

Technical Training Solutions

WEB DEVELOPMENT TECHNOLOGIES FOR MANAGERS (3 Day Course)

Technical
Course
Outline

Course Summary

Developing and implementing Enterprise applications for the web in today's world of "Web 2.0", "Cloud Computing", and "Rich Internet Applications" (RIAs), means that managers, analysts, architects, developers, and testers must rely on "mash-ups" of technologies, products, and services. Applications must meet design goals, be scalable, and, to the extent possible, follow standards.

This course provides managers a look at the technological ingredients that are used in modern mainstream web-based applications, so that they can better understand the puzzle pieces, how they fit together (or don't) and participate in the design process of today's web applications. The course will range from client-side languages, such as HTML, JavaScript and CSS, to the server-based products, technologies and architectures, such as Microsoft's .NET development platform (VB, C# and ASP .NET) and XML Web Services.

The course primarily consists of a combination of lecture, demonstration and discussion, but does have some opportunities for hands-on experimentation.

Intended Audience

This course is intended for managers and other decision-makers who may not have prior experience in development, but do need to understand the architectures, products, technologies, standards, and best practices that go into many of today's web applications.

Prerequisites

- There are no required prerequisites for this course.

Course Contents:

How Did We Get Here...And, Where Is "Here"?

- The Internet, World Wide Web, HTML and Browsers
- The Big Picture: Web Application Architectures

Development For The Client

- HTML
 - Just Because It Works, Doesn't Mean It's Right!
- The World Wide Web Consortium and Web Standards
- The CSS Revolution
- Moving & Presenting Data
 - XML, XSLT, XPath and XSD
- XHTML
- JavaScript, JScript and ECMAScript
 - The Document Object Model (DOM) Standards
- Rich Internet Applications (RIAs)
 - AJAX
 - jQuery
- Cross Browser Compatibility
- Development and Debugging Tools & Resources
 - IE's Developer Tools & FireFox's FireBug
- Client-Side Best Practices

Development For The Server

- The Job of a Web Server & Microsoft's Internet Information Server
- The HTTP Request/Response (Client/Server) Architecture Model
- Microsoft's .NET Development Platform
 - The .NET Framework
 - The Common Language Runtime (CLR)
 - Assemblies, Namespaces & .NET Class Libraries
 - .NET Naming Standards
 - The Global Assembly Cache (GAC)
 - Strong-Named Assemblies
 - Memory Management with the Garbage Collector (GC)
 - Visual Basic and/or C#
 - The Common Type Specification (CTS)
 - Naming Conventions & Best Practices
 - Visual Studio
 - Solutions and Projects
 - Project Types
 - Getting Help!
 - Exception Handling, Debugging & Testing Code
 - The Re-Sharper Add-In (for C# developers)

Object-Oriented Programming & Building Class Libraries

- Core OO Concepts
 - Classes vs. Instances
 - Abstraction, Encapsulation, Inheritance, Polymorphism
 - Overloading, Virtual/Overriding, Static/Shared, Hiding/Shadowing
- Designing The Class
 - Fields, Properties, Methods, Events, Constructors
 - Visual Studio 2010's "Class Diagram" & "Class Designer"
- Advanced Object-Oriented Programming Concepts
 - Abstract & Sealed Classes
 - Interfaces
 - Enumerable Types (Enums)

Utilizing Class Libraries

- References & Namespaces
- Instancing Classes

ASP .NET Web Development

- ASP .NET Web *Sites* vs. ASP .NET Web *Applications*
- Web Form, HTML, & HTML Server Controls
- ASP .NET Compilation Choices & Strategies
- State Management Options

ASP .NET XML Web Services

- What Are Web Services?
- Web Service Pro's and Con's
- Building Web Services
 - The **WebService** and **WebMethod** Compiler Attributes
- Using The Visual Studio Web Service Test Bed
- The Web Service Description Language (WSDL) Document
- Consuming A Web Service With A Client