Matthew Lonis

Jeff Whitmer

CSCI-A 290

October 21st, 2016

***MINI 1***

**Topic 1**

The “core” of .NET is the Common Language Runtime (CLR). Two of the primary components of the CLR are the CTS -- Common Type System and the CLS -- Common Language Specification. What is the CLR and what is its role in .NET? Why is it important to understand? What are the basics? What are the CTS and CLS all about? Why are they so crucial to the CLR?

**3 URLs**

1. <https://msdn.microsoft.com/en-us/library/zw4w595w(v=vs.110).aspx>
2. <https://msdn.microsoft.com/en-us/library/8bs2ecf4(v=vs.110).aspx>
3. <https://msdn.microsoft.com/en-us/library/zcx1eb1e(v=vs.110).aspx>
4. <http://www.developerin.net/a/39-Intro-to-.Net-FrameWork/23-Components-of-.Net-Framework>
5. <https://msdn.microsoft.com/en-us/library/12a7a7h3(v=vs.100).aspx>

**Precis of Sources**

The .NET Framework is a Microsoft technology that supports building and running the next generation of applications and XML Web Services. The .NET Framework consists of the common language runtime and the .NET Framework class library. The common language runtime manages memory, thread execution, code execution, code safety verification, compilation, and other system services. The runtime enforces code access security and enforces code robustness by implementing a strict type-and-code-verification infrastructure called the common type system (CTS). The runtime is designed to enhance performance. Although the common language runtime provides many standard runtime services, managed code is never interpreted. A feature called just-in-time (JIT) compiling enables all managed code to run in the native machine language of the system on which it is executing. Meanwhile, the memory manager removes the possibilities of fragmented memory and increases memory locality-of-reference to further increase performance.

The Common Type System describes set of data types that can be used in different .Net languages in common. It also defines how types are declared, used, and managed in the common language runtime, and is also an important part of the runtime's support for cross-language integration. For Communicating between programs written in any .NET complaint language, the types must be compatible on the basic level. To fully interact with other objects regardless of the language they were implemented in, objects must expose to callers only those features that are common to all the languages they must interoperate with. For this reason, the Common Language Specification (CLS), which is a set of basic language features needed by many applications, has been defined. The Common Language Specification is a sub set of CTS and it specifies a set of rules that needs to be adhered to or satisfied by all language compilers targeting CLR. It helps in cross language inheritance and cross language debugging.

**Summary of Topic**

The .NET Framework is a Framework that at its core includes the CLR and the .NET Framework class library. The CLR includes the CTS and CLS, both of which are crucial to the CLR. The CLR provides a runtime environment for .NET framework and allows for multiple different programs regardless of the language they were written in, to communicate and work together. The CTS manages how times are declared, used and managed by the CLR and aides in cross-language integration. The CLS provides a way for all objects, regardless of language, to interact with other objects.

**JIT 1**

What exactly is a runtime environment? What is Java’s Runtime?

**Topic 2**

.NET also makes use of a broad set of Base Class Libraries. Using our class discussion as a starting point, learn more and explain what a “class” is, as well as what a “library” is, if they are different. What are the basics? Then go further by learning what you can about the features of these libraries. It seems reasonable to assume they are unique to the .NET framework. If that is the case, how and why are they different?

**3 URLs**

1. <http://stackoverflow.com/questions/724438/what-is-the-difference-between-a-class-library-and-a-framework>
2. <https://msdn.microsoft.com/en-us/library/gg145045(v=vs.110).aspx>
3. <https://msdn.microsoft.com/en-us/library/f1yh62ef(v=vs.90).aspx>

**Precis of Sources**

A class is an extensible program-code-template for creating objects, providing initial values for variables and implementations of behavior. A Class library is a collection of class definitions. The reason behind the use of a library is for code reuse. A framework, on the other hand, is different. A framework more than often implies that you write certain pieces as dictated by the framework designers, slot them in the expected/correct places and it should work. The framework also handles all the control flow and acts like an environment more than anything else.

