1) Simple console application where we wire up a container with IFoo & Foo, fetch instance.

2) Build upon #1 but add in IBar & Bar. Have Foo take IBar in the constructor to show off constructor injection.

3) Build upon #2 but setup Property injection for IBar to Foo. Setup container for property injection. **(cannot hook into Page or User Control of web forms)**

4) Build upon #3 but use Scan with Default Conventions.

5) Build upon #4 but break out all interfaces and classes into standalone files. Setup a Registry.

6) Build upon #5 but use impl convention and use snippit to bring in custom convention.

7) Build upon #6 but setup Bar with a string constructor. Setup IBar in registry to take a name property.

8) Build upon #7 but setup Generic IRepository that accepts IDomain. Setup User domain class with ID and Name. Setup UserRepository that implements IRepository<User>. Constructor inject IRepository<User> to Bar.

9) Build upon #8 but setup open generic of IRepository instead of UserRepository. Add Name to IDomain. Wire up the open generic in the registry.

10) Build upon #10 but change up Repository to have an Add method that stores data into a private list, and the Get method fetches from the list. Setup the repository as a singleton in the registry.

11) Build upon #11 but setup logging decorator for all IFoo instances.

**WebForms**

Mention how before 3.0 we used ObjectFactory that was a static class. ObjectFactory has now been marked as obsolete and will soon be going away.

**MVC**

Mention that the MVC Dependency resolution is handled using service locator pattern. It is bad 99% of the time, but in cases where you have an existing API where changes to the contract can be harmful service locator is an option.

Used nuget package StructureMap.MVC5. It is a nuget package with just source code. It adds the source code to your project instead of adding a DLL.

**Web Api**

Used OWIN, can be setup just about the same using a Global file; just put the structuremap setup in the application\_startup method.

There is a similar project called Structuremap.WebApi2 that works like StructureMap.MVC5, but it uses the dependency resolver (which is just a service locator). We can hook into the dependency resolution using a ServiceActivator instead. Make note of the use of nested container and RegisterForDispose.

**Lazy & Func examples**