CECS 174 – Lecture 16 – For Loops

For Loop – Usually used when you know exactly how many times you want a loop to repeat, as opposed to a while loop where you want it to repeat until a specific condition is met. A for loop has the same components as a while loop, but in a more condensed form.

```
for ( initialize; condition; update)
{
     statements;
}
```

Any declarations made within the parenthesis or brackets will create a variable that is local that loop. The variable will not be accessible outside of the brackets.

Comparison – the following two loops will execute in the same manner. The only difference will be that in the while loop, the variable `count` will be available outside of the loop.

While Loop:

```
int count = 1;  //initialize

while ( count <= 5 ) //condition – ending value
{
     System.out.println("Count = " + count);
     count++; //update
}
```

For Loop:

```
for (int count = 1; count <= 5; count++)
{
     System.out.println("Count = " + count);
}

/* Output:

Count = 1
Count = 2
Count = 3
Count = 4
Count = 5
*/
```

If the variable needs to be accessed outside of the for loop, then it can be declared prior to the loop, and the initial value can be set in the initialization section of the for loop.

Calculations with For Loops – you can use a for loop to compute a solution.

Examples:

Calculating Interest:

```
double balance = 1000.00;
double rate = 0.05;
years = 5;

for ( int count = 1; count <= years; count++ )
{
     double interest = balance * rate;
     balance = balance + interest;
     System.out.printf("Year %d: $%.2f\n", count, balance);
}

/* Output:

Year 1: $1050.00
Year 2: $1102.50
Year 3: $1157.63
Year 4: $1215.51
Year 5: $1276.28
*/
```

Averaging User Input:

```
int sum = 0;
int amt;
int count;

for ( count = 1; count <= 5; count++ )
{
     amt = in.nextInt();
     sum = sum + amt;
}
System.out.printf("Avg=%.2f\n", (double)sum / (count-1));

/* Output:

1 2 3 4 5
Avg=3.00
*/
```

This for loop needs to declare count outside of the for loop because it has scope. Scope is the area where a variable can be seen. If count was declared inside the for loop, then the println statement would give an error because count would not be visible.

Nested For Loops – one for loop can be placed into another for a multiplicative effect.

Example:

```
for ( int ydir = 1; ydir <= 5; ydir++ )
{
     for ( int xdir = 1; xdir <= 5; xdir++)
     {
          System.out.print("*");
     }
     System.out.println();
}

/* Output:

*****
*****
*****
*****
*****
*/
```

Inner for-loops can test the outer for-loop's variable.

Example:

```
for ( int ydir = 1; ydir <= 5; ydir++)
{
     for (int xdir = 1; xdir <= ydir; xdir++)
     {
          System.out.print("*");
     }
     System.out.println();
}

/* Output:

*
**
***
****
*****
*/
```