syntax:

```
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:

```java
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.
Example: *ReverseNumberAsString.java*

```java
import java.util.Scanner;

public class ReverseNumberAsString {
    public static void main (String[] args) {
        int number;
        String reverse = "";
        Scanner scan = new Scanner (System.in);

        System.out.print ("Enter a positive integer: ");
        number = scan.nextInt();
        String original = Integer.toString(number);

        for (int i=0; i<original.length(); i++)
            reverse = original.charAt(i) + reverse;

        System.out.println ("That number reversed is " + reverse);
    }
}
```

**Sample Run**

Enter a positive integer: 2896
That number reversed is 6982
Example: Stars.java

```java
//********************************************************************
// Stars.java       Author: Lewis/Loftus
//
// Demonstrates the use of nested for loops.
//********************************************************************

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();
        }
    }
}
```

Output

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

Exercise: can you make it print the row number in (1, 2, 3…) at the beginning of each line?
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