# Assignment 8 LINQ and Session

In this assignment, you will take what you did in the last seven assignments and enhance the functionality to include LINQ.

Before starting on this assignment read about abstraction

* <http://msdn.microsoft.com/en-us/library/bb397926.aspx>
* <http://code.msdn.microsoft.com/101-LINQ-Samples-3fb9811b>
* <http://msdn.microsoft.com/en-us/library/bb397900.aspx>
* <http://msdn.microsoft.com/en-us/library/ms178581%28v=vs.100%29.aspx>
* <http://msdn.microsoft.com/en-us/library/ms972429.aspx>
* <http://msdn.microsoft.com/en-us/library/cscsdfbt%28v=vs.110%29.aspx>

### Session

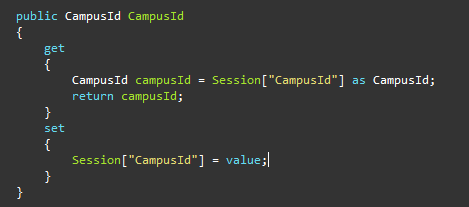
The session is used to persist data between multiple pages securely. Each user has a session associated to them, when the user closes their browser the session is closed and all data stored in it will be gone. If a user has multiple browsers, open they will have multiple sessions. Objects that are stored inside the session MUST be serializable, which means the object can be converted into a string representation of its structure and data. You can write your own serialization methods for a particular object, but 99% of the time the built in serialization is perfectly acceptable. In order to make your object serializable using built in methods, you will need to add the attribute **Serializable** to your class. If you do not have this attribute, your object will not be able to be stored in the session and you will receive an exception if you attempt to.



### Application Additions for the Session

You will need to modify your admin.aspx to have a property (with a getter and setter) that stores your list of submissions and only gets updated the first load of the page per session and any time update/delete are called from the grid. The property should be backed by the session, meaning the getter should pull the data from the session and the setter should store the data in the session. You will need to do casting when you go to retrieve the data, but not when you store it. The session will always return a generic object when you pull anything out of it. You will do this with the “as” operator. This is the safe way to cast a generic object to a specific type. If you do an implicit cats (e.g. (Type)object) you will receive an exception if the data stored in the session is not of your type or is null.

Example of a property being backed by the session:



**Note:** In a normal application you would **NOT** store submissions in the session. This would leave room for errors and improper results being returned to the user. For example if another user other than the current user modifies the records outside of the session, then the results in the session become invalid. This could occur when a new user submits the form or another administrator deleted/updated a record. This is just for demonstrating how to use the session and should not be used in an actual project.

You should also store your hear about options in the same manner. The only difference is you should only be storing them once per session, since there is no option for users to change the data.

### LINQ

LINQ is one of the most powerful tools in the .NET framework. If allows you to work with collections in many advanced ways. LINQ is kind of like writing SQL for a collection and even shares some syntactic similarities. There are two types of syntax types in LINQ, Query and Method. Both Query and Method syntaxes can accomplish the same thing. In this assignment you can use whichever makes more sense to you.

Example of the two syntaxes (from <http://msdn.microsoft.com/en-us/library/bb397947.aspx>):



### Additions for LINQ

You will need to add the following to your admin page, all data should come from the properties you created in the Session section of this assignment and all operations should use LINQ.

* A section containing the total submissions selected for each of your heard about selections
* A line that states “Show users who heard about CMU by“ with a drop down list displaying your head about options. You should use the property “AutoPostBack” equaling true, which will post back to the server when an event fires on the drop down list. You will need to use the event “OnSelectedIndexChanged”, which fires when the drop down list changes. In the event you should filter the grid down to only submissions that have the selected heard about option. The drop down list should also have a blank option, which when selected should have the grid show all results.