

Developing Windows Applications with Microsoft Visual Studio 2010

Code:	10262
Length:	5 days
URL:	View Online

In this course the student will learn how to develop rich client applications using .NET Framework 4.0. This also helps prepare you for certification exam 70-511.

Skills Gained

After completing this course, students will be able to:

- understand how varying business requirements influence the design decisions when planning a Windows Client application.
- understand the new features of Visual Studio 2010 WPF
- design and build a UI that provides the expected end-user experience and UI functionality
- create a consistent and manageable user interface
- understand best practices when testing and learn how to debug their applications
- use advanced exception handling in Windows Client application scenarios
- implement advanced data binding scenarios
- use coding techniques to improve the responsiveness of their applications
- implement localization, user assistance, and accessibility features within an application
- understand the basics of graphics in WPF
- customize controls and introduce students to custom controls
- implement application behaviors based on user actions or events by using attached properties and Expression Blend behaviors
- develop data visualization within their applications in a manner that enables the application user to drill down into data visually
- manage application state and settings throughout the application lifecycle
- deploy their applications using the various methods supported by Visual Studio 2010

Who Can Benefit

This course is intended for Technology Specialists in the area of Windows Client Development who work in a development

environment that uses Microsoft Visual Studio .NET 2010 and Microsoft .NET Framework 4.0 to create rich client applications for Windows.

Prerequisites

Before attending this course, students must have:

- An understanding of the problem-solving techniques that apply to software development, including the following principles of software development:
 -
 - modern software development models
 - typical phases of a software development lifecycle
 - concepts of event-driven programming
 - concepts of object-oriented programming
 - creating use-case diagrams
 - designing and building a user interface
 - developing a structured application
- A general understanding of the purpose, function, and features of following .NET Framework topics:
 -
 - Common Language Runtime
 - .NET Framework class library
 - Common Type System
 - Component interoperation
 - Cross-Language Interoperability
 - Assemblies in the Common Language Runtime
 - Application Domains
 - Runtime hosts supported by the .NET Framework
- Experience using Visual Studio 2008 in the following task areas:
 -
 - Declaring and initializing typed variables using the Camel case naming convention
 - Using arithmetic, relational, and logical operators in code statements
 - Using branching statements to control code execution
 - Using looping statements to iterate through collections or repeat steps until a specified condition is met
 - Creating classes and methods to establish the basic structure of an application
 - Using methods and events to implement the programming logic of an application
 - Identifying syntax and logic errors
 - Accessing and managing data from a data source
- Experience in object oriented design and development as follows:
 -
 - Creating and accessing classes and class properties
 - Creating and accessing methods and overloaded methods

- Implementing inheritance, base classes, and abstract classes
- Declaring, raising, and handling events
- Responding to and throwing exceptions
- Implementing interfaces and polymorphism
- Implementing shared and static members
- Implementing generics
- Creating components and class libraries
- Experience in N-Tier application design and development as follows:
 - - Managing a software development process
 - Controlling input at the user interface level in Windows Client applications
 - Debugging, tracing, and profiling .NET applications
 - Monitoring and logging .NET applications
 - Implementing basic testing best practices
 - Performing basic Data Access tasks with LINQ
 - - Basics of LINQ to XML
 - Basics of LINQ to Entities
 - Basics of LINQ to SQL
 - Implementing basic security best practices in .NET Applications
 - - Basics of Code Access Security
 - Basics of Role-Based Security
 - Basics of Cryptography Services
 - Implementing basic service calls
 - - Basics of consuming XML Web Services
 - Basics of consuming WCF Services
 - Using .NET Configuration Files
 - Deploying .Net Framework Applications using ClickOnce and the MS Installer

Course Details

Outline

Module 1: Windows Client Application Design

The goal of this module is to ensure that students understand how varying business requirements influence the design decisions when planning a Windows Client application. Students will learn how design requirements, specifications, and business goals affect the choice between WPF and Windows Forms when updating (or planning a new) Windows Client application.

