Collection

What are Collection?

- Groups of related objects
- here are two ways to group objects: by creating arrays of objects, and by creating collections of objects
- Array good for fix number of strong type object
- Collections provide a more flexible way to work with groups of objects

Collection using Array

```
static void Main(string[] args)
{
    // Create a list of strings.
    var salmons = new List<string>();
    salmons.Add("chinook");
    salmons.Add("coho");
    salmons.Add("pink");
    salmons.Add("sockeye");

    // Iterate through the list.
    foreach (var salmon in salmons)
    {
        Console.Write(salmon + " ");
    }
    // Output: chinook coho pink sockeye
}
```

List

```
static void Main(string[] args)
{
    // Create a list of strings by using a
    // collection initializer.
    var salmons = new List<string> { "chinook", "coho", "pink", "sockeye" };

    // Iterate through the list.
    foreach (var salmon in salmons)
    {
        Console.Write(salmon + " ");
    }
    // Output: chinook coho pink sockeye
}
```

Iterate using for

```
//iterates through the elements of a collection
// by using for instead of foreach
for (var index = 0; index < salmons.Count; index++)
{
    Console.Write(salmons[index] + " ");
}
// Output: chinook coho pink sockeye
```

Remove element

```
// Create a list of strings by using a
// collection initializer.
var salmons = new List<string> { "chinook", "coho", "pink", "sockeye" };

// Remove an element from the list by specifying
// the object.
salmons.Remove("coho");

// Iterate through the list.
foreach (var salmon in salmons)
{
    Console.Write(salmon + " ");
}
// Output: chinook pink sockeye
```

Foreach() method

```
var numbers = new List<int> { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 };

// Remove odd numbers.
for (var index = numbers.Count - 1; index >= 0; index--)
{
    if (numbers[index] % 2 == 1)
    {
        // Remove the element by specifying
        // the zero-based index in the list.
        numbers.RemoveAt(index);
    }
}
// Iterate through the list.
// A lambda expression is placed in the ForEach method
// of the List(T) object.
numbers.ForEach( number => Console.Write(number + " "));
// Output: 0 2 4 6 8
```

Generic and User Defined Type

```
public class Galaxy
{
    public string Name { get; set; }
    public int MegaLightYears { get; set; }
}

class Program
{
    static void Main(string[] args)
    {
        var myGalaxy = new List<Galaxy>
        {
            new Galaxy() { Name="Tadpole", MegaLightYears=400},
            new Galaxy() { Name="Pinwheel", MegaLightYears=25},
            new Galaxy() { Name="Milky Way", MegaLightYears=0},
            new Galaxy() { Name="Andromeda", MegaLightYears=3}
        };
        foreach (Galaxy g in myGalaxy)
        {
            Console.WriteLine(g.Name + "   " + g.MegaLightYears);
        }
    }
}
```

Exercise