Loop Concepts

A loop, or iteration structure, is a software construct that enables repetition of program code. The general logic is:

```
// set up a condition:
boolean again = true;
// test condition:
while (again) {
        // do statements while condition is true;
        // include code that might change condition:
        System.out.print("Repeat process? (y/n): ");
        String answer = kb.nextLine();  // assumes existence of kb
        if(answer.equalsIgnoreCase("n"))
                again = false;
}
```

Types of Loops:

- Event controlled: depends on condition external to program
    - unpredictable number of iterations
    - input (from user or file) determines when loop ends
- Count controlled: depends on value of internal counter
    - predictable number of iterations
    - use when you want process to repeat a certain number of times
    - often use shortcut operators:
        - ++
        - --
        - +=
        - -=
        - *=
        - /=
        - %=

Examples of event-controlled loops: everything we've seen thus far; also SumAvg

Examples of count-controlled loops: SumAvgCount, Circles

For loops: syntactic variation on while loops

- pretest
- almost always count-controlled
- all of loop "housekeeping" (including update) appears in heading
- Examples:

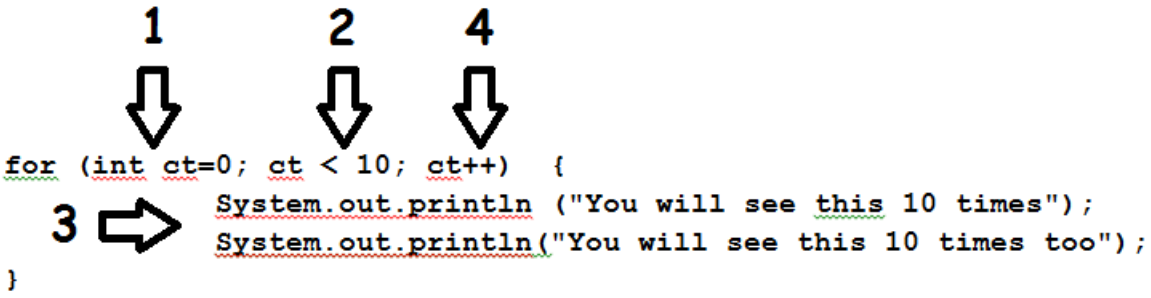
```
for (int ct=0; ct < 10; ct++)  {
            System.out.println ("You will see this 10 times");
            System.out.println("You will see this 10 times too");
}

for (int ct = 20; ct >= 0; ct -= 2)
            System.out.printf("ct=%d\n", ct);

for (int ct = 2; ct < 1024; ct *= 2)
            System.out.printf("%d\n", ct);

for (int ct=0; ct < 10; ct++)  {
```

- Notes:
  - Logic almost, but not quite the same as count-controlled while loop: in while loop, update can occur ***anywhere*** in the loop body, but ***always at end of iteration in for loop***
  - Declaration of count variable: happens only once, but is considered to be part of the for loop – so it doesn't exist outside the loop (reason for redeclaration)
  - Order of operations:



1. Initialization: performed exactly once
2. Check condition: performed before each iteration of loop body (and one more time, after the last update)
3. Execution of loop body: performed each time condition tests true
4. Update: performed after each iteration (before next condition check)

- Example: DataFileBuilderForNestLoop

Post-test loops: do/while

- Logically different from while & for: test occurs at end of each iteration, not before each iteration
- Means loop is guaranteed to go at least once, because condition value is unknown until after all instructions have been performed the first time
- Example: DoWhileLoops