## Exercise 4 – Binding to List

* What you'll learn
  + How to bind the items in a ListBox to a list of objects
  + How to bind the selected item
* Stuff to Know
  + You can use List<Person> for the list to bind to
  + C# syntax for list creation:  
    var list = new List<Person>() { new Person("x","y" ..), new Person … }
  + Use dotted notation to bind to drill down into data context
    - E.g. can bind to SelectedPerson.FirstName
  + ToString method in a class should be an override
* Best Practices
  + Bind any properties that need to be dynamic, or whose values can be shared
  + Think of binding to collections, rather than populating a control via code

### The Exercise

* Start from existing project: Exercise\_4\_BindingToList
  + The project includes Person and ViewModelBase classes that you'll use
  + Also included -- MainWindow with 2 column Grid
* Create the ViewModel
  + Create a new ViewModel
  + Add a People property that is a list of Person objects
  + Pre-populate People with several instances of Person (you can do this in your ViewModel's constructor)
* Set the DataContext
  + Set the DatContext of the main window to an instance of your ViewModel
* Create the ListBox
  + Add to MainWindow.xaml -- a ListBox in the first column
  + Bind the list to your list of people
  + Run / test => Is the number of entries in the list correct? What is displayed for each Person ?
* Fix the ListBox entries
  + Create a ToString method in Person that combines the first and last name
  + Run/test => Now what is displayed in the list?
* Display selected item info
  + Add a SelectedPerson property in your ViewModel
  + Bind the new property to the SelectedItem of the ListBox
  + Add a child StackPanel in the 2nd column of the main grid
  + Add TextBlock controls for the First and Last name of the selected person
  + Do the same for Birth and Death years
  + Run / Test -- as you select people in the main list, your labels on the right side should change