Java – Using Vectors and Hashtables

Vector Example

```
import java.util.Enumeration;
import java.util.Vector;


// Store Strings in a vector
Vector<String> v = new Vector<String>();
v.add("Football");
v.add("Basketball");
v.add("Tennis");


// Get first and second elements in the vector
String firstSport = v.firstElement();
String secondSport = v.get(1);


// Print all contents of the vector
Enumeration<String> sports = v.elements();
while (sports.hasMoreElements()) {
   String sport = sports.nextElement();
   System.out.println(sport);
}


// Remove Tennis
v.remove(2);


// Remove all
v.removeAllElements();
```

Hashtable Example

```
import java.util.Enumeration;
import java.util.Hashtable;


// Store Strings for the key & Strings for the values
Hashtable<String,String> ht =
   new Hashtable<String,String>();
ht.put("123", "Bob");
ht.put("456", "Jane");
ht.put("789", "Lulu");


// Look up a key in the hashtable
String key = "123";
String name = ht.get(key);
if (name == null)
   System.out.println(key + " not found.");
else
   System.out.println(name);


// Print all contents of the hashtable
Enumeration<String> keys = ht.keys();
while (keys.hasMoreElements()) {
   key = keys.nextElement();
   name = ht.get(key);
   System.out.println(key + " : " + name);
}


// Remove Jane
ht.remove("456");

// Remove everyone
ht.clear();
```

Vectors are ideal for storing lists of items where you typically do not need to search through the list for a specific item. **Hashtables** are ideal for storing key-value pairs. Access to any key-value is very fast when searching by key unlike Vectors which require a search through the entire Vector.

Hashtable of Vectors Example

```
// Store Strings for the key and Vectors of
// Strings for the values
Hashtable<String,Vector<String>> ht =
   new Hashtable<String,Vector<String>>();

Vector<String> v = new Vector<String>();
v.add("Denver");
v.add("Colorado Springs");
v.add("Fort Collins");
ht.put("CO", v);

v = new Vector<String>();
v.add("Little Rock");
v.add("Searcy");
v.add("Hot Springs");
ht.put("AR", v);

v = new Vector<String>();
v.add("Richmond");
v.add("Norfolk");
v.add("Virginia Beach");
ht.put("VA", v);
```

```
// Look up a key in the hashtable
String key = "CO";
Vector<String> cities = ht.get(key);
if (cities == null)
   System.out.println(key + " not found.");
else
   System.out.println(cities);


// Print all contents of the hashtable
Enumeration<String> keys = ht.keys();
while (keys.hasMoreElements()) {
   key = keys.nextElement();
   cities = ht.get(key);
   System.out.println(key + " : " + cities);
}

// Remove AR entry
ht.remove("AR");

// Remove all
ht.clear();
```