[0944] Dictionaries

Dictionaries are also very popular C# collections. Dictionaries store key-value pairs. Think of a "key" as the index for an item in a collection. Think of "value" as the actual item that is stored.

For example, you can consider an array as very simple key-value storage collection where the array index (the integer "i") is the key and the data item that is stored in the array ("foo[i]") is the value. Dictionaries expand on that concept by allowing anything to be the key - not just ints - and by using a fast, flexible hashtable under the covers to store the data. Note that keys must be unique.

Here's an example of a Dictionary in use:

```
using System;
using System.Collections.Generic;

namespace LWTech.Testing
{
   public class DictionaryDemo
   {
      public static void Main()
      {
         var d = new Dictionary<string, string>();

         d.Add("Father", "Homer");
         d.Add("Mother", "Marge");
         d.Add("Son", "Bart");

         Console.WriteLine(d["Father"]);
         Console.WriteLine(d["Mother"]);
         Console.WriteLine(d["Son"]);

      }
   }
}
```

Output:

```
Homer
Marge
Bart
```

Again, remember that Keys in a dictionary must be unique. Thus, the follow program will throw an exception when it is run:

```
using System;
using System.Collections.Generic;

namespace LWTech.Testing
{
```

```
  public class DictionaryDemo
  {
    public static void Main()
    {
      var d = new Dictionary<string, string>();

      d.Add("Father", "Homer");
      d.Add("Mother", "Marge");
      d.Add("Son", "Bart");
      d.Add("Daughter", "Lisa");
      d.Add("Daughter", "Maggie");

      Console.WriteLine(d["Father"]);
      Console.WriteLine(d["Mother"]);
      Console.WriteLine(d["Son"]);
      Console.WriteLine(d["Daughter"]);  // ???

    }
  }
}
```

Dictionaries are also great for counting items in a list. In that case, the items being counted would be the keys in the dictionary and each key's value would be an integer representing the number of times the item appears in the list. Here's a example:

```
using System;
using System.Collections.Generic;

namespace LWTech.Testing
{
  public class DictionaryDemo
  {
    public static void Main()
    {

      var randomStrings = new List<string>();
      var counts = new Dictionary<string, int>();

      // Create 1000 strings consisting of "rand" plus a random digit.
      for (int i = 0; i < 1000; i++)
        randomStrings.Add("rand" + new Random().Next(10));

      // Count up the frequence of each string in the list
      foreach (string key in randomStrings)
      {
        if (counts.ContainsKey(key))
          counts[key]++;
        else
          counts.Add(key, 1);
      }

      // Display the results
      foreach (string key in counts.Keys)
        Console.WriteLine(key + ": " + counts[key]);
```

```
    }
  }
}
```

Sample Output:

```
rand1: 100
rand8: 93
rand9: 86
rand6: 126
rand3: 91
rand2: 115
rand0: 82
rand4: 94
rand7: 114
rand5: 99
```