Lab: Delegates

Estimated time for completion: 30 minutes

Goals of this activity:

- Create a delegate
- Create a listener class with static and instance methods that match your delegate signature
- Invoke the listener functions via your delegate

Overview

To understand how to create a delegate, implement a method that can be invoked by the delegate and invoke the method through and instance of the delegate.

You will be creating an new Console application project using VS.NET and all of your work will be performed in that one project.

---

part 1- Starting out

Create a new console application project and rename the provided class to be called the Client class. The client class will be used to create an instance of the listener class, the delegate class and tie the two together in order to invoke the delegate.

Criteria:

- Make a new console application by opening VS.NET and chosing new project and selecting C# as the language and Console Application for your project template.
- Change the provided class name from Class1 to Client
- Create a delegate with the following signature
- `delegate void NotifyMe(string sInfo);`
- Create a class called Listener and add a static method called GetNotified that matches the signature of the delegate you just defined. Also, create a static method called GetNotifiedAgain that also matches the signature of the delegate. Add some implementation to your events that simply write out the data passed in to the console. Your Listener class should look like the following:
- `class Listener`
- `{`
- `  //instance function that matches signature of delegate above`

```
  public void GetNotified(string sInfo)
  {
          Console.WriteLine("I got notified with the following
information {0}",sInfo);
  }

  //static function that matches signature of delegate above
  public static void GetNotifiedAgain(string sInfo)
  {
          Console.WriteLine("I got notified with the following
information:{0}",sInfo);
  }
}
```

- Next, modify the Client class by adding a static method called InvokeDelegate that takes a NotifyMe delegate as a parameter. Be sure and invoke the delegate in the implementation. Your method should look like the following:

```
static void InvokeDelegate(NotifyMe d)
{
  d("You are late paying your bills!");
}
```

- Now go to your Client class and add code to Main to create an instance of your delegate, passing the Listener.GetNotifiedAgain method to the constructor and then call the InvokeDelegate function and test your application. Your client code will look something like the following:

```
class Client
{
  //Main creates the delegate, points it at a function implementation
  //and invokes the delegate
  static void Main(string[] args)
  {
          //use static function of Listener class for delegate call
          //create instance of Notify delegate and point it at static
function to call
          NotifyMe d = new NotifyMe(Listener.GetNotifiedAgain);

          //invoke the delegate function
          //notice function being called below takes a delegate that can
point
          //to ANY function on any class - this is loosly coupled!
          InvokeDelegate(d);
  }
```

- Last, create an instance of your Listener class and another instance of the NotifyMe delegate and try invoking the Instance method of the listener. Your code should look something like the following:

```
class Client
{
  //Main creates the delegate, points it at a function implementation
  //and invokes the delegate
  static void Main(string[] args)
  {
          //use static function of Listener class for delegate call
          //create instance of Notify delegate and point it at static
function to call
          NotifyMe d = new NotifyMe(Listener.GetNotifiedAgain);

          //invoke the delegate function
          //notice function being called below takes a delegate that can
point
          //to ANY function on any class - this is loosly coupled!
          InvokeDelegate(d);

          //invoke delegate using Instance function
          //create listener instance
          Listener lsnr = new Listener();

          //create delegate and point to listener instance method
          NotifyMe d2 = new NotifyMe(lsnr.GetNotified);

          //invoke just like before
          InvokeDelegate(d2);
  }
}
```

Notes:

- This sample assumes you are a little familiar with VS.NET