For Loops and Iteration

To introduce the ‘for’ loop, Let's reexamine our old friend, the 'while' loop:

```
int list [] = new int[5];
int i=0;
while (i<list.length)
{
   System.out.println(list[i]);
   i=i+1;
}
```

Within the code, there is a loop initialization, loop condition, and loop increment.

Java's 'for' loop places these three parts within a structured syntax:

```
for (int i=0;i<list.size();i++)
{
   System.out.println(list[i]);
}
```

Abstractly...

forStatement::= for (<init>;<condition>;<increment>) <statement>

When execution begins, the <init> statement is executed. It is only executed once. The condition is checked after the init statement and after each iteration. After each iteration, the <increment> statement is executed prior to re-checking the condition.

Iterators

Java provides special Iterator classes for iterating through collections like ArrayList. They provide an elegant way of looping. Assume ‘list’ is an ArrayList. Then:

```
ListIterator<String> iterator = list.listIterator();   // ask any list for an iterator.

while (listIterator.hasNext())
{
   String s = listIterator.next();
   // do something with the object s
}
```

In class assignment: Write a class TotalLoop with a main method that creates a built-in array of integers and then uses a for loop to compute the total of all elements in the list.