

# CMSC131

The FOR loop, Scope, and  
Nesting of loops.

# Iteration

There are times when you want to iterate until a certain condition or event occurs.

There are times when you know in advance exactly how many times you want to iterate.

In this second case, you are probably better off using a FOR loop than a WHILE loop...

# The FOR Loop

There will be three components within a FOR loop; initialization, a test condition, an update.

```
for (initialization-here; condition-here; update-here) {  
    body-here;  
}
```

NOTE: The term "update" is what our optional textbook uses. This is sometimes referred to as the counting expression or the increment. The term "update" is the most generic so technically the most accurate. We will almost always have the update either increment or decrement a value that is initialized and tested in the condition.

# A Simple For Loop

```
public class SimpleFor {  
    public static void main(String[] args) {  
        int sum = 0;  
  
        for (int i=1; i<=100; i=i+1) {  
            sum = sum + i;  
        }  
  
        System.out.println("The sum from 1 to 100 is "+sum);  
    }  
}
```

# Let's combine things...

```
import java.util.Scanner;

public class SumToN {
    public static void main(String[] args) {
        int sum = 0;
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter a number: ");
        int n = sc.nextInt();

        for (int i=1; i<=n; i=i+1) {
            sum = sum + i;
        }

        System.out.println("The sum from 1 to "+n+" is "+sum);
    }
}
```

# Counting down...

```
public class MajorTom {  
    public static void main(String[] args) {  
  
        for (int secsToLaunch=10; secsToLaunch>0; secsToLaunch=secsToLaunch-1) {  
            System.out.println(secsToLaunch);  
        }  
        System.out.println("Launch!");  
    }  
}
```

# Scoping Rules

Variables that are defined within a for loop exist only within the scope of that loop.

For example, the variable *i* in the previous example.

In general, variables and constants defined within a block are called local variables and their identifiers can only be used within that scope.

The memory associated with a variable may or may not be "free" after leaving a scope (we will see more about this soon).

# Out of Scope

When a variable goes "out of scope" its memory should be thought of as being "free" again and it is no longer valid to attempt to access it.

```
do {  
    int userResponse;  
    System.out.print("Enter an even number: ");  
    userResponse = sc.nextInt();  
} while ((userResponse%2) != 0);  
System.out.println("Thank you.");
```

# Nested Loops

You can have loops contained within the bodies of other loops.

- You will see this in a wide variety of contexts throughout your studies and career.

It is very common to nest FOR loops.

It is very common to create a new variable just within the scope for the FOR loop to act as the loop control variable.

# Rows and Columns (I)

```
public class NestedForLoops1 {
    public static void main(String[] args) {

        for (int row = 1; row <= 5; row = row + 1) {
            for (int col = 1; col <= 5; col = col + 1) {
                System.out.print(row + "," + col + " ");
            }
            System.out.println();
        }
    }
}
```

# Rows and Columns (II)

```
public class NestedForLoops1 {
    public static void main(String[] args) {

        for (int row = 1; row <= 5; row = row + 1) {
            for (int col = 1; col <= row; col = col + 1) {
                System.out.print(row + "," + col + " ");
            }
            System.out.println();
        }
    }
}
```

# WHILE rather than FOR

```
public class NestedWhileLoops {
    public static void main(String[] args) {

        int row = 1;
        while (row <= 5) {
            int col = 1;
            while (col <= row) {
                System.out.print(row + "," + col + " ");
                col = col + 1;
            }
            System.out.println();
            row = row + 1;
        }
    }
}
```

Copyright © 2012 : Evan Golub