Lessons

- Windows Client Technologies
- Architectural Patterns

- Interoperability between Windows Forms and WPF

Lab : Planning Windows Client Applications

After completing this module, students will be able to:

- Choose appropriate Windows client technologies
- Choose appropriate architectural patterns
- Identify areas for migration from WinForms
- Identify areas for interoperability between WinForms and WPF

Module 2: Introduction to Visual Studio 2010 and WPF Version 4

The goal of this module is to introduce students to the new features that Visual Studio 2010 and WPF version 4 provide

Lessons

- Whats New in Visual Studio 2010?
- Whats New in WPF Version 4

Lab : Guided Tour of Visual Studio 2010 and XAML

After completing this module, students will be able to:

- Use new features in Visual Studio 2010
- Use new features in WPF version 4

Module 3: Designing and Developing a User Interface

The goal of this module is to teach the student how to design and build a UI that provides the expected end-user experience and UI functionality, and retains that experience/functionality in various end-user environments, as well as when the application window resizes.

Lessons

- Defining Page Layout
- Using Content Controls
- Using Item Controls
- Sharing Logical Resources in a Window

Lab : Building a User Interface

After completing this module, students will be able to:

- Define page layout
- Use content controls
- Use item controls
- Share resources within a window or user control

Module 4: Taking Control of the User Interface

The goal of this module is to enable students to create a consistent and manageable user interface.

Lessons

- Sharing Logical Resources in an Application
- Creating Consistent User Interfaces by Using Styles
- Changing the Appearance of Controls by Using Templates

- Handling Events and Commands

Lab : Dynamically Controlling the User Interface

After completing this module, students will be able to:

- Share logical resources throughout an application
- Create a consistent user interface by using styles
- Change the appearance of controls by using templates
- Handle events and commands

Module 5: Testing, Unit Testing, and Debugging

The goal of this module is to help students to develop good habits in regard to testing as well as enabling students to debug their applications. Students will also learn about advanced exception handling pertaining directly to Windows Client application scenarios

Lessons

- WPF Testing Strategies
- Debugging XAML
- Providing User Feedback for Unhandled Exceptions
- Understanding Security Features

Lab : Testing and Debugging WPF Applications

After completing this module, students will be able to:

- Implement a WPF unit test strategy
- Debug XAML by using the WPF Visualizer and PresentationTraceSources
- Provide user feedback for unhandled exceptions
- Understand security features of an application

Module 6: Simple Data Binding and Validation

The goal of this module is to teach the student how to implement simple data binding and data validation in order to manage data in a data source (CRUD). In addition to learning how to bind a value to a UI element, the student should learn best practices concerning when to use the various implementations of data binding and how to connect to a data source with LINQ.

Lessons

- Overview of Data Binding
- Creating a Data Binding
- Implementing Property Change Notification
- Converting Data
- Validating Data
- Presenting Data at Design Time

Lab : Data Binding

After completing this module, students will be able to:

- Explain WPF data binding concepts and terminology
- Create a binding between a data source and a control
- Implement property change notification
- Convert data between the binding source and the binding target

- Validate data entered by the user
- Present data at design time

Module 7: Data Binding to Collections

The goal of this module is to enable the student to implement more advanced data binding scenarios, picking up where the previous module on data binding left off. The student will learn how to bind to ListView, GridView, DataGrid, and other collection classes.

Lessons

- Binding to Collections of Objects
- Using Collection Views
- Creating Master-Detail User Interfaces
- Using Data Templates
- Presenting Design Time Data Collections

Lab : Data Binding to Collections

After completing this module, students will be able to:

- Bind to a collection of objects
- Sort, filter, and group collections by using collection views
- Create master-detail user interfaces
- Customize data display by using data templates
- Present data collections at design time

Module 8: Enhancing UI Responsiveness

The goal of this module is to teach the students how coding techniques can be used to improve the responsiveness of their applications.

