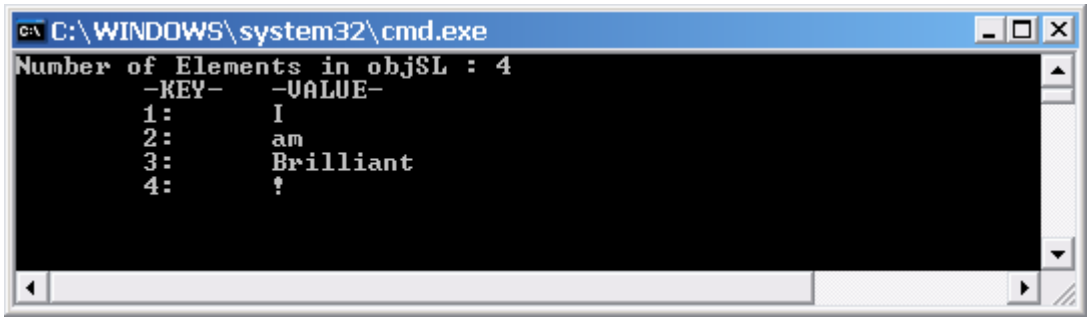
Objectives:

- **Collections**

1. Sample of SortedList

```
using System;
using System.Collections;
public class SampleSortedList
{
    public static void Main()
    {
        SortedList objSL = new SortedList();
        objSL.Add("4", "!");
        objSL.Add("3", "Brilliant");
        objSL.Add("2", "am");
        objSL.Add("1", "I");

        Console.WriteLine("Number of Elements in objSL : {0}",
objSL.Count);
        Console.WriteLine("\t-KEY-\t-VALUE-");
        for (int i = 0; i < objSL.Count; i++)
        {
            Console.WriteLine("\t{0}:\t{1}", objSL.GetKey(i),
objSL.GetByIndex(i));
        }
        Console.ReadLine();
    }
}
```

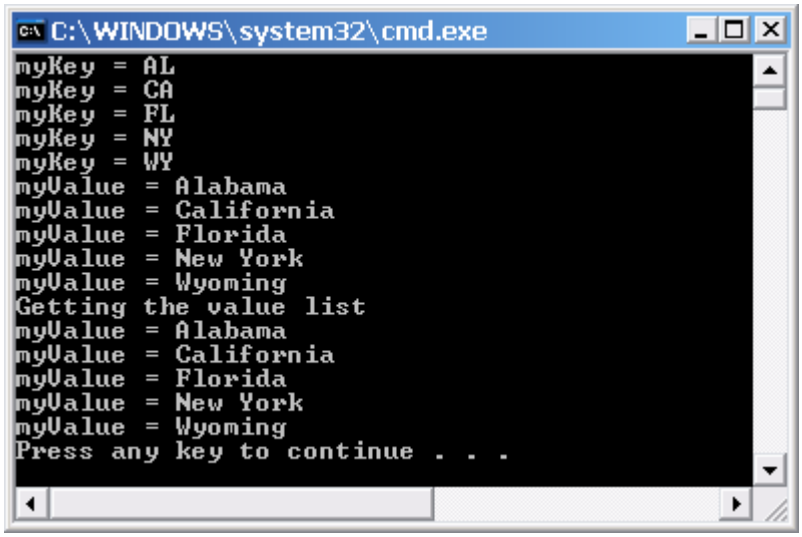
The output of the program:



2. SortedList: Get the value list using the GetValueList() method

```
using System;
using System.Collections;

class MainClass
{

    public static void Main()
    {
        SortedList mySortedList = new SortedList();

        mySortedList.Add("NY", "New York");
        mySortedList.Add("FL", "Florida");
        mySortedList.Add("AL", "Alabama");
        mySortedList.Add("WY", "Wyoming");
        mySortedList.Add("CA", "California");

        foreach (string myKey in mySortedList.Keys)
        {
            Console.WriteLine("myKey = " + myKey);
        }

        foreach (string myValue in mySortedList.Values)
        {
            Console.WriteLine("myValue = " + myValue);
        }
```

