Return to Module Overview Page (https://onlinelearning.berkeley.edu/courses/665040/pages/module-six)

6.7: Iteration

A very common collections operation is iteration or looping through each element in a collection. Java provides mechanisms to make it easy.

All the collections have an Iterator object that the iterator( ) method returns. Iterator has two methods that enable iterating through a collection: the hasNext( ) method, which returns a boolean on whether there is an element available, and the next( ) method, which returns the current element in the iteration. The next( ) method throws a NoSuchElementException when it reaches the end of the iteration.

The following example shows the usage of an iterator. We will assume that employees is a collection of employee objects:

```
for (Iterator iter = employees.iterator(); iter.hasNext();) {
   Employee employee = (Employee)iter.next();
   System.out.println(employee.name);
}
```

Java 1.5 introduced the for/in loop. This is similar to the foreach operator in other languages. It does the same thing as the iterator using the following format:

```
for (type var: collection) { block }
```

Our example becomes more concise if rewritten using for/in:

```
for (Employee employee: employees) {
   System.out.println(employee.name);
}
```