Lessons

- Implementing Asynchronous Processes
- Implementing Responsive User Interfaces

Lab : Enhancing Application Performance

After completing this module, students will be able to:

- Implement asynchronous processes
- Implement responsive user interfaces

Module 9: Integrating Localization and User Assistance Features

The goal of this module is to teach students how to implement localization, user assistance, and accessibility features within an application.

Lessons

- Localization and Globalization
- Implementing User Assistance Features
- Providing User Accessibility Features

Lab : Localizing a WPF Application

After completing this module, students will be able to:

- Describe WPF localization and globalization features
- Implement localized WPF applications
- Implement user assistance features

- Provide user accessibility features

Module 10: WPF 2D Graphics, Multimedia, and Printing

The goal of this module is to teach the students the basics of graphics in WPF. The module will also provide an overview of multimedia (audio and video).

Lessons

- Displaying 2D Graphics
- Displaying Images
- Adding Multimedia to WPF Applications
- Creating and Printing Documents

Lab : Drawing 2-D Graphics

After completing this module, students will be able to:

- Display 2D graphics
- Add images in a WPF application
- Add multimedia content to a WPF application
- Create and print documents

Module 11: Control Customization

The goal of this module is to teach students how to customize controls and introduce students to custom controls.

Lessons

- Overview of Control Authoring
- Creating User Controls
- Creating Custom Controls
- Managing Control Appearance by Using Visual States
- Integrating WPF and Windows Forms

Lab : Building a User Control

After completing this module, students will be able to:

- Explain scenarios and options for creating new controls
- Create user controls
- Create custom controls
- Integrate WPF and WinForms controls

Module 12: Attached Properties and Behaviors in WPF

The goal of this module is to teach the students how to implement application behaviors based on user actions or application events by using attached properties and Expression Blend behaviors.

Lessons

- Implementing Attached Properties
- Implementing Drag-and-Drop User Interfaces
- Implementing Expression Blend Behaviors, Triggers and Actions

Lab : Implementing Drag-and-Drop Operations

After completing this module, students will be able to:

- Describe WPF attached properties
- Implement drag and drop
- Implement Expression Blend behaviors

Module 13: Animations in WPF

The goal of this module is to teach the student how to implement animations and to teach students how to develop data visualization within their applications in a manner that enables the application user to drill down into data visually. Students should also learn when animations should be used and why.

Lessons

- Using Animations
- Using Triggers
- Implementing Data Visualizations

Lab : Creating Animations

After completing this module, students will be able to:

- Explain when animations are appropriate for the user interface
- Implement animation in WPF
- Initiate animation by using triggers
- Present data visualizations by using WPF

Module 14: Application State, Settings, and Lifecycle

The goal of this module is to teach students how to manage application state and settings throughout the application lifecycle.

Lessons

- Creating Application Settings
- Consuming Application Settings
- Creating Custom Configuration Sections

Lab : Creating a Settings Dialog

After completing this module, students will be able to:

- Persist user and application settings
- Consume user settings

Module 15: Configure and Deploy Windows Client Applications

The goal of this module is to teach students how to deploy their applications using the various methods supported by Visual Studio 2010.

Lessons

- Deployment Options
- Deploying a Standalone WPF Application
- Deploying an XBAP Application
- Configuring Security Settings

Lab : Deploying Applications

After completing this module, students will be able to:

- Explain deployment options

- Deploy a standalone WPF application by using a Setup and Deployment project
 - Deploy a ClickOnce application
 - Configure security settings
-

ExitCertified® Corporation and iMVP® are registered trademarks of ExitCertified ULC and ExitCertified Corporation and Tech Data Corporation, respectively
Copyright ©2021 Tech Data Corporation and ExitCertified ULC & ExitCertified Corporation.
All Rights Reserved.

Generated 2