# Assignment 5 Abstraction and Namespaces

In this assignment, you will take what you did in the last four assignments and enhance the functionality to include data abstraction. Which means you will need to add an additional project in your solution.

Before starting on this assignment read about abstraction

* <http://en.wikipedia.org/wiki/Abstraction_%28computer_science%29>
* <http://msdn.microsoft.com/en-us/library/z2kcy19k%28v=vs.80%29.aspx>
* <http://msdn.microsoft.com/en-us/library/f3st0d45%28v=vs.100%29.aspx>

### Namespaces done right

Up until this point you have not had to do anything with namespaces. However, namespaces are extremely important and we prefer that your Project name reflects the namespace for the project. We use a more complex namespace than you will for this project, but the concept is the same. Here is an example of how we name a project CMich.PersonnelTransactions.Web. The “CMich” portion identifies that it is a project done by our development shop. The “PersonnelTransactions” portion identifies the actual project name. The “Web” portion identifies that all code inside the project is specific to the web. It contains absolutely no database calls or data modeling. That may sound crazy, but it doesn’t mean that the application cannot call into the database or some other data source it just means the underlying data interaction has been abstracted to another project. In this case the other project is CMich.PersonnelTransactions.Model. The model portion identifies that this project holds the data model. **Also please do not add anything other than letters, numbers, and periods. No symbols or spaces. Project names allow these, but namespaces will not allow spaces or symbols. Numbers CANNOT appear as the first character in any of the three portions.**

You may be asking why we would want to do this and not just put it all in one location. The simple answer is reusability. If we had all database calls in the main project it would require much more work to do two very useful things. First if we had a windows form application that needed to connect to the same data structure, we would essentially have to program that over entirely inside of that application. If we had the data model/interaction coded separately then we can use that same assembly for both projects. The second major advantage is if we need to swap out the data source for a different data source. Take for example we want to go from MySQL to MSSQL. Rather than recoding our entire application for every database call, we just need to change that one assembly and all code follows.

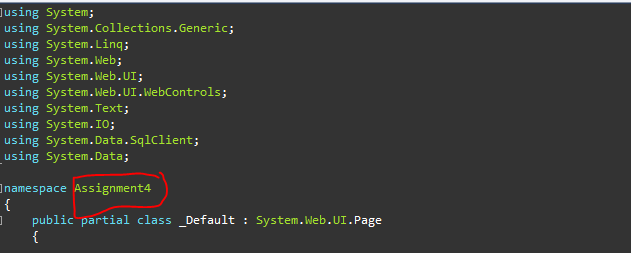
Now you will need to name your project and namespace correctly. Your namespace should be something that reflects the project name, which I am going to leave up to you. Your current project should also have the “Web” postfix because it is where the web code is going to remain. Your development shop prefix should be your globalId.

Example Bridg2bb.ContactForm.Web and Bridg2bb.ContactForm.Models would be my two projects. You do not have to use ContactForm as the project name, but you can if you want.

You will need to make sure on all of your pages and any classes you may have already created exactly reflects your new “Web” or "Models" namespace. Please note that when changing the namespace of pages, you will need to actually change it in both the aspx and cs files.

Here is an example of where to find the namespace for the cs and aspx files.

CS File



ASPX File



### New Project

You will need to add a new project to your solution. There are numerous ways to do this, but the easiest is to just right click your solution and selecting Add -> New Project. By default Visual Studio does not display the solution on certain versions. You will know if the solution is visible by looking at “Solution Explorer” and you should see something similar to this.  “StudentAssignments” would be the solution.

If the solution is not visible by default, you can make it visible by going to Tools -> Options -> Projects and Solutions. On this page there is a check box for “Always show solution”

When you add your new project select “Visual Studio C#” and then “Class Library” name the project immediately following the Model project namespace you decided above. You should automatically have a Class1.cs file in the solution. If you look at this file, you will notice that the Namespace automatically reflects the project name you selected. Any new files you add will also have that namespace

### Project Names

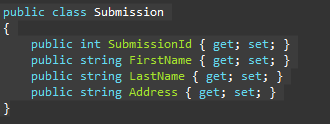
Make sure both of your projects are named correctly to reflect the namespace inside of them. You can do this by right clicking on the project in Solution Explorer and selecting rename.

### Abstracting Data

First off you will need a class that represents the table structure of your database. If you have a submission table with the following columns:

* SubmissionId INT
* FirstName VARCHAR(50)
* LastName VARCHAR(50)
* Address VARCHAR(100)

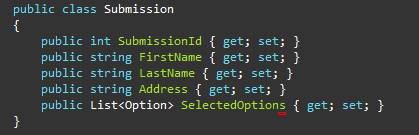
You would need the following class:



For link tables, you will not need a data model. Your other data models should have properties that reflect the fact that they are linked together.

For Example if you had two tables Submissions and Options where Submissions can have any number of options. You would add a property to the Submission data model that is a container filled with Options for that submission

Here is what the class looks like



You will also need a class, which you can simply call DAL, which stands for “Data Access Layer.” This class will contain all of your SQL calls. The class should only have SQL connection declared and part if the class’s constructor should require a connection string be called in. You will need a method for each of your stored procedure calls. For example, you should have an add/update/delete/get list for your submission table. The get list method should return a collection of your submission model.

Here is an example of a GetSubmissions method. Note that \_connection and \_connectionString are private variables assigned elsewhere in the class

