Introduction to `for` and while Loop Structure in Java

The general `while` loop structure for Java adheres to the following form:

```
INITIAL-STATEMENT
while(CONTINUING-CONDITION)
      ITERATIVE-STATEMENT
```

The general `for` loop structure for Java adheres to the following form:

```
for (INITIAL-STATEMENT; CONTINUING-CONDITION; ITERATIVE-STATEMENT)
```

The `INITIAL-STATEMENT` section is evaluated once before the loop begins and is usually where a counter variable is declared and initialized. The `CONTINUING-CONDITION` section typically contains at least one `boolean` statement usually regarding the counter variable and is evaluated before each iteration begins. The `ITERATIVE-STATEMENT` section is evaluated at the end of each iteration and often contains a statement iterating the counter variable.

Though the `for` loop is merely a modified form of the `while` loop, by convention and ease they are used to achieve different purposes. A `for` loop is used when the exact number of iterations in known, while a `while` loop is used when the number of iterations is dependent upon satisfying some conditional. For instance, a program simulating a person eating until he is full would probably use a `while` loop, because the person must continue eating until the condition of being full is satisfied. However, if another person decided that she would only eat ten chicken nuggets, the program would use a `for` loop.

Case Study: `while` Loop

The following is an example of a `while` loop that will add the numbers 1 through 3 together and print the final result:

```
1      int i = 1, total = 0;
2      while(i < 4){
3        total = total + i;
4        i++;
5      }
6      System.out.println(total);
```

The `INITIAL-STATEMENT` in this case is just line 1 that declares and initializes the variables `i` and `total`. The `CONTINUING-CONDITION` is `i < 4` meaning that the loop will continue while the statement `i` is less than `4` evaluates to `true`. The `ITERATIVE-STATEMENT` is line 4 where the variable `i` is incremented by 1.

Thus the example can be colored like so:

```
1      int i = 1, total = 0;
2      while(i < 4){
3        total = total + i;
4        i++;
5      }
6      System.out.println(total);
```

Before the loop even begins, everything within the section INITIAL-STATEMENT (line 1) will be carried out so that there is a variable `i` initialized to 1 and a variable `total` initialized to 0. The CONTINUING-CONDITION is evaluated and must be evaluated to `true` for the loop to be carried out. Because `i` is equal to 1, the CONTINUING-CONDITION statement `i < 4` is `true` and the program enters the loop.

In the first iteration, the current `i` value, 1, is added to the previous `total` value, 0, to create the new `total` value, 1, in line 3. In line 4, ITERATIVE-STATEMENT is carried out and `i` becomes 2.

Because `i` is equal to 2, the CONTINUING-CONDITION statement `i < 4` is `true`, and thus the loop continues. The current `i` value is now 2 in the second iteration and is added to the previous `total` value, 1, to create the new `total` value, 3 in line 3. In line 4, ITERATIVE-STATEMENT is carried out and `i` becomes 3.

Because `i` is equal to 3, the CONTINUING-CONDITION statement `i < 4` is `true`, and thus the loop continues. In the third iteration, the current `i` value is now 3 and is added to the previous `total` value, 3, to create the new `total` value, 6 in line 3. In line 4, ITERATIVE-STATEMENT is carried out and `i` becomes 4.

Because `i` is equal to 4, the CONTINUING-CONDITION statement `i < 4` is `false`, and thus the loop ends. The result of 6 is printed.

The Conversion from MATLAB and Python to Java

The following are `while` loops in MATLAB, Python and Java that will go from 0 to 9 (for a total of 10 items) and print the current iteration.

MATLAB
```
i = 0;
while(i < 10)
   disp(i);
   i = i + 1;
end
```

Python
```
while i < 10:
    print i
    i=i+1
```

Java
```
int i = 0;
while(i < 10){
  System.out.println(i);
  i++;
}
```

The following are `for` loops in MATLAB, Python and Java that will go from 0 to 9 (for a total of 10 items) and print the current iteration.

MATLAB
```
for i = 0:1:9
   disp(i);
end
```

Python
```
for i in range(10):
   print i
```

Java
```
for (int i = 0; i < 10;
     i++){
   System.out.println(i);
}
```

Practice Questions:

1. How many times will the following loops iterate? Do any of them not iterate at all or loop infinitely? (To get the solution, run the code.)

a.
```
int i = -1, counter = 0;
while(i < 4){
   System.out.println(i);
   i++;
   counter++;
}
System.out.println("This loop iterated "+ counter + " times ");
```

b.
```
int i = 0, counter = 0;
while(i < -1){
   System.out.println(i);
   i++;
   counter++;
}
System.out.println("This loop iterated "+ counter + " times ");
```

c.
```
int counter = 0;
for (int i = 0; i <= 4; i++){
   System.out.println(i);
   counter++;
}
System.out.println("This loop iterated "+ counter + " times ");
```

d.
```
int counter = 0;
for (int i = 7; i > -1; i--){
   System.out.println(i);
   counter++;
}
System.out.println("This loop iterated "+ counter + " times ");
```

2. For the following situations indicate whether it would be more appropriate to use a for loop or a while loop:
   a. John has to run five laps around the park every day to stay in shape.
   b. Sheila has to apply aloe vera to her skin until the sunburn goes away.
   c. Doug will keep playing poker until he runs out of money.
   d. Hope will not stop studying until she has all of the formulae memorized.
   e. Sam like to drink coffee four times a day to stay sharp.