Generics

generic

ge·ner·ic - [j*uh*-**ner**-ik]

1. of, applicable to, or referring to all the members of a genus, class, group, or kind; general.

(dictionary.com)

The interface Comparable in java.lang:

```
public interface Comparable {

   public int compareTo(Object o)

}
```

An illegal use of *compareTo* that cannot be detected until *run time*:

```
Comparable c =new Date();

System.out.println(c.compareTo("red"));
```

The interface Comparable in java.lang since Java 5.0:

```
public interface Comparable <T> {

   public int compareTo(<T> o)

}
```

An illegal use of *compareTo* that will be detected at *compile time*:

```
Comparable<Date> c =new Date();

System.out.println(c.compareTo("red"));
```

What is the advantage of using generics?

Recall the SSN list's node:

```
public class SSNListNode {
    SSN data;
    SSNListNode next;
    public SSNListNode(SSN mySSN) {
        data=mySSN;
        next=null;
    }
}
```

The only data the list can hold is type SSN

We can make it *general* but not *generic:*

```
public class ListNode {
    Object data;
    ListNode next;
    public ListNode(Object myObject) {
        data=myObject;
        next=null;
    }
}
```

But this opens us up to *run time* exceptions.

We can make the *ListNode* generic:

```
public class ListNode <E> {
   E data;
   ListNode next;
   public ListNode(E myData) {
      data=myData;
      next=null;
   }
   public ListNode() {
      data=null;
      next=null;
   }
}
```

We can make the *LinkedList* generic

```
public class LinkedList<E> {
   private ListNode first;
   private ListNode last;
   private int length;

   public LinkedList() {
      ListNode ln = new ListNode();
      first = ln;
      last = ln;
      length = 0;
   }
   public void append ( E myData){
      ListNode n = new ListNode(myData);
      last.next = n;
      last = n;
      length++;
   }
   ....
}
```

Using the LinkedList class to contain *Strings*

```
public class StringListMain {
   public static void main(String[] args) {
      String[] strings =
           {"bat","cat","fat","hat","mat","sat"};
      LinkedList<String> stringList =
                          new LinkedList<String>();

      for (int i=0; i<strings.length; i++) {
         stringList.append(strings[i]);
      }
      stringList.print();
   }
}
```

Using the LinkedList class to contain *SSNs*

```
public class SSNListMain {
  public static void main(String[] args) {
    String[]ssnStrings =
     {"123456789","234567890","345678901","456789012"};
      SSN[] ssnArray = new SSN[ssnStrings.length];

      for (int i=0;i<ssnStrings.length;i++){
         ssnArray[i]=new SSN(ssnStrings[i]);
      }
      LinkedList<SSN> SSNList = new LinkedList<SSN>();
      for (int i=0; i<ssnArray.length; i++) {
         SSNList.append(ssnArray[i]);
      }
      SSNList.print();
  }
}
```