ITERATION (REPETITION OF CODE, OR LOOPING)
Outline

- Loop Statements
- Types of Loops
  - for
  - for each
- Programming with Loops
for Loop

• **for loop**: another common type of loop
  • Execute an *initialization* statement
  • Evaluate a **boolean** expression
  • If **true**, do code block then increment
  • If **false**, done with loop

```c
for (init; expression; increment) {
  statement1;
  statement2;
  ...
}
```
for Loop Versions

```c
for (init; expression; increment) {
    statement1;
    statement2;
    ...
}
```

- **{} block version**
- **single line version**
- **buggy version**
for Loop Example

- Print out summations, $0 + 1 + 2 + ... + N$

```java
public class SummationFor {
    public static void main(String[] args) {
        int limit = Integer.parseInt(args[0]);
        long sum = 0;

        for (int i = 1; i <= limit; i++) {
            sum += i;
            System.out.println("sum 0..." + i + " = " + sum);
        }
    }
}
```
for Loop Anatomy

Declare and initialize a variable for use inside and outside the loop body

Condition which must be true to execute loop body

Changes the loop counter variable

Declare and initialize a loop control variable

Loop body, executes 0 or more times

```java
long sum = 0;
for (int i = 1; i <= limit; i++) {
    sum += i;
    System.out.println("sum 0..." + i + " = " + sum);
}
```
The `for` Statement

- A `for` statement executes the body of a loop a fixed number of times.
- Example
  ```java
  for (int count = 1; count < 3; count++)
      System.out.println(count);
  ```
The **for** Statement

- **Syntax**
  
  ```
  for (Initialization, Condition, Update)
  
  Body Statement
  ```

- **Body Statement** can be either a simple statement or a compound statement in `{}`.

- **Corresponding** `while` **statement**
  
  ```
  Initialization
  while (Condition)
  
  Body Statement Including Update
  ```
The **for** Statement

- The semantics of the **for** statement
The `for` Statement

• Possible to declare variables within a `for` statement

```c
int sum = 0;
for (int n = 1 ; n <= 10 ; n++)
    sum = sum + n * n;
```

• Note that variable `n` is local to the loop
The **for** Statement

- A comma separates multiple initializations
- Example
  ```c
  for (n = 1, product = 1; n <= 10; n++)
      product = product * n;
  ```
- Only one boolean expression is allowed, but it can consist of `&&`s, `||`s, and `!`s.
- Multiple update actions are allowed, too.
  ```c
  for (n = 1, product = 1; n <= 10;
      product = product * n, n++);
  ```
The for-each Statement

• Possible to step through values of an enumeration type
• Example

```java
enum Suit {CLUBS, DIAMONDS, HEARTS, SPADES}
for (Suit nextSuit : Suit.values())
    System.out.print(nextSuit + " ");
System.out.println();
```
Nested Loops

- A loop inside another loop

```java
public class StarTriangle {
    public static void main(String[] args) {
        int limit = Integer.parseInt(args[0]);
        for (int i = 0; i < limit; i++) {
            for (int j = 0; j <= i; j++)
                System.out.print("*");
            System.out.println();
        }
    }
}

% java StarTriangle 4
* 
**
***
****
```
Loop Choice

• Does your loop need a **counter variable**?
  • e.g. Going from 0 to N or N to 0 in fixed steps
  • Use a **for loop**
  • Counter variable is local to loop
  • Harder to forget the increment/decrement

• Do you need an **unknown number of loops**?
  • Use a **while loop**

• Do you need to **loop at least once**?
  • Use a **do while loop**
Tracing Variables

- *Tracing variables* means watching the variables change while the program is running.
  - Simply insert temporary output statements in your program to print the values of variables of interest.
  - Or, learn to use the debugging facility that may be provided by your system.
Loop Bugs

• Common loop bugs
  • Unintended infinite loops
  • Off-by-one errors
  • Testing equality of floating-point numbers

• Subtle infinite loops
  • The loop may terminate for some input values, but not for others.
  • For example, you can’t get out of debt when the monthly penalty exceeds the monthly payment.
Summary

• Loop Statements
• Types of Loops
  • for
  • for each
• Programming with Loops
Your Turn

• Write a for loop that generates 10 random numbers between 0.0 and 100.0 as test scores. Calculate the average of those scores and print the average out to the screen.

• Name your program RandomGrades.java and submit it to the Activity 03 dropbox on Moodle. 1 point for turning something in, 2 points for turning in something that is correct.