WestLake
training + development

600- WestLake Capstone series: .NET Web Developer

Introduction to .NET Programming

- **What is the .Net Platform**
- Common Type System
- The Common Languate Runtime and the Common Intermediate Language
- What is the .Net Framework?
- Namespaces
- Framework Packages
- The .NET Languages
- C# as a Replacement for C++
- VB.NET as Object Oriented VB8
- Object Members and Comments
- Example: Welcome to the .NET Program
- Example: Welcome Using Alert
-

Creating a Simple Web Site

- Understanding Web Sites and Applications
- Compile Models for Web Sites in VS.NET 2005
- The ASP.NET Development Web Server
- Understanding Web Forms and Events

Creating a Simple Windows Application

- Understanding Windows Applications
- Understanding Windows Forms and Events

Introduction to Visual Studio .NET

- What is Visual Studio .NET, and is it required to write .NET and ASP.NET applications?
- VS.NET Editions and Alternatives
- The Integrated Development Environment
- Working with Solutions and Projects
- Documentation And Help
- Microsoft Document Explorer
- Help Integrated Into the IDE
- Debugging and Diagnostics
-

Setting Up Course Exercise Solutions

- Create a Blank Solution
- Console Applications
- Web ASP.NET Applications
- Windows Forms Applications

WestLake
training + development

Language Fundamentals Part 1: Data Types, Literals, and Constants

- Data Types
- Literals and Identifiers
- Variables and Constants
- Strongly Typed
- Data Casting and Conversion
- The ToString() Method
- Example: Some Conversions
-

Language Fundamentals Part 2: Statements, Expressions, and Subroutines

- Statements
- Expressions
- Subroutines
- Example: A Subroutine to Calculate Movie Sales
- Declaring Subroutines
- Calling Subroutines
- Example: Subroutines with Parameters and Return Types
- Overloading
-

Branching

- What is Branching?
- Unconditional Branching
- Conditional Branching
- If-Else Statements
- Example: Simple If Statement
- Example: Simple If-Else Statement
- Multiple Branching Conditional Statements
- Example: Simple Multiple Branch
-

Loops

- For Loops
- Example: For Loops
- While and Do-While Loops
- Example: While and Do-While Loops
- Loop Jump Statements
-

Operators

- Introduction
- Operator Precedence
- Mathematical
- Logical
- Bitwise
- Boolean
- Relational
- Short Circuiting
-

Classes and Objects

- Introduction to Object Oriented Concepts
- Why do we need object oriented programming?
- Classes are Templates
- Members
- Class Members vs. Object Members
- Objects are Instantiated From Classes
- To Instantiate an Object, Call the Class's Constructor
- Classes Inheritance
- Example: Building and Using a Movie Viewer Class
- Declaring a Class
- Setting Up Data Members to Hold the Property Values
- Defining Properties
- Declaring the Constructor Method
- Defining Methods
-

Inheritance

- Generalization and Specialization
- Inheritance and Polymorphism
- Constructors
- Example: Inheriting from the Movie Viewer Class
- The .NET Object Base Class
- Example: Overriding the ToString() Method of the Object Class
- Value and Reference Types
- Boxing and Unboxing
-

Interfaces

- What is an Interface?
- Example: Defining and Implementing an IRenter Interface
-

Arrays

- What Are Arrays?
- Declaring and Initializing Arrays
- Iterating Through Arrays Using For-Each
- Example: Using For-Each to Iterate Through an Array
- Multi-Dimensional Arrays
- Example: Rectangular Arrays
- Example: Jagged Arrays
- Passing Arrays as Parameters
-

Collections Part 1: Hashtables and Enumeration

- Hashtables
- Indexers/Default Properties
- Example: Hashtable
- Encapsulating Collections with ICollection, IEnumerable, and IEnumerator
- Example: Implementing An Encapsulated Collection

Collections Part 2: ArrayLists, Other Collection Types, and Sorting

- ArrayLists
- Example: Simple ArrayList
- Some Less Common Collection Classes
- Queues
- Example: Simple Queue
- Stacks
- Example: Simple Stack
- Sorting using IComparable and IComparer
- Example: Implementing IComparabable and ArrayList.Sort() To Enable Sorting

Strings

- What is the String Class?
- Strings are Immutable
- Example: Some Useful String Properties and Methods
- Comparing Strings
- What is the StringBuilder Class?
- Example: Working with the StringBuilder Class
- Regular Expressions and the RegEx Class
- Example: A Regular Expression Tester
- Special Characters for Pattern Matching
- Escape Sequences for Special Characters
- Specifying Ranges in Patterns
- Matching a Specified Number of Occurrences
- Character-Range Escape Sequences
- Matching at the Beginning or End of a String with ^ and $
- The Word-Boundary Pattern Anchors: \b and \B

Introduction to Streams

- What are Streams?
- Reading and Writing Data
- Why are there Streams and Reader/Writer Classes?
- Why do we need Streams?
- Example: Writing to a Tab-Delimited File

Structured Exception Handling

- What is an Exception?
- What is Structured Exception Handling?
- The Exception Class
- Try-Catch-Finally
- Example: Throwing and Catching Exceptions

