Day 5: Introduction to For Loops

Please help NCC conserve resources by not printing this document on campus.

A **for loop** is a code construct that allows you to tell the computer to perform a set of tasks many times in a row. For loops are controlled by a **loop control variable** (LCV). Syntax:

Listing 1: For Loop Syntax

```
for( initialization; condition; increment ) {
    /*
     * body of loop
     */
}
```

The **initialization** happens once, before the first **iteration** of the loop. The **condition** is checked before each iteration, and the body of the loop is only performed if the condition is true. The **increment** is performed after each iteration.

Here's a loop that creates and displays five randomly-generated integers:

Listing 2: Create Five Integers

```
int roll;

for( int i = 0; i < 5; i++ ) {
    roll = ( int )( Math.random( ) * 6 + 1 );
    System.out.println( "You rolled " + roll );
}
```

The LCV in this example is the variable `i`; this is a common choice. Notice that this variable is declared within the loop's *scope*, not the method's scope. It is a syntax error to refer to `i` after the bottom of the loop. If you need to do so, declare `i` before the loop begins.

What does the code in Listing 2 do? It is often helpful to use a table to help answer this question:

i	roll	Output

An **infinite loop** is one in which the condition never becomes false. This is a common bug.

Loops are often used to keep a *running total*. This example creates five random integers, and determines their sum:

Listing 3: Add Five Integers

```
int sum = 0;
int roll;
for( int i = 0; i < 5; i++ ) {
    roll = ( int )( Math.random( ) * 6 + 1 );
    sum = sum + roll;
}
System.out.println( "The sum is " + sum );
```