The .NET Framework class library is a library of classes, interfaces, and value types that provide access to system functionality. It is the foundation on which .NET Framework applications, components, and controls are built. The .NET Framework has many class libraries. For the purposes of this class, since we are going to use C#, we will use the Microsoft.CSharp namespace library. The .NET Framework class library is made up of namespaces. Each namespace contains types that you can use in your program: classes, structures, enumerations, delegates, and interfaces. When we create a Visual Basic or Visual C# project in Visual Studio, the most common base class DLLs (assemblies) are already referenced.

**Summary of Topic**

Classes are the foundation for object-oriented programming. Class libraries contain multiple classes and frameworks contain multiple class libraries. The .NET Framework has many different class libraries. The libraries are all oriented towards a specific goal. For example, there is a specific class for all C# classes. There is even a Visual Studio set of classes for use within visual studio.

**JIT 1**

Much like in C, do users have access to .NET Framework libraries and can add/modify it?

**Topic 3**

Two of the most important or most used Base Class Libraries are ADO.NET and ASP.NET. What is contained in ADO.NET? What are the basics? What makes it such a commonly used library? What features, services, methods or other factors make it important? What is contained in ASP.NET? What are the basics? What makes it such a commonly used library? What features, services, methods or other factors make it important?

**3 URLs**

1. <https://msdn.microsoft.com/en-us/library/h43ks021(v=vs.110).aspx>
2. <https://msdn.microsoft.com/en-us/library/4w3ex9c2.aspx>
3. <http://stackoverflow.com/questions/8699088/ado-net-with-asp-net>
4. <https://msdn.microsoft.com/en-us/library/ms971481.aspx>

**Precis of Sources**

ADO.NET provides consistent access to data sources such as SQL Server and XML, and to data sources exposed through OLE DB and ODBC. Data-sharing consumer applications can use ADO.NET to connect to these data sources and retrieve, handle, and update the data that they contain. ADO.NET separates data access from data manipulation into discrete components that can be used separately or in tandem. ADO.NET includes .NET Framework data providers for connecting to a database, executing commands, and retrieving results. The ADO.NET classes are found in System.Data.dll, and are integrated with the XML classes found in System.Xml.dll. ADO.NET provides functionality to developers who write managed code like the functionality provided to native component object model (COM) developers by ActiveX Data Objects (ADO). ADO.NET provides the most direct method of data access within the .NET Framework.

ASP.NET is a unified Web development model that includes the services necessary for you to build enterprise-class Web applications with a minimum of coding. ASP.NET is part of the .NET Framework, and when coding ASP.NET applications you have access to classes in the .NET Framework. ASP.NET offers three frameworks for creating web applications: ASP.NET Web Forms, ASP.NET MVC, and ASP.NET Web Pages. All three frameworks are stable and mature, and you can create great web applications with any of them. The Web Forms framework targets developers who prefer declarative and control-based programming, such as Microsoft Windows Forms (WinForms) and WPF/XAML/Silverlight. It offers a WYSIWYG designer-driven (drag-and-drop) development model, so it's popular with developers looking for a rapid application development (RAD) environment for web development. ASP.NET MVC targets developers who are interested in patterns and principles like test-driven development, separation of concerns, inversion of control (IoC), and dependency injection (DI). This framework encourages separating the business logic layer of a web application from its presentation layer. ASP.NET Web Pages targets developers who want a simple web development story, along the lines of PHP. In the Web Pages model, you create HTML pages and then add server-based code to the page to dynamically control how that markup is rendered. Web Pages is specifically designed to be a lightweight framework, and it's the easiest entry point into ASP.NET for people who know HTML but might not have broad programming experience.

**Summary of Topic**

ADO.NET is popular because it helps manage data and databases. ADO.NET is great for data-sharing consumer applications. ADO.NET could be used to develop a nationwide service for a hospital to manage and keep records of all the patients. ASP.NET is popular because it contains many classes for web development. ASP.NET has three different flavors, all designed for an audience, to give developers a more efficient way to program.

**JIT 1**

Will we be using ADO.NET and ASP.NET in this class to make a live forum like web page?