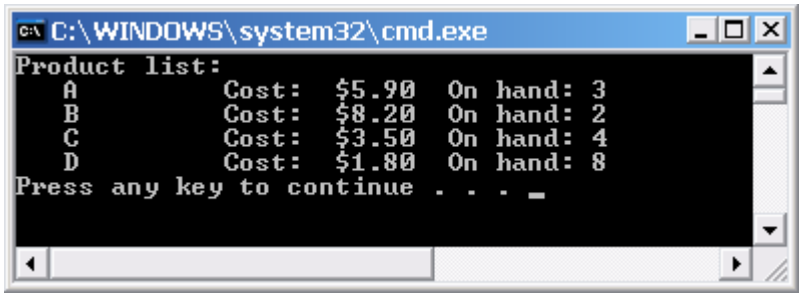
```
        Console.WriteLine("Getting the value list");
        IList myValueList = mySortedList.GetValueList();
        foreach (string myValue in myValueList)
        {
            Console.WriteLine("myValue = " + myValue);
        }
    }

}
```



3.Add user-defined object to an ArrayList

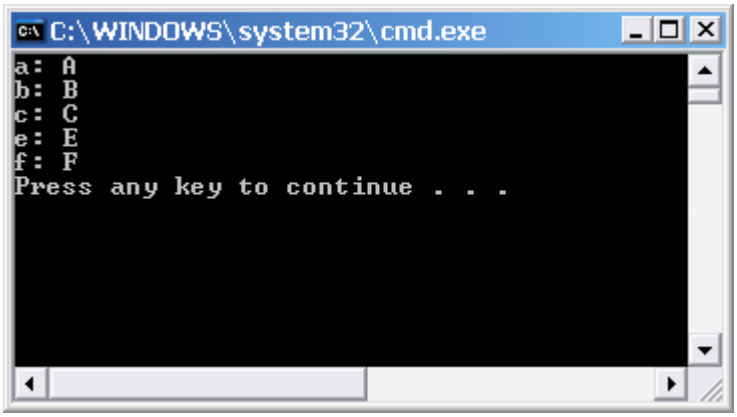
```
using System;
using System.Collections;

class Product
{
    string name;
    double cost;
    int onhand;

    public Product(string n, double c, int h)
    {
```

```
        name = n;
        cost = c;
        onhand = h;
    }

    public override string ToString()
    {
        return
          String.Format("{0,-10}Cost: {1,6:C}  On hand: {2}",
                        name, cost, onhand);
    }
}

class MainClass
{
    public static void Main()
    {
        ArrayList inv = new ArrayList();

        // Add elements to the list
        inv.Add(new Product("A", 5.9, 3));
        inv.Add(new Product("B", 8.2, 2));
        inv.Add(new Product("C", 3.5, 4));
        inv.Add(new Product("D", 1.8, 8));

        Console.WriteLine("Product list:");
        foreach (Product i in inv)
        {
            Console.WriteLine("   " + i);
        }
    }

}
```



4. Hashtable: Add key-value pair to Hashtable by using the indexer

```
using System;
using System.Collections;

class HashtableDemo
{
    public static void Main()
    {
        Hashtable ht = new Hashtable();


        ht.Add("a", "A");
        ht.Add("b", "B");
        ht.Add("c", "C");
        ht.Add("e", "E");


        ht["f"] = "F";

        // Get a collection of the keys.
        ICollection c = ht.Keys;

        foreach (string str in c)
            Console.WriteLine(str + ": " + ht[str]);
    }
}
```



Do It Yourself

7.1. Do the workshop 13 in the CD.

7.2. ACTCSharp_Module13_Assignment.pdf in CD

7.3. Create a new source file. In a method, declare a variable temperatures of type List. Add some numbers to the list. Write a foreach loop to count the number of temperatures that equal or exceed 25 degrees.
Write a method GreaterCount with signature
static int GreaterCount(List list, double min) { ... }
that returns the number of elements of list that are greater than or equal to min.
Call the method on your temperatures list.

7.4. Write a method with signature
static int GreaterCount(IEnumerable eble, double min) { ... }
that returns the number of elements of the enumerable eble that are greater than or equal to min. Then call the method on an array of type double[].

References

1) **MSDN Document**
2) **http://www.java2s.com/Tutorial/CSharp/CatalogCSharp.htm**
3) **CD ROM C# Programming, Aptech Computer Education**
4) **[ebook] MSDN training, Introduction to C#, Microsoft Press**