ADO.NET Introduction

- What is ADO.NET?
- An Overview of Relational Database Concepts

- What is a database?
- Characteristics of Relational Databases – The Basics
- What is an RDBMS?
- RDBMS Features
- The ADO.NET Object Model
- The Built-In Managed Providers: SQL, OleDB
- Additional Connection Classes
- Getting the Data Out
- The IDataReader Interface
- Introducing the Database Used for the Course
- Example: Selecting Records Using a DataReader
-

ADO.NET DataSets and the Disconnected Model

- What is a DataSet?
- Accessing Data Through a DataSet's Tables and Rows
- Using IDataAdapter
- Example: Selecting Records from a Database using a DataSet
-

Advanced: ADO.NET Commands and Stored Procedures

- What is the Command class?
- Executing a Stored Procedure or Query
- Example: Executing a Query Using a Command Object
- Passing Input Parameters to a Stored Procedure or Query
- Example: Passing Input Paramters to a Query Using a Command Object
-

Advanced: ADO.NET Inserts, Updates, and Deletes

- Modifying Data
- Example: A Form to Insert, Update, and Delete Data
-

Introduction to .NET Programming

- What is the .NET Platform?
- What is the .NET Framework?
- The .NET Languages
- Example: Welcome to .NET Program
- Example: Welcome Using Alert
-

Creating a Simple Web Site

- Understanding Web Sites and Applications
- Compile Models for Web Sites in VS.NET 2005
- The ASP.NET Development Web Server
- Understanding Web Forms and Events

Refresher: Introduction to Visual Studio .NET

- What is Visual Studio .NET, and is it required to write .NET and ASP.NET applications?

- The Integrated Development Environment
- Working with Solutions and Projects
- Documentation And Help
- Debugging and Diagnostics

Setting Up Course Exercise Solutions

- Create a Blank Solution

Introducing ASP.NET

- What is ASP.NET?
- What software do I need in order to run ASP.NET applications?
- The Page Class
- Structure of an ASP.NET Web Application
- Example: Outputting the Current Time
- Example: Processing Form Submissions in ASP.NET

Fundamentals of Web Forms and Controls

- The Architecture of a Page: Pages as Web Forms
- Adding Controls to Your Forms

Introduction to Server Controls

- What are Server Controls
- Page Control Hierarchy
- ASP Web Controls vs. HTML Controls
- Using the Visual Studio .NET Web Form Editor
- Example: Calculating Birthdays

Custom User Controls

- What are User Controls?
- The Architecture of a User Control: User Controls as Reusable Components

Events in ASP.NET

- What is an Event?
- Page events
- Web Server Control Events
- Example: Simple Loan Calculator
- Immediate versus Cached Event Submission

Refresher: ADO.NET Introduction

- What is ADO.NET?
- An Overview of Relational Database Concepts
- What is an RDBMS?
- The ADO.NET Object Model
- Getting the Data Out
- The IDataReader Interface

- Introducing the Database Used for the Course
- Example: Selecting Records Using a DataReader

Refresher: ADO.NET DataSets and the Disconnected Model

- What is a DataSet?
- Accessing Data Through a DataSet's Tables and Rows
- Example: Selecting Records from a Database using a DataSet

Introduction to Data Binding in Web Forms

- What is Data Binding?
- What is a Data Source?
- DataSource Code and DataSource Controls
- Data Binding in Code
- Data Binding with DataSource Controls

Data Drilldowns with Data Bound Controls

- What is a Data Drill Down Interface?
- Example: A sample Data Bound Data Drill Down
- Example: Details and More

Web Development Support

- The Request Object
- The Response Object
- The Server Object
- The Uri Class
- The SmtpClient and MailMessage Classes

ASP.NET Applications, The Application Object, and the HttpApplicationState Class

- What is an Application?
- Inspecting the Global.asax File

The Session Object and the HttpSessionState Class

- What is a Session?
- How Does ASP.NET Keep Track of Sessions?
- Example: A User Name as a Session Variable

Application Configuration With Web.config

- ASP.NET Configuration is XML
- ASP.NET Configuration is Applied in a Cascading Way
- Inspecting Configuration Files
- Example: Using Web.config to Create Custom Error Pages

Working with Validation Controls

- What are Validation Controls?

- Example: RequiredFieldValidator, CompareValidator, and RangeValidator
- The ValidationSummary Control

Regular Expression Validation

- Introducing Regular Expressions
- The RegularExpression Validator Control

Custom Validators

- The CustomValidator Control

Introducing XML

- What is XML?
- XML Logical Structure
- XML Physical Structure
- XML Design Patterns
- Applications of XML (and related technologies)

Producing XML from Database Data via ADO.NET

- Example: Generate and View XML from a DataSet

Performing XSLT with ASP.NET

- XSLT in ASP.NET

Deployment

- Overview of Deployment in .NET
- VS.NET Build Configurations
- Suggested Build Configuration Model
- Copying a Web Application
- Pre-compiling and Copying a Web Site
- Creating a Web Setup Project

Advanced: Deployment Settings Using Conditional Compilation With Compile-Time Constants

- Compile-Time Compilation Constants
- Example: Using Bulid Configuration Settings to Show and Hide Controls

Advanced: Introducing Components

- Why Are Components Useful?
- Using Assemblies to Package and Deploy Components