There is a starter project for you to use in the assignment.  Download it [here](https://d37djvu3ytnwxt.cloudfront.net/assets/courseware/v1/39592bd94159b8f5cad6d47a0bcac0c2/asset-v1:Microsoft+DEV204x+1T2017+type@asset+block/Mod_9_Homework.zip).

The demo for this unit showed how to create an event handler for the Create Student button.  If you have not watched the demo, do so now for an example of accessing the values in a GUI application.

The current project is very simple and only offers only a single event. Using the starter project, complete the following:

* Add a new class to the project to represent a Student with three properties for the text fields.  (Feel free to copy your Student class from previous modules)
* Create a collection to store Student objects (Use a List<T> for this assignment)
* Implement the code in the button click event handler to create a Student object and add it to the collection
* Clear the contents of the text boxes in the event handler  ***textbox.Text = "";*** or ***textbox.Clear();***
* Using the form and button, create a number of Student objects and add them to the collection (at least 3)
* There are two additional buttons on the form that can be used to move through a collection of students, (previous (<) and next (>) ).
* Create event handlers for each of these buttons that will iterate over your Students collection and display the values in the text boxes
* HINT: Use the syntax ***textbox.Text = <student property>***for assigning the values from a Student object to the text boxes
* ***NOTE: You only need to submit the MainWindow.xaml.cs file as that is where your logic resides for this exercise. You do NOT need to make any changes to the application form (MainWindow.xaml).  Visual Studio will update the xaml appropriately for the added events.  This assignment is only focused on the logic in your code.***

Grading Criteria:

1. Event handler created for the Create Student button
2. Event handler creates a Student object using values from the text boxes on the form
3. Textbox values are cleared
4. Event handler adds a Student object to the List<T>
5. Next button displays each student's information in the text boxes
6. Previous button displays each student's information in the text boxes

**Challenge:**

Search on the properties of the buttons to learn how to disable the buttons if you are at the beginning or end of the collection.  Doing this prevents an unhandled exception if the button click attempts to access a value in the collection past the end or beginning.

**Bonus challenge:**

Instead of disabling the buttons, implement code in the next or previous events to cause the iteration to wrap around.  In other words, if you reach the last student in collection, a click of the "next" button should load the first student in the collection again so that the collection is listed again.  Do the same for the "previous" button.