.NET Framework: Introduction

Course length: 1 day

Course Description

During the summer of 2000, at the Professional Developers Conference, the Microsoft Corporation announced its vision of the next generation of Windows services—.NET. Since that time, many steps have been taken to ensure that .NET becomes a reality. The paradigm has been refined with each release of the .NET Framework, with the public releases of the Visual Studio.NET beta software, and with the final release of both. The key to understanding .NET and how it may impact your life is to recognize the technical innovations that make it both powerful and unique. This course is the first in a series of .NET-related courses. It will provide you with the basic concepts required to decode the terminology and features of .NET, and will serve as the foundation for the other .NET courseware.
Course Objective: You will describe the .NET Framework components and the development tool, Visual Studio.NET, and discuss their role in enabling and supporting the .NET paradigm.

Target Student: This course is for you if you are familiar with object-oriented programming concepts and have an interest in .NET and Visual Studio.NET.

Prerequisites: To ensure your success, we recommend that you first take the following courses or have equivalent knowledge:

- Object-Oriented Programming Principles

Delivery Method: Instructor led, group-paced, classroom-delivery learning model with structured hands-on activities.

Performance-Based Objectives

Upon successful completion of this course, students will be able to:

- Examine the .NET paradigm and the general concepts associated with .NET.
- Identify the five main components of the .NET Framework, and explain how each functions as part of the whole.
- Identify the major components of the CLR and explain how each supports the overall function of the CLR.
- Identify the advantages of using the .NET Framework CLR.
- Describe the advantages of working with members of the .NET Class Framework.
- Characterize deployment within .NET, and show how .NET deployment eliminates DLL Hell.
- Examine the features of the new integrated IDE by building and compiling your first .NET application.

Course Content

Lesson 1: Introducing .NET–The Big Picture
Topic 1A: Characterize The .NET Paradigm
Topic 1B: Describe Web Services

Lesson 2: Building .NET–The Framework Components
Topic 2A: Describe the .NET Framework
Topic 2B: Describe the Common Language Runtime (CLR)
Topic 2C: Compare the .NET Class Framework to a Language-Specific Class Library
Topic 2D: Decide When to Use .NET Windows Forms
Topic 2E: Describe the Uses of Web Forms and Web Services
Topic 2F: Identify When to Use Console Applications

Lesson 3: Managing .NET–The Common Language Runtime Components
Topic 3A: Identify the Components of the CLR

Apptius Computer Solutions Inc.
301-135 Innovation Drive, Winnipeg, MB R3T 6A8
Phone: (204) 989-0202 | Fax: (204) 480-1045 | **www.apptius.com**

Topic 3B: Describe Microsoft Intermediate Language (MSIL)
Topic 3C: Distinguish Between the .NET Compilers
Topic 3D: Describe How the CLR Manages Memory

Lesson 4: Taking Advantage of the Common Language Runtime
Topic 4A: Identify Advantages of RAD
Topic 4B: Describe Multiple Language Support in .NET
Topic 4C: Explain Cross-Language Interoperability
Topic 4D: Explain Garbage Collection
Topic 4E: Describe Structured Error-Handling

Lesson 5: Unifying .NET–The Class Framework
Topic 5A: Describe the .NET Class Framework
Topic 5B: Describe the Purpose of Namespaces
Topic 5C: To Use or Not to Use Inheritance
Topic 5D: Differentiate Between Interface- and Inheritance-Based Polymorphism
Topic 5E: Indicate When Overloading is Used

Lesson 6: Deploying .NET Applications
Topic 6A: Describe Component Deployment
Topic 6B: Describe DLL Hell
Topic 6C: Describe an Assembly

Lesson 7: Getting Started with Visual Studio.NET
Topic 7A: Access Online Resources
Topic 7B: Set Up a VB.NET Windows Forms Application
Topic 7C: Create a Simple .NET Application
Topic 7D: Compile and Debug